# 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 = 'codegemma'
In [4]: config = {
            'model': model name, # 'mistral' # "starcoder2"
        vn = MyVanna(config=config)
In [5]: hostname = os.uname().nodename
        print("Hostname:", hostname)
       Hostname: ducklover1
```

nostname: dacktover i

```
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.CustomerId, COUNT(ii.InvoiceLineId) A	sql
	1	0658ba3d-98ff-51f4-9006- a24f87045858-sql	How many customers are there	SELECT COUNT(*) FROM "customers"	sql
	2	127fd4bd-b9af-539d-9313- 1d0234d073b7-sql	\n There are 3 tables: artists, albums and	SELECT a.Name, COUNT(t.TrackId) AS TotalTracks	sql
	3	3013d1b4-feb2-519d-bfb9- 114500436e3d-sql	\n Find the customer with the most invoi	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot	sql
	4	32b99e7b-31ab-55d8-8431- fb010fa7af85-sql	\n Find the top 5 customers who spent th	SELECT c.CustomerId, SUM(i.Total) AS TotalSpen	sql
	5	49e67df3-a604-51f8-ad01- b8f5a2043eac-sql	\n Get the total number of invoices for e	SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tot	sql
	6	584873f8-1904-50f1-8f80- 7ccf08059264-sql	\n List all customers from Canada and th	SELECT c.Email, c.Country\nFROM "customers" c\	sql
	7	6bed484b-9a80-57f4-ad89- 5f775b5df252-sql	\n Get the average invoice total for each	SELECT c.CustomerId, AVG(i.Total) AS AverageIn	sql
	8	6f22268c-5062-5f11-ba2d- 8555f06b409d-sql	\n Find all tracks with a name containing	SELECT * \nFROM "tracks" \nWHERE LOWER(Name) L	sql
	9	70b4f686-c71b-5ee8-9458- 6bbc776349bf-sql	\n Find all invoices since 2010 and the t	SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmo	sql
	10	9a396a33-ecea-51a8-bd05- 28f58a86eb86-sql	\n Hint: album quantity is found in invoi	SELECT c.Customerld, COUNT(ii.Trackld) AS Tota	sql
	11	9a9c970b-b94c-5f22-b54c- b86921a38b65-sql	\n Identify artists who have albums with	SELECT a.ArtistId, a.Name AS ArtistName\nFROM	sql
	12	a7185c88-7417-5b75-a52e- 4eaef5f9deca-sql	\n List all albums and their correspondin	SELECT a.Title, a.ArtistId, ar.Name AS ArtistN	sql
	13	aea89953-21b2-55d1-9dda- 431ee6033c3d-sql	\n List all invoices with a total exceedi	SELECT * \nFROM "invoices" \nWHERE Total > 10.00	sql
	14	d1d70c18-f5d9-5970-a32c- 914deeca1087-sql	\n Find the customer who bought the most	SELECT c.Customerld, COUNT(ii.Trackld) AS Tota	sql
	15	d8a2f948-dffa-5524-a5f9- 174cc1a8da73-sql	Can you list all tables in the SQLite database	SELECT name FROM sqlite_master WHERE type='table'	sql
	16	d8a37163-5ce5-58cd-a316-	what are the top 5 countries	SELECT c.Country, COUNT(*) AS	sql

	id	question	content	training_data_type
	ea5598d44d27-sql	that customers co	TotalCustomers\n	
17	dd282d7c-a4ef-5e3a-87e0- cb45fac50808-sql	\n Find the total number of invoices per	SELECT i.BillingCountry, COUNT(*) AS TotalInvo	sql
18	e7c4b3aa-664f-5f87-8b25- 449a4482f3fd-sql	\n Get all playlists containing at least	SELECT pt.PlaylistId, p.Name AS PlaylistName,	sql
19	f33f8cb6-1b12-5ea7-8d9a- aef8166b9970-sql	\n Find the top 5 most expensive tracks (	SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "t	sql
20	f626b681-4d8f-563a-beee- 1ea759baaa82-sql	\n List all genres and the number of trac	SELECT g.Name, COUNT(t.GenreId) AS TotalTracks	sql
21	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 ArtistId IN	ddl
1	0db84e3d-ef41-563c-803e- 21c1b985dc19-ddl	None	CREATE TABLE "invoices"\r\n(\r\n InvoiceId	ddl
2	10cba811-ddba-5042-9e90- d764dfcd1629-ddl	None	CREATE INDEX IFK_InvoiceCustomerId ON "invoice	ddl
3	2c711317-b93d-5f60-a728- cb1c6fcbc040-ddl	None	CREATE INDEX IFK_CustomerSupportRepId ON "cust	ddl
4	37319c81-65f7-50ee-956b- 795de244bee5-ddl	None	CREATE TABLE sqlite_stat1(tbl,idx,stat)	ddl
5	40bd77cd-e1de-5872-8693- 624117ff413c-ddl	None	CREATE INDEX IFK_InvoiceLineInvoiceId ON "invo	ddl
6	41130543-7164-562a-90a7- 0fd0a409c154-ddl	None	CREATE TABLE "albums"\r\n(\r\n AlbumId INTE	ddl
7	458debc8-8082-5450-a17a- 66028bd55ace-ddl	None	CREATE TABLE "playlists"\r\n(\r\n Playlist!	ddl
8	4815f3fd-925b-53ce-9dfa- 0e4285d5abd3-ddl	None	CREATE TABLE "invoice_items"\r\n(\r\n Invoi	ddl
9	48d484e9-984c-58ff-b391- 75521c69d486-ddl	None	CREATE INDEX IFK_PlaylistTrackTrackId ON "play	ddl
10	551e1120-a6ee-554f-8b8a- ccf4f22d3636-ddl	None	CREATE INDEX IFK_AlbumArtistId ON "albums" (Ar	ddl

	id	question	content	training_data_type
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 CustomerI	ddl
21	e5879308-329e-543f-a693- 0c14e2f9972e-ddl	None	CREATE INDEX IFK_InvoiceLineTrackId ON "invoic	ddl
22	ea84418b-1a28-59b4-a1f4- 2fb674208adc-ddl	None	CREATE TABLE sqlite_sequence(name,seq)	ddl
0	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

Out[13]:		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"\ $r\n$ [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]:
    if False:
        # 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
   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,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 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 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 TABLE "media type s"\r\n(\r\n MediaTypeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\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:

Ollama parameters: model=codegemma:latest, options={}.

keep\_alive=None

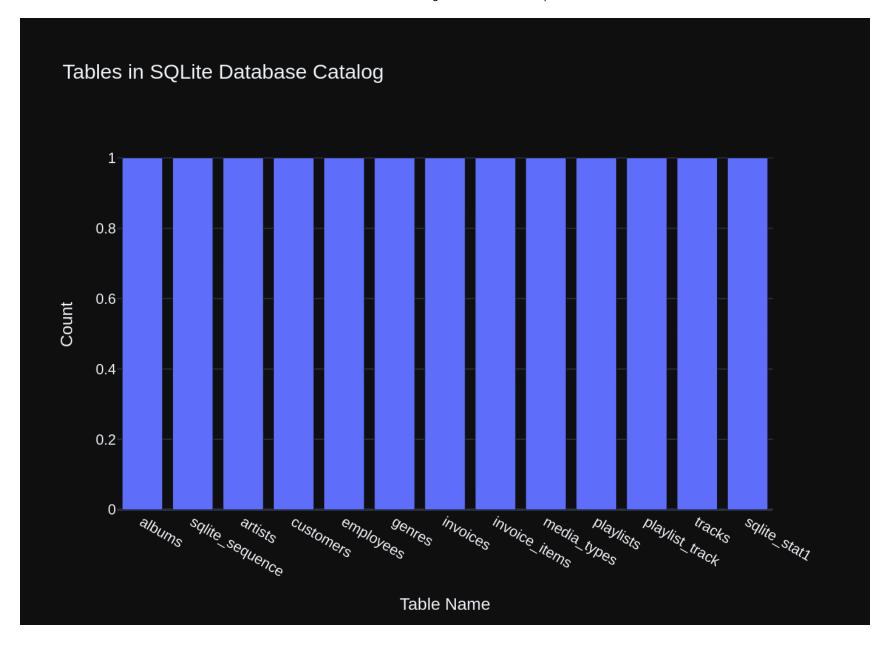
Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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\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\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\nCREATE TABLE \"invoice items\"\r\n(\r\n  $L,\r\n$ InvoiceLineId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n 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': 'codegemma:latest', 'created at': '2024-06-13T22:23:55.953309228Z', 'message': {'role': 'assistan
t', 'content': "SELECT name FROM sqlite master WHERE type='table'"}, 'done reason': 'stop', 'done': True,
'total duration': 61378011525, 'load duration': 2077984799, 'prompt eval count': 1658, 'prompt eval duratio
n': 56210673000, 'eval count': 12, 'eval duration': 2455748000}
SELECT name FROM sqlite master WHERE type='table'
SELECT name FROM sqlite master WHERE type='table'
               name
0
             albums
1
    sqlite sequence
2
            artists
3
          customers
4
          employees
5
             genres
6
           invoices
7
      invoice items
8
        media types
9
          playlists
10
     playlist track
11
             tracks
12
       sqlite stat1
Ollama parameters:
model=codegemma:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'Can you list all tables in the SQLite database catalog?'\n\nThe
DataFrame was produced using this guery: SELECT name FROM sglite master WHERE type='table'\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n name
e: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of t
he dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataf
rame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:24:15.807720729Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='name', title='Tables in SQLit
e Database Catalog')\n\nfig.update layout(\n xaxis title='Table Name',\n yaxis title='Count'\n)\n\nfi
q.show()\n``"}, 'done reason': 'stop', 'done': True, 'total duration': 19826653274, 'load duration': 66780
8, 'prompt eval count': 149, 'prompt eval duration': 5641634000, 'eval count': 64, 'eval duration': 1409462
8000}
```



```
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
           10
                playlist track
           11
                        tracks
           12
                  sglite stat1,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'name=%{x}<br/>br>count=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v'.
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['albums', 'sqlite_sequence', 'artists', 'customers', 'employees',
                                      'genres', 'invoices', 'invoice items', 'media types', 'playlists',
                                      'playlist track', 'tracks', 'sqlite stat1'], dtype=object),
                         'xaxis': 'x',
                         'y': array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Tables in SQLite Database Catalog'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Table Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Count'}}}
           }))
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 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 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 UnitPrice NUMERIC(10.2) NOT NULL.\r InvoiceId INTEGER NOT NULL.\r\n 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 TABLE "customers"\r\n(\r\n CustomerId INTEGER P LastName NVARCHAR(20) N RIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) 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 FirstName NVARCHAR(20) NOT NULL,\r\n RCHAR(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 State NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n  $HAR(40), \r\n$ Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n ne NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n FOREIGN KEY (ReportsTo) REFEREN CES "employees" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sqlite se quence(name,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 FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t\t0N DELETE rtistId INTEGER NOT NULL,\r\n 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\tON 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 t monev overall, \n Hint: order total can be found on invoices table, calculation using invoice \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': 'what are the top 5 countries that customers come from?'}, {'role': 'assis tant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nOR DER BY TotalCustomers DESC\nLIMIT 5'}, {'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': '\n Get the average invoice total fo r each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoice Total\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 i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountr y'}, {'role': 'user', 'content': "which table stores customer's orders"}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n Billin 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\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 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  $0), r\n$ State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NV  $ARCHAR(24).\r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL.\r\n SupportRepId INTEGER.\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO A CTION\r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL L,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(3 Address NVARCHAR 0), r nReportsTo INTEGER.\r\n BirthDate DATETIME.\r\n HireDate DATETIME.\r\n  $(70), \r\n$ 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  $(\r\n$ PlavlistId INTEGER NOT NULL.\r\n TrackId INTEGER NOT NULL.\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"media types\"\r\n(\r\n peld INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Name NVARCHAR(120)\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 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 guery 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 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", "content 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": " \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 \"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 Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in tota 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": "wha t 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 List all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.Email, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canad a'"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJO IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.Billin qCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "co ntent": "which table stores customer's orders"}] Ollama Response:

{'model': 'codegemma:latest', 'created\_at': '2024-06-13T22:25:36.27719413Z', 'message': {'role': 'assistan t', 'content': 'The "invoices" table stores customer orders. Each row in the table represents a single invoice.'}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 80053449238, 'load\_duration': 674455, 'promp t\_eval\_count': 1846, 'prompt\_eval\_duration': 74525417000, 'eval\_count': 20, 'eval\_duration': 4892958000} The "invoices" table stores customer orders. Each row in the table represents a single invoice. The "invoices" table stores customer orders. Each row in the table represents a single invoice. Couldn't run sql: Execution failed on sql 'The "invoices" table stores customer orders. Each row in the table represents a single invoice.': 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 **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 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 FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON U NOT NULL,\r\n 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 LastName NVARCHAR(20) NOT NULL,\r\n FirstName 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 State NVARCHAR(40),\r\n PostalCode N  $(70), \r\n$ City NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60).\r\n VARCHAR(10),\r\n Phone NVARCHAR(24),\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

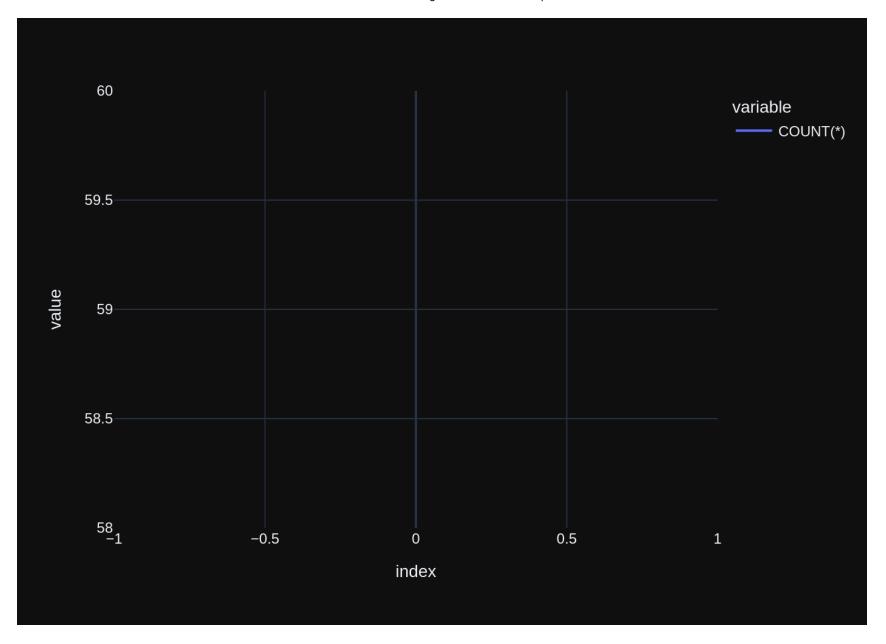
il table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalS pent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought 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 Get the average invoice total for ea ch customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTota l\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the total number of invoices per country:\n'}, {'role': 'assistant', 'cont ent': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': 'How many customers are there'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin gAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(40),\r\n BillingState NVARCHAR(40),\r\n BillinaCou 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\t0N DELETE NO ACTION ON UPDATE NO ACTION \r\n)\n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nCREATE TABLE \"customers FirstName NVARCHAR(40) NOT NU \"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n City NVARCHAR(40),\r\n PostalCode NVARCHAR(1 0),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n Fax NVARCHAR(24),\r\n Support

RepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\t0N 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 iceId 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 D Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\t0 ELETE NO ACTION ON UPDATE NO ACTION.\r\n 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\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLine TrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NUL BirthDate DATETIME.\r\n L.\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER.\r\n HireDate DATETIM State NVARCHAR(40),\r\n E, r nAddress NVARCHAR(70),\r\n City 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 Name  $NVARCHAR(120)\r\n)\n\n===Additional Context \n\nIn the SQLite database invoice$ MENT NOT NULL.\r\n means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please generate a valid S QL query without any explanations for the question. \n2. If the provided context is almost sufficient but r equires knowledge of a specific string in a particular column, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most re levant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it was given before. \n"}, {"role": "user", "content": "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\n0RDER 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\nJ0IN \"invoices\" i ON c.CustomerId = i.Customer Id\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5"}, {"role": "user", "content": " \n 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 guan 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 Hint: album quantity is found in invoice items, \n e": "user", "content": " \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 Get the average invoice total for each customer:\n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invo ices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, C OUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": "Ho w many customers are there"}]

Add of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql Insert of existing embedding ID: 0658ba3d-98ff-51f4-9006-a24f87045858-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:26:50.267923127Z', 'message': {'role': 'assistan
t', 'content': 'SELECT COUNT(*) FROM "customers"'}, 'done reason': 'stop', 'done': True, 'total duration':
73945077797, 'load duration': 958339, 'prompt eval count': 1681, 'prompt eval duration': 71552952000, 'eval
count': 8, 'eval duration': 1764248000}
SELECT COUNT(*) FROM "customers"
SELECT COUNT(*) FROM "customers"
   COUNT(*)
         59
0
Ollama parameters:
model=codegemma:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'How many customers are there'\n\nThe DataFrame was produced usin
g this query: SELECT COUNT(*) FROM \"customers\"\n\nThe following is information about the resulting pandas
DataFrame 'df': \nRunning df.dtypes gives:\n COUNT(*) int64\ndtype: object"}, {"role": "user", "content
t": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a
pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with o
nly Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:27:06.429904722Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.indicator(\n
                                                                                  df.\n
          title='Number of Customers'\n)\n\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total
(*)',\n
duration': 16136673040, 'load duration': 678517, 'prompt eval count': 117, 'prompt eval duration': 4852533
000, 'eval count': 45, 'eval duration': 11193775000}
```



```
Out[19]: ('SELECT COUNT(*) FROM "customers"',
              COUNT(*)
           0
                    59,
           Figure({
               'data': [{'hovertemplate': 'variable=COUNT(*)<br>index=%{x}<br>value=%{y}<extra></extra>',
                         'legendgroup': 'COUNT(*)',
                         'line': {'color': '#636efa', 'dash': 'solid'},
                         'marker': {'symbol': 'circle'},
                         'mode': 'lines',
                         'name': 'COUNT(*)',
                         'orientation': 'v',
                         'showlegend': True,
                         'type': 'scatter',
                         'x': array([0]),
                         'xaxis': 'x',
                         'y': array([59]),
                         'yaxis': 'y'}],
               'layout': {'legend': {'title': {'text': 'variable'}, 'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'index'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'value'}}}
          }))
In [ ]:
In [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 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 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 Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n PostalCode NVARCHAR(10).\r\n Email NVARCHAR(60) NOT SupportRepId INTEGER,\r\n NULL,\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 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 Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n LL,\r\n FirstName NVARCHAR(20) NOT NULL,\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 FOREIGN KEY (ReportsTo) REFERENCES "employee Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n 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\cREATE 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\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeI d) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===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

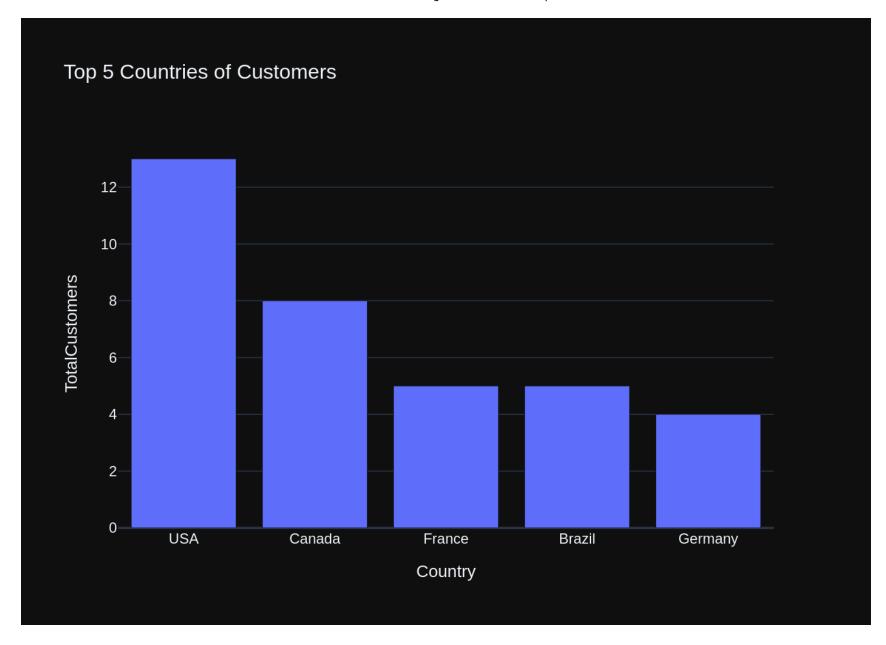
ery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant table(s). \n5. If the guestion has been asked and answered before, please repeat the answer exactly as it was given before. \n'}, {'role': '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 Find the top 5 most expen sive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitP rice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'How many customer s are there'}, {'role': 'assistant', 'content': 'SELECT COUNT(\*) FROM "customers"'}, {'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.InvoiceId\n GROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'role': 'user', 'content': 'what are the top 5 countries that customers come from?'\1 Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n Billin

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 FOREIGN KE R NOT NULL.\r\n UnitPrice NUMERIC(10,2) 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 \n e most money overall, \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 Find the top 5 customers who bought the most albums in total quantity found in invoice items, \n \n (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 er", "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.C ustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"role": "user", Hint: album quantity is found in invoice items. \n \n Find the top 5 customers wh "content": " \n 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 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.UnitPrice DES C\nLIMIT 5"}, {"role": "user", "content": "How many customers are there"}, {"role": "assistant", "content": "SELECT COUNT(\*) FROM \"customers\""}, {"role": "user", "content": " \n Find the customer who bought t he most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.Cu stomerId, 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 TotalAl bums DESC\nLIMIT 1"}, {"role": "user", "content": "what are the top 5 countries that customers come fro m?"}l

Add of existing embedding ID: d8a37163-5ce5-58cd-a316-ea5598d44d27-sql Insert of existing embedding ID: d8a37163-5ce5-58cd-a316-ea5598d44d27-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:28:30.487754072Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER
BY TotalCustomers DESC\nLIMIT 5'}, 'done_reason': 'stop', 'done': True, 'total duration': 83956663321, 'loa
d duration': 713196, 'prompt eval count': 1776, 'prompt eval duration': 74862620000, 'eval count': 33, 'eva
l duration': 8454393000}
SELECT c.Country, COUNT(*) AS TotalCustomers
FROM "customers" c
GROUP BY c.Country
ORDER BY TotalCustomers DESC
LIMIT 5
SELECT c.Country, COUNT(*) AS TotalCustomers
FROM "customers" c
GROUP BY c.Country
ORDER BY TotalCustomers DESC
LIMIT 5
   Country TotalCustomers
       USA
0
                         8
1
   Canada
                         5
2 France
                         5
3 Brazil
                         4
4 Germany
Ollama parameters:
model=codegemma:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: 'what are the top 5 countries that customers come from?'\n\nThe D
ataFrame was produced using this query: SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM \"customers\" c
\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5\n\nThe following is information about the resul
ting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Country
                                                                           object\nTotalCustomers
                                                                                                      int64
\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': 'codegemma:latest', 'created at': '2024-06-13T22:28:49.347854475Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfiq = px.bar(df, x='Country', y='TotalCustomers',
title='Top 5 Countries of Customers')\n\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total dur
ation': 18834425107, 'load duration': 695414, 'prompt eval count': 174, 'prompt eval duration': 7299655000,
'eval count': 44, 'eval duration': 11444574000}
```



```
Out[20]: ('SELECT c.Country, COUNT(*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nORDER BY TotalCust
         omers DESC\nLIMIT 5',
              Country TotalCustomers
          0
                 USA
                                   13
          1 Canada
                                    8
                                    5
          2 France
                                    5
           3 Brazil
          4 Germany
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{x}<br>TotalCustomers=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['USA', 'Canada', 'France', 'Brazil', 'Germany'], dtype=object),
                         'xaxis': 'x',
                         'y': array([13, 8, 5, 5, 4]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Top 5 Countries of Customers'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Country'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalCustomers'}}}
          }))
```

### 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 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 AlbumArtistId ON "albums" (ArtistId)\n\nCREATE TABLE "alb AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NUL FOREIGN KEY (ArtistId) REFERENCES "artists" (ArtistId) \r\n\t L.\r\n ArtistId INTEGER NOT NULL,\r\n \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 UnitPrice NUMERIC(10.2) NOT NULL.\r\n NULL,\r\n FOREIGN KEY (AlbumId) REFERENC ES "albums" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFEREN CES "genres" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) RE FERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX I FK TrackAlbumId ON "tracks" (AlbumId)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AU 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 "genres"\r\n(\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 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'}, {'role': 'user', 'conte Hint: album quantity is found in invoice items, \n \n Find the top 5 customers 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 5'}, {'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 "invoic e items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'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.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoice i tems" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'rol e': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}, {'role': 'assi stant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\nJOIN "tracks" t ON q.G enreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' \n Get all playlists containing a t 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\nJ0IN "p laylists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROUP BY pt.Playlis tId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': ' \n Find the top 5 most exp ensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.Uni tPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': 'Can you list al l tables in the SQLite database catalog?'}, {'role': 'assistant', 'content': "SELECT name FROM sglite maste r WHERE type='table'"}, {'role': 'user', 'content': ' \n List all albums and their corresponding artist names \n'}l Ollama parameters: model=codegemma:latest,

options={}.

keep alive=None

Prompt Content:

[{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE 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 Medi GenreId INTEGER,\r\n aTypeId INTEGER NOT NULL.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEG ER 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 (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 \"qenres\"\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 guery to find the distinct strings in that column. Prepend the guery 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", 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 q2.GenreId\n FROM \"genres\" q2\n JOI 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"}, {"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 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": " 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 quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in tota l quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.Invo iceLineId) 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"}, {"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 playlists containing at least 10 tracks and the total duration n 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.Pl aylistId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.Trac kId) >= 10"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit pric e):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "Can you list all tables in the SQLite database catalog?"}, {"role": "assistant", "content": "SELECT name FROM sqlite master WHERE type='table'"}, {"role": "user", "content": " \n List 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

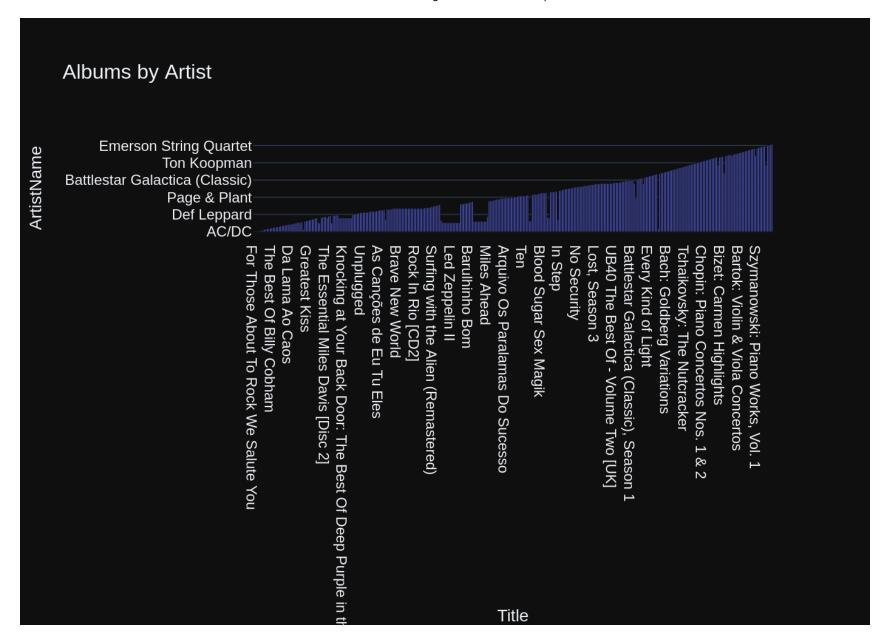
```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:30:10.206279662Z', 'message': {'role': 'assistan
t', 'content': 'SELECT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.
ArtistId = ar.ArtistId'}, 'done reason': 'stop', 'done': True, 'total duration': 80763921205, 'load duratio
n': 685847, 'prompt eval count': 1586, 'prompt eval duration': 70035942000, 'eval count': 39, 'eval duratio
n': 10091699000}
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 \
0
                 For Those About To Rock We Salute You
                                                                1
                                     Balls to the Wall
1
2
                                     Restless and Wild
3
                                     Let There Be Rock
                                                               1
4
                                                                3
                                              Big Ones
                                                              . . .
342
                                Respighi:Pines of Rome
                                                              226
     Schubert: The Late String Quartets & String Qu...
                                                              272
                                   Monteverdi: L'Orfeo
                                                             273
344
345
                                 Mozart: Chamber Music
                                                              274
346 Koyaanisgatsi (Soundtrack from the Motion Pict...
                                                             275
                                            ArtistName
0
                                                 AC/DC
1
                                                Accept
2
                                                Accept
3
                                                 AC/DC
4
                                             Aerosmith
342
                                        Eugene Ormandy
343
                                Emerson String Quartet
     C. Monteverdi, Nigel Rogers - Chiaroscuro; Lon...
345
                                         Nash Ensemble
346
                                 Philip Glass Ensemble
[347 rows x 3 columns]
Ollama parameters:
model=codegemma:latest,
options={},
```

keep\_alive=None
Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n List all albums and their corresponding artist names \n'\n\nThe DataFrame was produced using this query: SELECT 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 resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Title object\nArtistId int64\nArtistName 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. Respond with only Python code. Do not answer with any explana tions -- just the code."}]

Ollama Response:

{'model': 'codegemma:latest', 'created\_at': '2024-06-13T22:30:29.114851912Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Title', y='ArtistName', title ='Albums by Artist')\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 1888006435 8, 'load\_duration': 626179, 'prompt\_eval\_count': 187, 'prompt\_eval\_duration': 8143825000, 'eval\_count': 41, 'eval duration': 10647469000}



```
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
                                                                             2
           3
                                                  Let There Be Rock
                                                                             1
           4
                                                           Big Ones
                                                                             3
                                                                           . . .
           . .
           342
                                             Respighi: Pines of Rome
                                                                           226
           343
                Schubert: The Late String Quartets & String Qu...
                                                                           272
           344
                                                Monteverdi: L'Orfeo
                                                                           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': 'Title=%{x}<br>ArtistName=%{y}<extra></extra>',
                          'legendgroup': '',
                          'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                          'name': '',
                          'offsetgroup': ''
                          'orientation': 'v',
                          'showlegend': False,
                          'textposition': 'auto',
                          'type': 'bar',
                          'x': array(['For Those About To Rock We Salute You', 'Balls to the Wall',
                                       'Restless and Wild', ..., "Monteverdi: L'Orfeo",
```

```
'Mozart: Chamber Music',
                                     'Koyaanisqatsi (Soundtrack from the Motion Picture)'], dtype=object),
                         'xaxis': 'x',
                         'y': array(['AC/DC', 'Accept', 'Accept', ...,
                                     'C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sa
         ckbu',
                                     'Nash Ensemble', 'Philip Glass Ensemble'], dtype=object),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Albums by Artist'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Title'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'ArtistName'}}}
          }))
         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

file:///home/gongai/projects/wgong/py4kids/lesson-18-ai/vanna/docs/ollama-codegemma-chromadb-sqlite-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 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 "playlists"\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 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 "tracks" t ON 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.InvoiceId\n GROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'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': 'us Hint: album quantity is found in invoice items, \n \n er', 'content': ' \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\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 Hint: album quantity is found in invoi Find the top 5 customers who bought the most albums in total quantity (across all inv \n oices):\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.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find all tracks with a name containing "W hat" (case-insensitive)\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 Bvtes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n GER 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 EGER 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 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\nJOIN \"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 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", "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 There are 3 tables: artis ts, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by Alb umId,\n Can you find the top 10 most popular artists based on the number of tracks\n"}, {"role": "assist ant", "content": "SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId $\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId \nGROUP BY a.Name \nORDER BY TotalTrack$ s DESC\nLIMIT 10"}, {"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 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 c.CustomerId, COUNT(ii.TrackId) AS TotalAl bums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJOIN \"invoice items\" ii O N 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 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

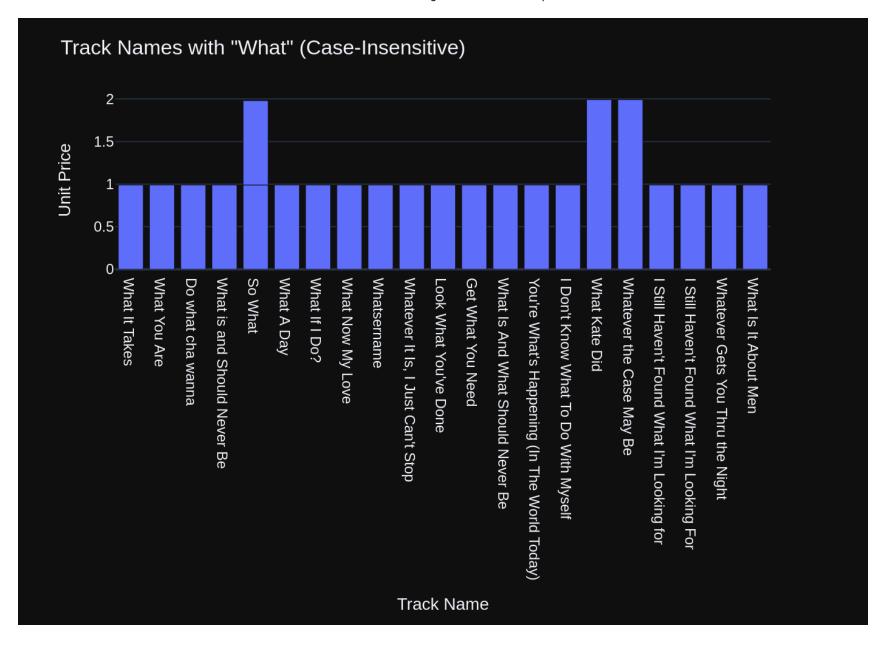
N i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}]

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': 'codegemma:latest', 'created at': '2024-06-13T22:31:45.769027519Z', 'message': {'role': 'assistan
t', 'content': 'SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, 'done_reason': 'stop', 'don
e': True, 'total duration': 76415815336, 'load duration': 694892, 'prompt eval count': 1627, 'prompt eval d
uration': 70717840000, 'eval count': 20, 'eval duration': 5057237000}
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
                                             What If I Do?
                                                                  80
       1000
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
                                                                 146
             You're What's Happening (In The World Today)
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
18
       2992
                                                                 237
               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
               1
                           Dave Grohl, Taylor Hawkins, Nate Mendel, Chris...
                       12
7
               1
                                    carl sigman/gilbert becaud/pierre leroyer
8
               1
                        4
                                                                     Green Day
               1
                        1
9
                                                              Jay Kay/Kay, Jay
               1
                        4
10
                                                                     N. Cester
               1
                        4
                                                C. Cester/C. Muncey/N. Cester
11
12
               1
                        1
                                                      Jimmy Page, Robert Plant
                       14
13
               1
                                        Allen Story/George Gordy/Robert Gordy
                        3
14
               1
                                                                  Culmer/Exalt
                        7
               1
15
                                                                           None
               3
                       19
16
                                                                           None
               3
17
                       19
                                                                           None
18
              1
                        1
                               Bono/Clayton, Adam/Mullen Jr., Larry/The Edge
               1
                        1
19
                                                                             U2
                        9
20
               2
                                                                           None
               2
                           Delroy "Chris" Cooper, Donovan Jackson, Earl C...
21
    Milliseconds
                       Bytes UnitPrice
0
                    10144730
          310622
                                    0.99
1
                                    0.99
          249391
                     5988186
2
                                    0.99
                     9018565
          274155
3
          260675
                     8497116
                                    0.99
4
          564009
                    18360449
                                    0.99
5
          158275
                     5203430
                                    0.99
6
          302994
                     9929799
                                    0.99
7
                     4913383
                                    0.99
          149995
8
          252316
                                    0.99
                     8244843
9
                                    0.99
          247222
                     8249453
          230974
                                    0.99
10
                     7517083
          247719
                     8043765
                                    0.99
11
12
          287973
                     9369385
                                    0.99
                     4631104
13
                                    0.99
          142027
14
          189152
                     6162894
                                    0.99
15
                                    0.99
          221387
                     7251478
16
         2610250
                   484583988
                                    1.99
17
                  183867185
         2616410
                                    1.99
                                    0.99
18
          353567
                    11542247
19
          280764
                     9306737
                                    0.99
20
          215084
                                    0.99
                     3499018
          209573
21
                     3426106
                                    0.99
Ollama parameters:
model=codegemma:latest,
```

options={}. keep alive=None Prompt Content: [{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Find all tracks with a name containing \"What\" (case-in sensitive)\n'\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': 'codegemma:latest', 'created at': '2024-06-13T22:32:13.22540461Z', 'message': {'role': 'assistan t', 'content': '```python\nimport plotly.express as  $px\n = px.bar(df, x=\Name\', y=\UnitPrice\')\n\n$ fig.update layout(\n title=\'Track Names with "What" (Case-Insensitive)\',\n xaxis title=\'Track Name yaxis title=\'Unit Price\'\n)\n\nfig.show()\n```'}, 'done reason': 'stop', 'done': True, 'total du ration': 27429692524, 'load duration': 635948, 'prompt eval count': 212, 'prompt eval duration': 848781000 0, 'eval count': 77, 'eval duration': 18846163000}



Out[22]:	('SE	('SELECT * \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\'',					
		TrackId		Name	AlbumId	\	
	0	26		What It Takes	5		
	1	88		What You Are	10		
	2	130		Do what cha wanna	13		
	3	342		What is and Should Never Be	30		
	4	607		So What	48		
	5	960		What A Day	76		
	6	1000		What If I Do?	80		
	7	1039		What Now My Love	83		
	8	1145		Whatsername	89		
	9	1440	W	hatever 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		're What'	s Happening (In The World Today)	146		
	14	1823		So What	149		
	15	2772	I D	on't Know What To Do With Myself	223		
	16	2884		What Kate Did	231		
	17	2893		Whatever the Case May Be	230		
	18			ven't Found What I'm Looking for	237		
	19			ven't Found What I'm Looking For	238		
	20	3258		Whatever Gets You Thru the Night	255		
	21	3475		What Is It About Men	322		
		MediaTypeId	GenreId		,	Compocor	`
	0	nedialypeid 1	1	Steven Tyler, Joe Per		•	\
	1	1	1		ave/Chris		
	2	1	2	Audios		rge Duke	
	3	1	1	limmy	Page/Robe		
	4	1	2	31111119		es Davis	
	5	1	1	Mike Bordin, Billy 0			
	6	1	1	Dave Grohl, Taylor Hawkins, Nate			
	7	1	12	carl sigman/gilbert beca			
	8	1	4	ou. 1 0_gu., g_ 100. 1 0000	•	reen Day	
	9	1	1		Jay Kay/k	-	
	10	1	4			. Cester	
	11	1	4	C. Cester/C.			
	12	1	1		Page, Robei		
	13	1	14	Allen Story/George G			
	14	1	3		-	er/Exalt	
	15	1	7			None	

```
16
              3
                       19
                                                                          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
0
          310622
                   10144730
                                   0.99
                    5988186
                                   0.99
1
          249391
2
                    9018565
                                   0.99
          274155
3
          260675
                    8497116
                                   0.99
4
          564009
                   18360449
                                   0.99
5
                                   0.99
          158275
                    5203430
6
                    9929799
                                   0.99
          302994
7
                                   0.99
          149995
                    4913383
8
                                   0.99
          252316
                    8244843
9
          247222
                    8249453
                                   0.99
10
          230974
                    7517083
                                   0.99
                                   0.99
11
          247719
                    8043765
12
          287973
                                   0.99
                    9369385
          142027
                    4631104
                                   0.99
13
          189152
                                   0.99
14
                    6162894
          221387
                    7251478
                                   0.99
15
16
         2610250
                  484583988
                                   1.99
17
         2616410
                  183867185
                                   1.99
18
          353567
                   11542247
                                   0.99
          280764
                    9306737
19
                                   0.99
          215084
20
                                   0.99
                     3499018
          209573
21
                     3426106
                                   0.99
Figure({
    'data': [{'alignmentgroup': 'True',
              'hovertemplate': 'Name=%{x}<br>UnitPrice=%{y}<extra></extra>',
              'legendgroup': '',
              'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
              'name': '',
              'offsetgroup': '',
              'orientation': 'v',
               'showlegend': False,
              'textposition': 'auto',
              'type': 'bar',
              'x': array(['What It Takes', 'What You Are', 'Do what cha 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([0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99, 0.99,
                                     0.99, 0.99, 0.99, 0.99, 1.99, 1.99, 0.99, 0.99, 0.99, 0.99]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'title': {'text': 'Track Names with "What" (Case-Insensitive)'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Track Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Unit Price'}}}
          }))
         question = """
In [23]:
             Get the total number of invoices for each customer
         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 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 RCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER,\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 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 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 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\nJ0IN "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\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP 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 List all invoices with a total exceeding \$1 0:\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'use r', 'content': '\n Get the total number of invoices for each customer\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL.\r\n CustomerId INTEGER NOT NULL.\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin BillingState NVARCHAR(40),\r\n BillinaCou gAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n 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 \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 Phone NVARCHAR(2 PostalCode NVARCHAR(10).\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n 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 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 FOREIGN KEY (ReportsTo) REFERENCES \"employees\" Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\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 NOT NULL,\r\n Name NVARCHAR(200) 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 lbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" FOREIGN KEY (MediaTypeId) REFERENCES \"me (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n 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 guery 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 Find the total number of invoices per country:\n"}, {"role": "assistant", "conten r", "content": " \n 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", "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.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n Find the top 5 customers who \n Hint: order total can be found on invoices table, calculation u spent the most money overall. \n 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 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 List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "user", "content": " \n 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': 'codegemma:latest', 'created at': '2024-06-13T22:33:42.15362354Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoice
s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done reason': 'stop', 'done': True, 'total d
uration': 88815844935, 'load duration': 738381, 'prompt eval count': 1872, 'prompt eval duration': 77487299
000, 'eval count': 42, 'eval_duration': 10691605000}
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
2
             3
                            7
3
                            7
             4
             5
                            7
4
5
                            7
             6
             7
                            7
6
7
             8
                            7
                            7
8
             9
                            7
9
            10
                            7
10
            11
                            7
11
            12
12
            13
13
                            7
            14
                            7
14
            15
15
                            7
            16
16
            17
17
            18
                            7
18
            19
                            7
19
            20
                            7
20
            21
                            7
21
            22
22
            23
                            7
23
                            7
            24
                            7
24
            25
25
                            7
            26
26
            27
```

27	28	7
28	29	7
29	30	7
30	31	7
31	32	7
32	33	7
33	34	7
34	35	7
35	36	7
36	37	7
37	38	7
38	39	7
39	40	7
40	41	7
41	42	7
42	43	7
43	44	7
44	45	7
45	46	7
46	47	7
47	48	7
48	49	7
49	50	7
50	51	7
51	52	7
52	53	7
53	54	7
54	55	7
55	56	7
56	57	7
57	58	7
58	59	6
Ollama	parameters:	
model = 0	odegemma:latest.	

model=codegemma:latest,

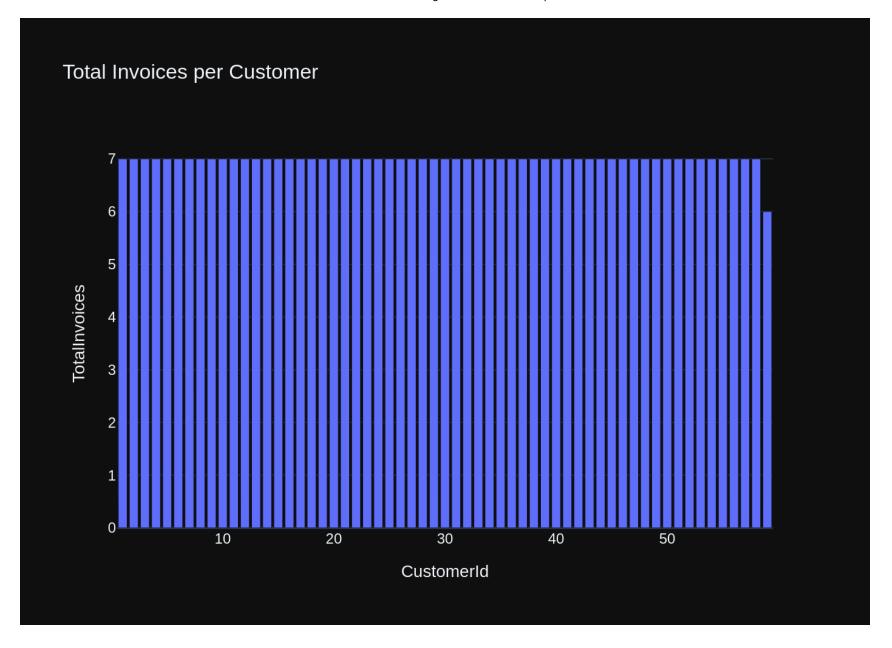
options={},

keep alive=None

Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Get the total number of invoices for each customer\n'\n \nThe DataFrame was produced using this query: SELECT c.CustomerId, 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 wit
h any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created\_at': '2024-06-13T22:34:01.323337806Z', 'message': {'role': 'assistan
t', 'content': "``python\nimport plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='TotalInvoice
s', title='Total Invoices per Customer')\nfig.show()\n``"}, 'done\_reason': 'stop', 'done': True, 'total\_du
ration': 19143396917, 'load\_duration': 708696, 'prompt\_eval\_count': 188, 'prompt\_eval\_duration': 804638700
0, 'eval count': 43, 'eval duration': 11006526000}



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',

come		omeria/ugkoup B
	CustomerId	TotalInvoices
0	1	7
1	2	7
2	3	7
3	4	7
4	5	7
5	6	7
6	7	7
7	8	7
8	9	7
9	10	7
10	11	7
11	12	7
12	13	7
13	14	7
14	15	7
15	16	7
16	17	7
17	18	7
18	19	7 7
19	20	7
20 21	21 22	7
		7
22 23	23 24	7
24	25	7
25	26	7
26	27	7
27	28	7
28	29	7
29	30	7
30	31	7
31	32	7
32	33	7
33	34	7
34	35	7
35	36	7
36	37	7
37	38	7
38	39	7

```
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
          56
                        7
          57
                        7
56
                        7
57
          58
                        6,
58
          59
Figure({
   'data': [{'alignmentgroup': 'True',
            'hovertemplate': 'CustomerId=%{x}<br>TotalInvoices=%{y}<extra></extra>',
            'legendgroup': '',
            'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
            'name': '',
            'offsetgroup': '',
            'orientation': 'v',
            'showlegend': False,
            'textposition': 'auto',
            'type': 'bar',
            'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                      19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                      37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                      55, 56, 57, 58, 59]),
            'xaxis': 'x',
            7, 7, 7, 7, 7, 7, 7, 7, 7, 6]),
            'yaxis': 'y'}],
   'layout': {'barmode': 'relative',
             'legend': {'tracegroupgap': 0},
```

[{'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 \t0N 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 Email NVA  $R(40), \r\n$ PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsTo) REFERENCES "employees" (EmployeeId) \r\n\t\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 FirstName NVARCHAR(40) NOT NULL,\r\n T NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n  $NVARCHAR(80).\r\n$ Address NVARCHAR(70),\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 sql \n3. If the provi ded context is insufficient, please explain why it can\'t be generated. \n4. Please use the most relevant t able(s). \n5. If the question has been asked and answered before, please repeat the answer exactly as it wa s given before. \n'}, {'role': 'user', 'content': ' \n 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\'\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 List all invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'user', 'content': ' \n Hint: album quantity is found in invoice items, \n Find the t op 5 customers who bought the most albums in total quantity (across all invoices):\n'}, {'role': 'assistan t', 'content': 'SELECT c.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 the top 5 customers w ho spent the most money overall, \n \n Hint: order total can be found on invoices table, calculatio n using invoice items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.Customer Id, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJ0IN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'content': '\n Find the custom er 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 Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\n FROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'us er', 'content': ' \n Find the total number of invoices per country:\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 gAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n BillingState 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 FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI T NULL,\r\n Quantity 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 InvoiceCustomerId ON \"invoi ces\" (CustomerId)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"invoice items\" (InvoiceId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n EmployeeI d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVAR LastName NVARCHAR(20) NOT NULL.\r\n 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 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 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 State NVARC  $HAR(40), \r\n$ 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  $NVARCHAR(24).\r\n$ FOREIGN KEY (Support RepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n Albumid INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTE AlbumId INTEGER.\r\n GER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n GenreId INTEGER,\r\n MediaTypeId INTEGER NOT NULL.\r\n Composer NVARCHAR(220),\r\n Milliseconds I NTEGER NOT NULL,\r\n Bvtes 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\n\$  IFK EmployeeReportsTo ON \"employees\" (ReportsTo)\n\n\===Additional Context \n\nIn the S QLite 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 Get t 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

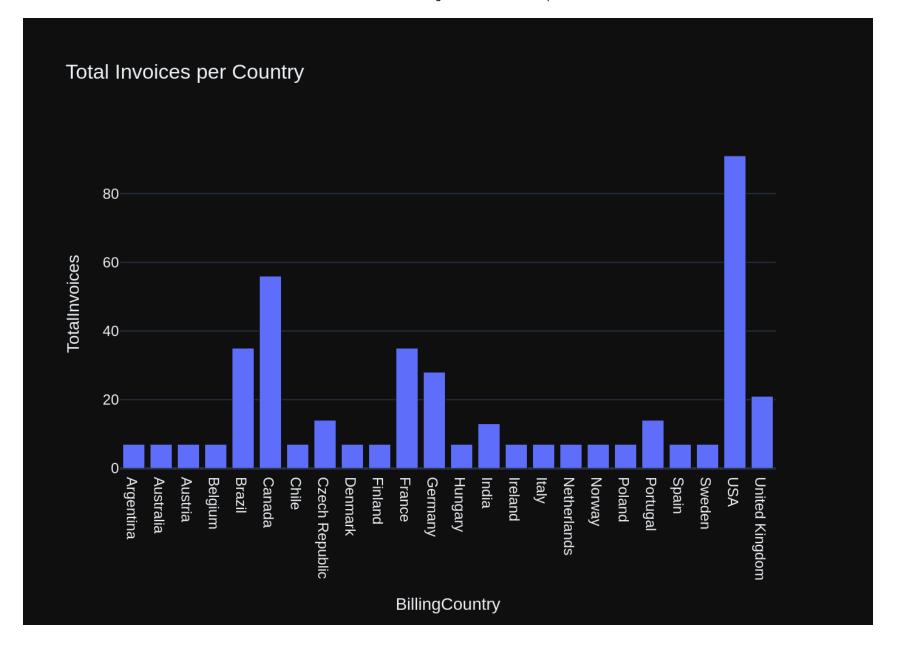
FROM \"invoices\" i\nWHERE i.InvoiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "conte Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.Cu nt": " \n stomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON 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": " List all invoices with a total exceeding \$10:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"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 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\nJ0IN \"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 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.Inv oiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "conte Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELEC T c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJ0IN \"invoices\" i ON c.Custome rId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}]

Add of existing embedding ID: dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql Insert of existing embedding ID: dd282d7c-a4ef-5e3a-87e0-cb45fac50808-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:35:31.280288135Z', 'message': {'role': 'assistan
t', 'content': 'SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCo
untry'}, 'done reason': 'stop', 'done': True, 'total duration': 89864674283, 'load duration': 750602, 'prom
pt eval count': 1926, 'prompt eval duration': 82750414000, 'eval count': 25, 'eval duration': 6478972000}
SELECT i.BillingCountry, COUNT(*) AS TotalInvoices
FROM "invoices" i
GROUP BY i.BillingCountry
SELECT i.BillingCountry, COUNT(*) AS TotalInvoices
FROM "invoices" i
GROUP BY i.BillingCountry
    BillingCountry TotalInvoices
0
         Argentina
                                7
                                7
1
         Australia
2
                                7
           Austria
                                7
3
           Belgium
                               35
4
            Brazil
5
                               56
            Canada
6
             Chile
                                7
7
   Czech Republic
                               14
8
           Denmark
                                7
                                7
9
           Finland
10
           France
                               35
11
                               28
           Germany
12
           Hungary
                                7
                               13
13
             India
                                7
14
           Ireland
15
             Italv
                                7
                                7
16
       Netherlands
                                7
17
            Norway
                                7
18
            Poland
19
                               14
          Portugal
20
                                7
             Spain
                                7
21
            Sweden
22
               USA
                               91
                               21
23 United Kingdom
Ollama parameters:
model=codegemma:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
```

that answers the question the user asked: '\n Find the total number of invoices per country:\n'\n\nThe DataFrame was produced using this query: SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoice s\" i\nGROUP BY i.BillingCountry\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n BillingCountry object\nTotalInvoices 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."}]
Ollama Response:

{'model': 'codegemma:latest', 'created\_at': '2024-06-13T22:35:50.252772492Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='BillingCountry', y='TotalInvo ices', title='Total Invoices per Country')\n\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'tota l\_duration': 18945260434, 'load\_duration': 641791, 'prompt\_eval\_count': 169, 'prompt\_eval\_duration': 732785 8000, 'eval count': 44, 'eval duration': 11529350000}



```
Out[24]: ('SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM "invoices" i\nGROUP BY i.BillingCountry',
               BillingCountry TotalInvoices
           0
                    Argentina
           1
                                           7
                    Australia
           2
                                           7
                      Austria
           3
                                           7
                      Belgium
           4
                       Brazil
                                           35
           5
                       Canada
                                           56
           6
                        Chile
                                           7
           7
                                           14
               Czech Republic
           8
                                           7
                      Denmark
           9
                                           7
                      Finland
                                           35
           10
                       France
                                           28
           11
                      Germany
           12
                                           7
                      Hungary
           13
                        India
                                           13
           14
                      Ireland
                                           7
           15
                                           7
                        Italy
                                           7
           16
                  Netherlands
           17
                                           7
                       Norway
                                           7
           18
                       Poland
           19
                     Portugal
                                           14
                                           7
           20
                        Spain
                                           7
           21
                       Sweden
           22
                          USA
                                           91
           23 United Kingdom
                                          21,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'BillingCountry=%{x}<br>TotalInvoices=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Argentina', 'Australia', 'Austria', 'Belgium', 'Brazil', 'Canada',
                                      'Chile', 'Czech Republic', 'Denmark', 'Finland', 'France', 'Germany',
                                     'Hungary', 'India', 'Ireland', 'Italy', '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, 14, 7, 7, 91, 21]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Total Invoices per Country'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'BillingCountry'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalInvoices'}}}
          }))
         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" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES "med ia types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeRe portsTo ON "employees" (ReportsTo)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRIMARY KEY A FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n UTOINCREMENT NOT NULL,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Company NVARCHAR(80),\r\n State NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Fax NVARCHAR(24),\r Country NVARCHAR(40),\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 Title NVARCHAR(30),\r\n ReportsTo INTEGER,\r\n L.\r\n FirstName NVARCHAR(20) NOT NULL,\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 mployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\CREATE 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

he most money overall, \n \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 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.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 otal 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 Find the customer with the most invoices \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\nORDER BY TotalInvoices DESC\nLIM IT 1'}, {'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 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 ontent': '\n List all invoices with a total exceeding \$10:\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY K EY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCE tPrice NUMERIC(10,2) 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 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 ER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds I Bvtes INTEGER.\r\n FOREIGN KEY (Album NTEGER NOT NULL,\r\n 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\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 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) 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 State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n  $AR(70), \r\n$ City NVARCHAR(40),\r\n PostalCode  $NVARCHAR(10).\r\n$ Phone NVARCHAR(24).\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n **FOREIGN** KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n) \n\nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\n\===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": Hint: order total can be Find the top 5 customers who spent the most money overall. \n \n 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\nGR0UP 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 Get the total number of invoices for each customer \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"}, {"role": "user", Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELEC T 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 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 Get the average invoice total for each customer:\n"}, {"role": "assistant", "conte nt": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i 0 N c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n List all invoi ces with a total exceeding \$10:\n"}]

Add of existing embedding ID: aea89953-21b2-55d1-9dda-431ee6033c3d-sql Insert of existing embedding ID: aea89953-21b2-55d1-9dda-431ee6033c3d-sql

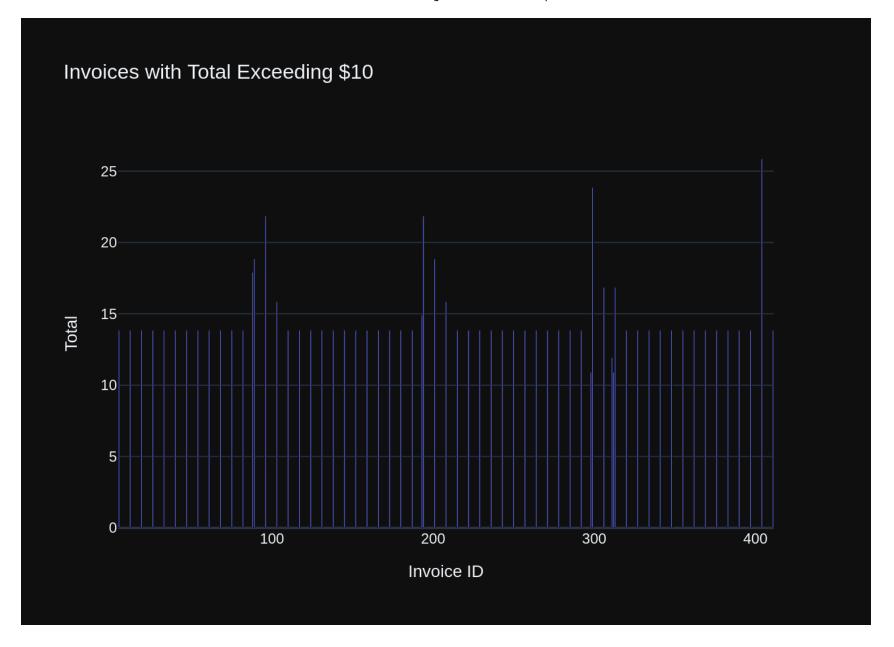
```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:37:17.94656018Z', 'message': {'role': 'assistan
t', 'content': 'SELECT * \nFROM "invoices" \nWHERE Total > 10.00'}, 'done reason': 'stop', 'done': True, 't
otal duration': 87588612562, 'load duration': 1062109, 'prompt eval count': 1874, 'prompt eval duration': 8
1870508000, 'eval count': 20, 'eval duration': 5085360000}
SELECT *
FROM "invoices"
WHERE Total > 10.00
SELECT *
FROM "invoices"
WHERE Total > 10.00
    InvoiceId CustomerId
                                   InvoiceDate
                                                            BillingAddress \
0
            5
                       23 2009-01-11 00:00:00
                                                           69 Salem Street
                        2 2009-02-11 00:00:00
1
           12
                                                   Theodor-Heuss-Straße 34
2
           19
                       40 2009-03-14 00:00:00
                                                           8, Rue Hanovre
3
                       19 2009-04-14 00:00:00
           26
                                                          1 Infinite Loop
4
           33
                       57 2009-05-15 00:00:00
                                                           Calle Lira, 198
          . . .
59
          383
                       10 2013-08-12 00:00:00 Rua Dr. Falcão Filho, 155
60
          390
                       48 2013-09-12 00:00:00
                                                     Lijnbaansgracht 120bg
61
          397
                       27 2013-10-13 00:00:00
                                                           1033 N Park Ave
62
          404
                        6 2013-11-13 00:00:00
                                                             Rilská 3174/6
63
          411
                       44 2013-12-14 00:00:00
                                                           Porthaninkatu 9
   BillingCity BillingState BillingCountry BillingPostalCode Total
                                                         2113 13.86
0
                                        USA
        Boston
                         MA
1
                                                         70174 13.86
     Stuttgart
                       None
                                    Germany
2
         Paris
                       None
                                     France
                                                         75002 13.86
     Cupertino
3
                         CA
                                        USA
                                                         95014 13.86
4
      Santiago
                       None
                                       Chile
                                                          None 13.86
           . . .
                        . . .
                                         . . .
                                                           . . .
     São Paulo
                                                     01007-010 13.86
59
                         SP
                                     Brazil
     Amsterdam
                                Netherlands
                                                          1016 13.86
60
                         ۷V
                                                         85719 13.86
61
                         ΑZ
                                        USA
        Tucson
62
        Prague
                       None
                             Czech Republic
                                                         14300 25.86
63
      Helsinki
                       None
                                    Finland
                                                         00530 13.86
[64 rows x 9 columns]
Ollama parameters:
model=codegemma:latest,
options={},
keep alive=None
```

## Prompt Content:

duration': 18416461000}

[{"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 guery: SELECT \* \nFROM \"invoices\" \nWHERE Total > 10.00\n\nThe followin q is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n InvoiceId int64\nCustomerId int64\nInvoiceDate object\nBillingAddress obiect\nBillinaCit object\nBillingState obiect\nBillinaCountrv object\nBillingPostalCode obiec float64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python t\nTotal plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer wi th any explanations -- just the code."}] Ollama Response: {'model': 'codegemma:latest', 'created at': '2024-06-13T22:37:44.994516855Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as px\n\nfiq = px.bar(df, x='InvoiceId', y='Total', title ='Invoices with Total Exceeding \$10')\n\nfig.update layout(\n xaxis title='Invoice ID',\n yaxis title ='Total',\n)\n\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total duration': 27021081555, 'loa

d duration': 721488, 'prompt eval count': 203, 'prompt eval duration': 8514490000, 'eval count': 72, 'eval



```
Out[25]: ('SELECT * \nFROM "invoices" \nWHERE Total > 10.00',
               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
                                   19 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},
                          'template': '...',
                          'title': {'text': 'Invoices with Total Exceeding $10'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Invoice ID'}},
                          '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 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 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 Id INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\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 > 10.00'}, {'role': 'user', 'content': ' \n 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 Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.Customer Id, 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 top 5 customers who spent the m ost money overall, \n Hint: order total can be found on invoices table, calculation using invoic \n e items detail table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Tota l) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.Customer Id\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.C ustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Cust omerId\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.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 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 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 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 Get the average invoice tot al for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageIn voiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 and the total amount invoiced:\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN

CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL.\r\n Billin 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\t0N 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 UnitPrice NUMERIC(10,2) NO T NULL,\r\n Ouantity INTEGER NOT NULL.\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (Tra ckid) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLineInvoiceId ON \"in voice items\" (InvoiceId)\n\nCREATE INDEX IFK InvoiceCustomerId ON \"invoices\" (CustomerId)\n\nCREATE INDE X IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE TABLE \"employees\"\r\n(\r\n FirstName NVAR d INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n BirthDate DATETIME.\r\n CHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGER.\r\n State NVARCHAR(40), \r\n HireDate DATETIME.\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r Email NVARCHAR(60).\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON D ELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMA RY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT N City NVARCHAR(40),\r\n ULL.\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n State NVARC  $HAR(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n  $NVARCHAR(24), \r\n$ 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 TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n TABLE \"tracks\"\r\n(\r\n Name NVARCHAR(200) AlbumId INTEGER.\r\n GenreId INTEGER,\r\n NOT NULL,\r\n MediaTypeId INTEGER NOT NULL.\r\n oser NVARCHAR(220).\r\n Milliseconds INTEGER NOT NULL.\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 UPDATE NO ACTION.\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\t0N DELETE NO ACTION 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 \"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) PlavlistId INTEGER NOT NULL,\r\n \n\nCREATE TABLE \"playlist track\"\r\n(\r\n TrackId INTEGER NOT N CONSTRAINT 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 (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 SOLite database invoice means order\n\n===Response Guidelines \n1. If the provide d context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular co lumn, please generate an intermediate SQL 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":

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 > 10.00"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "cont ent": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountr y"}, {"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": " 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 Find the customer 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 \"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": " 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": " Find the customer with the most invoices \n"}, {"role": "assistant", "content": "SELECT c.CustomerI d, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.Custo merId\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 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 Get the average invoice total for each customer:\n"}, {"rol e": "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 all invoices since 2010 and the total amount invoiced:\n"}] Ollama Response: {'model': 'codegemma:latest', 'created at': '2024-06-13T22:39:19.136611169Z', 'message': {'role': 'assistan t', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM "invoices" i\nWHERE i'}, 'done reas on': 'stop', 'done': True, 'total duration': 94046624476, 'load duration': 745062, 'prompt eval count': 202 0, 'prompt eval duration': 86711387000, 'eval count': 25, 'eval duration': 6698290000} SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount FROM "invoices" i WHERE i SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount FROM "invoices" i

```
WHERE i
    Couldn't run sql: Execution failed on sql 'SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount
    FROM "invoices" i
    WHERE i': no such column: i

In [27]: question = """
        List all employees and their reporting manager's name (if any):
        """
        vn.ask(question=question)

Number of requested results 10 is greater than number of elements in index 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  $L,\r\n$ LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\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  $0), \r\n$ Phone NVARCHAR(24),\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 Total NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (CustomerId) REFERENCES "cust alCode NVARCHAR(10),\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  $\n(\r\n$ InvoiceId INTEGER NOT NULL.\r UnitPrice NUMERIC(10,2) NOT NULL,\r\n TrackId INTEGER NOT NULL.\r\n Ouantity INTEGER NOT NUL  $L.\r\n$ FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE N FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n ArtistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r Name NVARCHAR(120)\r\n)\n\nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NU LL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n tes INTEGER.\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES "albums" (Al FOREIGN KEY (GenreId) REFERENCES "genres" (G bumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n 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 5 customers who spent the most money overall. \n \n Hint: order total can be found on invoices tabl 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\nJOIN "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 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 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.Custo merId = i.CustomerId\nGROUP BY c.CustomerId'}, {'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': 'user', 'content': " \n List all employees and their reporting manager's name (if any):\n"}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 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 TEGER,\r\n BirthDate DATETIME,\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCH

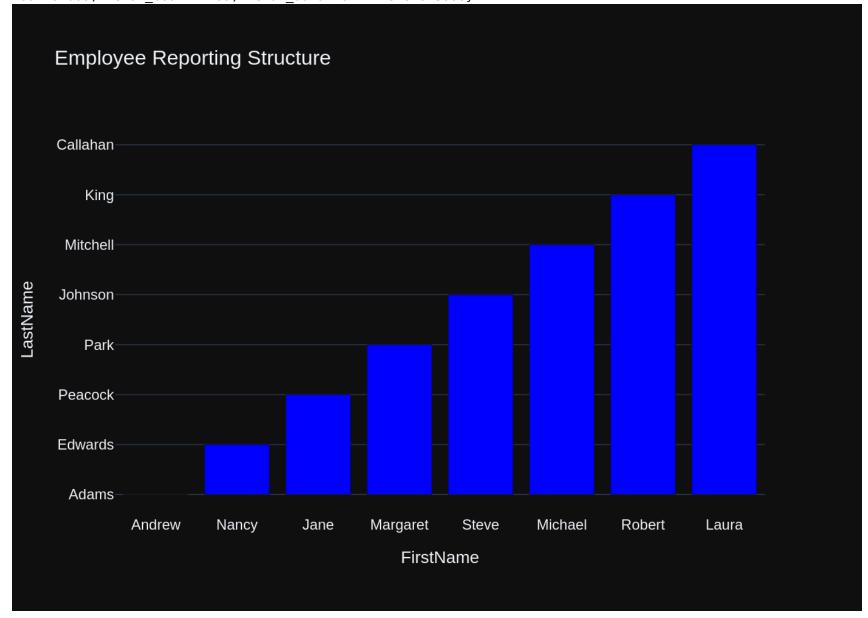
State NVARCHAR(40),\r\n  $AR(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10).\r\n Phon e NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60).\r\n FOREIGN KEY (ReportsTo) REFERENC ES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"custom CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n ers\"\r\n(\r\n FirstName NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(1 0), r nPhone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL.\r\n Support FOREIGN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO RepId INTEGER.\r\n 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 erId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n В 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 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 Ouantity INTEGER FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON NOT NULL,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION UPDATE NO ACTION,\r\n ON UPDATE NO ACTION\r\n)\n\CREATE 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 AlbumId INTEGER,\r\n AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n MediaTypeId INT Milliseconds INTEGER NOT NUL EGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Bvtes INTEGER.\r\n FOREIGN KEY (AlbumId) REFERENCES L.\r\n UnitPrice NUMERIC(10.2) NOT NULL.\r\n \"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 T 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===Add itional Context \n\nIn the SOLite database invoice means order\n\n===Response Guidelines \n1. If the provid ed context is sufficient, please generate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular co lumn, please generate an intermediate SQL query to find the distinct strings in that column. Prepend the qu ery with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the guestion 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 Find the customer with the most invoices \n"}, {"role": "assistant", "content": "S r", "content": " \n 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 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 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 Get the average invoice total for each customer:\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customer s\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "conte Hint: album quantity is found in invoice items, \n \n Find the top 5 customers 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.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 List all employees and their reporting manage r's name (if any):\n"}]

Add of existing embedding ID: fd25ebba-4066-5a0f-8613-7b1c2ace0339-sql Insert of existing embedding ID: fd25ebba-4066-5a0f-8613-7b1c2ace0339-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:40:54.492637981Z', 'message': {'role': 'assistan
t', 'content': 'SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLas
tName\nFROM "employees" e\nLEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId'}, 'done reason': 'sto
p', 'done': True, 'total duration': 95315032160, 'load duration': 803111, 'prompt eval count': 1874, 'promp
t eval duration': 82768583000, 'eval count': 45, 'eval duration': 11921072000}
SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName
FROM "employees" e
LEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId
SELECT e.FirstName, e.LastName, mt.FirstName AS ManagerFirstName, mt.LastName AS ManagerLastName
FROM "employees" e
LEFT JOIN "employees" mt ON e.ReportsTo = mt.EmployeeId
  FirstName LastName ManagerFirstName ManagerLastName
     Andrew
                Adams
                                  None
                                                  None
1
     Nancv Edwards
                                Andrew
                                                Adams
2
                                              Edwards
       Jane
             Peacock
                                 Nancv
3 Margaret
                 Park
                                Nancy
                                              Edwards
4
      Steve Johnson
                                Nancy
                                              Edwards
5
   Michael Mitchell
                               Andrew
                                                Adams
6
     Robert
                Kina
                              Michael
                                             Mitchell
7
      Laura Callahan
                              Michael
                                             Mitchell
Ollama parameters:
model=codegemma:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n List all employees and their reporting manager's name (i
f any):\n'\nThe DataFrame was produced using this query: SELECT e.FirstName, e.LastName, mt.FirstName AS
ManagerFirstName, mt.LastName AS ManagerLastName\nFROM \"employees\" e\nLEFT JOIN \"employees\" mt ON e.Rep
ortsTo = mt.EmployeeId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
df.dtvpes gives:\n FirstName
                                      object\nLastName
                                                                  object\nManagerFirstName
               object\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly co
de to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is
only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any ex
planations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:41:21.210647233Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfiq = px.bar(df, x='FirstName', y='LastName', hov
er name='ManagerFirstName', hover data=['ManagerLastName'],\n
                                                                          title='Employee Reporting Struct
ure')\n\nfig.update traces(marker color='blue')\n\nfig.show()\n```"}, 'done reason': 'stop', 'done': True,
```

'total\_duration': 26690848586, 'load\_duration': 659533, 'prompt\_eval\_count': 173, 'prompt\_eval\_duration': 7894262000, 'eval count': 69, 'eval duration': 18704873000}



```
Out[27]: ('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',
            FirstName LastName ManagerFirstName ManagerLastName
           0
                Andrew
                           Adams
                                             None
                                                             None
           1
                Nancy
                         Edwards
                                           Andrew
                                                            Adams
           2
                 Jane
                         Peacock
                                            Nancy
                                                          Edwards
          3 Margaret
                            Park
                                            Nancy
                                                          Edwards
           4
                Steve
                        Johnson
                                            Nancy
                                                          Edwards
           5
              Michael Mitchell
                                                            Adams
                                           Andrew
               Robert
                            King
                                          Michael
                                                         Mitchell
           7
                Laura Callahan
                                          Michael
                                                         Mitchell,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'customdata': array([[None],
                                              ['Adams'],
                                              ['Edwards'],
                                              ['Edwards'],
                                              ['Edwards'],
                                              ['Adams'],
                                              ['Mitchell'],
                                              ['Mitchell']], dtype=object),
                         'hovertemplate': ('<b>%{hovertext}</b><br>Fir' ... '{customdata[0]}<extra></extra>'),
                         'hovertext': array([None, 'Andrew', 'Nancy', 'Nancy', 'Nancy', 'Andrew', 'Michael',
                                             'Michael'l, dtvpe=object).
                         'legendgroup': '',
                         'marker': {'color': 'blue', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Andrew', 'Nancy', 'Jane', 'Margaret', 'Steve', 'Michael', 'Robert',
                                     'Laura'], dtype=object),
                         'xaxis': 'x',
                         'y': array(['Adams', 'Edwards', 'Peacock', 'Park', 'Johnson', 'Mitchell', 'King',
                                     'Callahan'], dtype=object),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Employee Reporting Structure'},
```

[{'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 Y AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n Price NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERE NCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK InvoiceLin eTrackId ON "invoice items" (TrackId)\n\nCREATE TABLE sqlite stat1(tbl,idx,stat)\n\nCREATE INDEX IFK Custom erSupportRepId ON "customers" (SupportRepId)\n\nCREATE TABLE "customers"\r\n(\r\n CustomerId INTEGER PRI MARY KEY AUTOINCREMENT NOT NULL,\r\n FirstName NVARCHAR(40) NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n  $CHAR(40), \r\n$ Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n  $\times$  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 Address NVARCHAR(70),\r\n State NVARCHAR(40),\r\n eDate DATETIME,\r\n City 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\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 Find the top 5 custome merId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n 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 c.CustomerId, COUNT(i.Invoice

Id) 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 all invoices sinc e 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.To tal) AS TotalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'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 Hint: album quantity is found in invoice items, \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 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.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 Hint: album quantity is found in invoice i 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\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 List all invoices with a total exceeding \$1 0:\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "invoices" \nWHERE Total > 10.00'}, {'role': 'use Get the average invoice total for each customer:\n'}] r', 'content': ' \n Ollama parameters: model=codegemma:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 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

Find the c

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 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 Title NVARCHAR(30).\r\n L.\r\n FirstName NVARCHAR(20) NOT NULL,\r\n ReportsTo INTEGER,\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\" 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\nJ0IN \"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 customer with the most invoices \n"}, {"role": "assistant", "cont ent": "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 all invoices since 2010 and the total amount invoiced:\n"}, {"role": "assista nt", "content": "SELECT i.InvoiceDate, SUM(i.Total) AS TotalAmount\nFROM \"invoices\" i\nWHERE i.InvoiceDat e >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "content": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvo ices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n antity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.TrackI d) 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 5"}, {"r ole": "user", "content": " \n Find the customer who bought the most albums in total quantity (across a

Add of existing embedding ID: 6bed484b-9a80-57f4-ad89-5f775b5df252-sql Insert of existing embedding ID: 6bed484b-9a80-57f4-ad89-5f775b5df252-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:42:47.621940418Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoice
s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, 'done reason': 'stop', 'done': True, 'total d
uration': 86293220440, 'load duration': 717957, 'prompt eval count': 1704, 'prompt eval duration': 74453778
000, 'eval count': 42, 'eval duration': 11216094000}
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
    CustomerId AverageInvoiceTotal
0
             1
                           5.660000
             2
                           5.374286
1
2
             3
                           5.660000
3
             4
                           5.660000
             5
4
                           5.802857
5
             6
                           7.088571
6
             7
                           6.088571
7
             8
                           5.374286
8
             9
                           5.374286
9
            10
                           5.374286
10
            11
                           5.374286
11
            12
                           5.374286
12
            13
                           5.374286
13
            14
                           5.374286
14
            15
                           5.517143
15
            16
                           5.374286
16
            17
                           5.660000
17
            18
                           5.374286
18
            19
                           5.517143
19
            20
                           5.660000
20
            21
                           5.374286
21
            22
                           5.660000
22
            23
                           5.374286
23
            24
                           6.231429
```

6.088571

6.802857

5.374286

24

25

26

25

26

27

27	28	6.231429
28	29	5.374286
29	30	5.374286
30	31	5.374286
31	32	5.374286
32	33	5.374286
33	34	5.660000
34	35	5.374286
35	36	5.374286
36	37	6.231429
37	38	5.374286
38	39	5.517143
39	40	5.517143
40	41	5.374286
41	42	5.660000
42	43	5.802857
43	44	5.945714
44	45	6.517143
45	46	6.517143
46	47	5.374286
47	48	5.802857
48	49	5.374286
49	50	5.374286
50	51	5.517143
51	52	5.374286
52	53	5.374286
53	54	5.374286
54	55	5.374286
55	56	5.374286
56	57	6.660000
57	58	5.517143
58	59	6.106667
011 ama	naramatare	

Ollama parameters:

model=codegemma:latest,

options={},

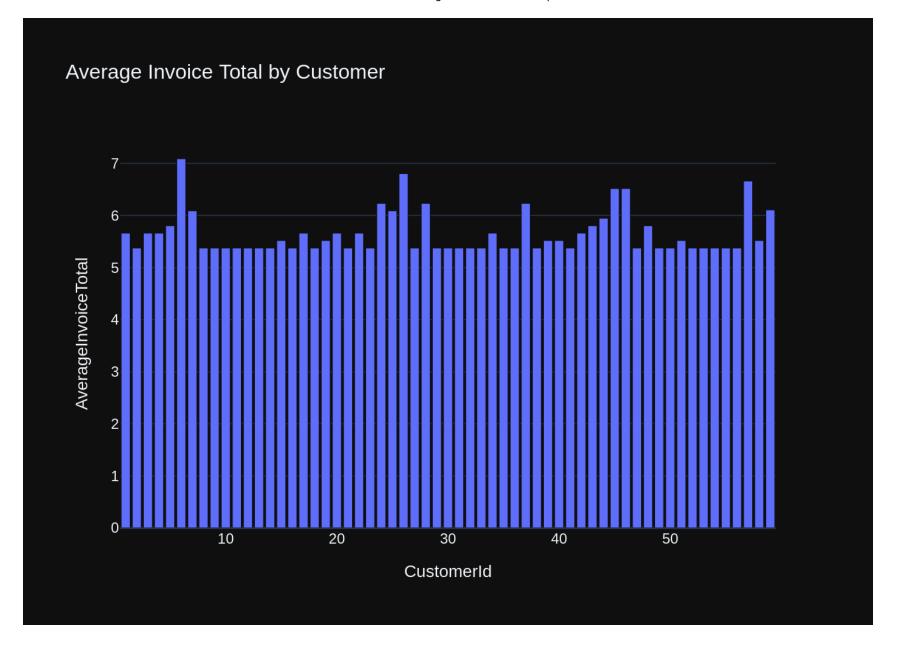
keep alive=None

Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Get the average invoice total for each customer:\n'\n\nT he DataFrame was produced using this query: SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId

int64\nAverageInvoiceTotal float64\ndtype: object"}, {"role": "user", "content": "Can you generate the P ython plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'd f'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not an swer with any explanations -- just the code."}]
Ollama Response:

{'model': 'codegemma:latest', 'created\_at': '2024-06-13T22:43:07.578432644Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as  $px\nfig = px.bar(df, x='CustomerId', y='AverageInvoic eTotal', title='Average Invoice Total by Customer')\nfig.show()\n``"}, 'done_reason': 'stop', 'done': Tr ue, 'total_duration': 19928640860, 'load_duration': 640644, 'prompt_eval_count': 189, 'prompt_eval_duration': 8189889000, 'eval count': 44, 'eval duration': 11648984000}$ 



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',

tome	rId = i.Cust	omerId\nGROUP BY c.Custon
	CustomerId	AverageInvoiceTotal
0	1	5.660000
1	2	5.374286
2	3	5.660000
3	4	5.660000
4	5	5.802857
5	6	7.088571
6	7	6.088571
7	8	5.374286
8	9	5.374286
9	10	5.374286
10	11	5.374286
11	12	5.374286
12	13	5.374286
13	14	5.374286
14	15	5.517143
15	16	5.374286
16	17	5.660000
17	18	5.374286
18	19	5.517143
19	20	5.660000
20	21	5.374286
21	22	5.660000
22	23	5.374286
23	24	6.231429
24	25	6.088571
25	26	6.802857
26	27	5.374286
27	28	6.231429
28	29	5.374286
29	30	5.374286
30	31	5.374286
31	32	5.374286
32	33	5.374286
33	34	5.660000
34	35	5.374286
35	36	5.374286
36	37	6.231429
37	38	5.374286
38	39	5.517143

```
39
            40
                           5.517143
40
            41
                           5.374286
41
            42
                           5.660000
42
            43
                           5.802857
43
            44
                           5.945714
            45
44
                           6.517143
45
            46
                           6.517143
46
            47
                           5.374286
47
            48
                           5.802857
48
            49
                           5.374286
            50
49
                           5.374286
50
            51
                           5.517143
51
            52
                           5.374286
52
            53
                           5.374286
53
            54
                           5.374286
54
            55
                           5.374286
55
            56
                           5.374286
            57
56
                           6.660000
57
            58
                           5.517143
58
            59
                           6.106667,
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([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
                          19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
                          37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
                          55, 56, 57, 58, 59]),
              'xaxis': 'x'.
              'v': array([5.66
                                                         , 5.66
                                     , 5.37428571, 5.66
                                                                          , 5.80285714, 7.08857143,
                          6.08857143, 5.37428571, 5.37428571, 5.37428571, 5.37428571, 5.37428571,
                                                                                     . 5.37428571.
                          5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.66
                          5.51714286. 5.66
                                                 , 5.37428571, 5.66
                                                                      , 5.37428571, 6.23142857,
                          6.08857143, 6.80285714, 5.37428571, 6.23142857, 5.37428571, 5.37428571,
                                                                          , 5.37428571, 5.37428571,
                          5.37428571, 5.37428571, 5.37428571, 5.66
```

```
6.23142857, 5.37428571, 5.51714286, 5.51714286, 5.37428571, 5.66
                                     5.80285714, 5.94571429, 6.51714286, 6.51714286, 5.37428571, 5.80285714,
                                     5.37428571, 5.37428571, 5.51714286, 5.37428571, 5.37428571, 5.37428571,
                                     5.37428571, 5.37428571, 6.66 , 5.51714286, 6.10666667]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Average Invoice Total by Customer'},
                          '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
```

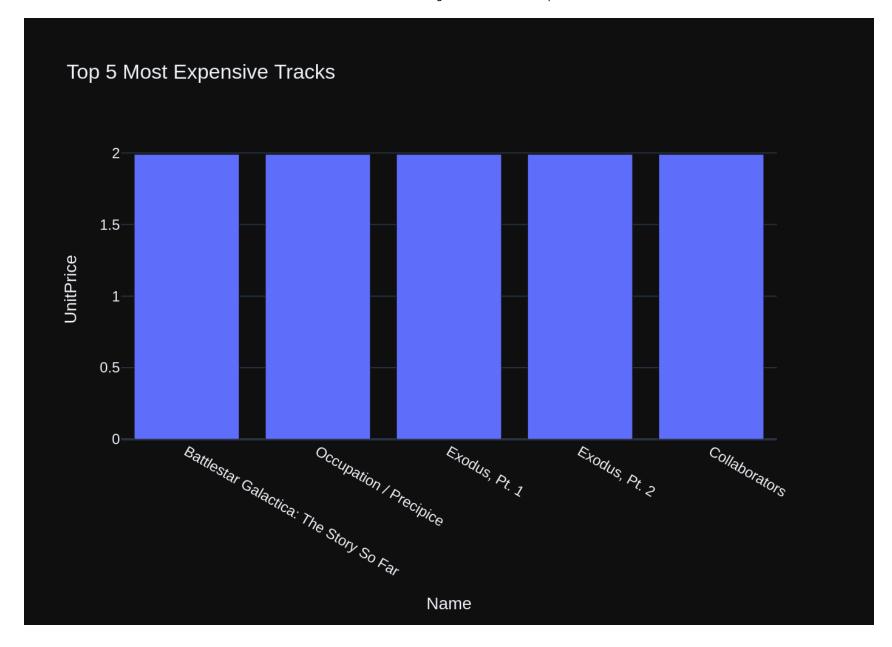
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL guery to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMEN 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 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 X IFK AlbumArtistId ON "albums" (ArtistId)\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 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.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJO IN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMI T 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 Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.Track Id) AS 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 1'}, {'role': Get all playlists containing at least 10 tracks and the total duration of thos 'user', 'content': ' \n e tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Millis econds) 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': ' \n Identify artists who have albums with tracks appearing in multiple ge nres:\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" 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 List all genres and the number of tra cks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nF ROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'content': ' Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'conten t': 'SELECT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': ' \n ind the top 5 most expensive tracks (based on unit price):\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL.\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK Trac kAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX I FK PlaylistTrackTrackId ON \"playlist track\" (TrackId)\n\nCREATE INDEX IFK InvoiceLineTrackId ON \"invoice items\" (TrackId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"invoi

ce items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n 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\t0N DELETE NO AC TION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlaylistId INTEGER NOT NUL CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI TrackId INTEGER NOT NULL.\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UP d), r nFOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON DATE NO ACTION.\r\n 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 SOLite 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.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 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 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": "

Add of existing embedding ID: f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql Insert of existing embedding ID: f33f8cb6-1b12-5ea7-8d9a-aef8166b9970-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:44:36.098204214Z', 'message': {'role': 'assistan
t', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT
5'}, 'done reason': 'stop', 'done': True, 'total duration': 88423453404, 'load duration': 758094, 'prompt e
val count': 1799, 'prompt eval duration': 79604055000, 'eval count': 31, 'eval duration': 8171465000}
SELECT t.TrackId, t.Name, t.UnitPrice
FROM "tracks" t
ORDER BY t.UnitPrice DESC
LIMIT 5
SELECT t.TrackId, t.Name, t.UnitPrice
FROM "tracks" t
ORDER BY t.UnitPrice DESC
LIMIT 5
   TrackId
                                             Name UnitPrice
     2819 Battlestar Galactica: The Story So Far
                                                        1.99
     2820
                            Occupation / Precipice
                                                        1.99
1
                                    Exodus, Pt. 1
2
     2821
                                                        1.99
3
     2822
                                    Exodus, Pt. 2
                                                        1.99
      2823
                                    Collaborators
                                                        1.99
Ollama parameters:
model=codegemma:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Find the top 5 most expensive tracks (based on unit pric
e):\n'\nThe DataFrame was produced using this query: SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks
\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5\n\nThe following is information about the resulting pandas DataFra
me 'df': \nRunning df.dtypes gives:\n TrackId
                                                    int64\nName
                                                                         object\nUnitPrice
pe: 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 data
frame, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the cod
e."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:44:53.983570949Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='UnitPrice', title
='Top 5 Most Expensive Tracks')\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total duration':
17858013273, 'load duration': 628930, 'prompt eval count': 185, 'prompt eval duration': 7458965000, 'eval c
ount': 43, 'eval duration': 10306053000}
```



```
Out[29]: ('SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5',
             TrackId
                                                         Name UnitPrice
                2819 Battlestar Galactica: The Story So Far
          0
                                                                    1.99
                                       Occupation / Precipice
           1
                2820
                                                                    1.99
           2
                2821
                                                Exodus, Pt. 1
                                                                    1.99
           3
                2822
                                                Exodus, Pt. 2
                                                                    1.99
                2823
                                                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'}],
               'lavout': {'barmode': 'relative'.
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Top 5 Most Expensive Tracks'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'UnitPrice'}}}
          }))
         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
```

[{'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 MediaTypeId INTEGER NOT NU T NOT NULL,\r\n 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 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 ArtistId INTEGER NOT NULL,\r\n le NVARCHAR(160) NOT NULL.\r\n FOREIGN KEY (ArtistId) REFERENCES "ar tists" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK AlbumArtistId ON "albums" (ArtistId)\ $n\n===Additional$  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 (g2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'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.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 Find the top

5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.N ame, 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 \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\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.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 "arti sts" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'user', 'content': ' \n Find the customer who bought t he most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.Cu stomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.Custo merId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums D ESC\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.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 Find all tracks with a name containing "What" (case-insen sitive)\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what% \''}, {'role': 'user', 'content': ' \n List all genres and the number of tracks in each genre:\n'}] Ollama parameters: model=codegemma:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 AlbumId INTEGER.\r\n ENT 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 Trac kGenreId ON \"tracks\" (GenreId)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINC REMENT NOT NULL.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist trac k\" (TrackId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE TABLE \"playlists\"\r\n(\r PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120)\r\n)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE TABLE \"playlist track\"\r\n(\r\n PlavlistId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (Playl FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO istId, TrackId).\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE ACTION ON UPDATE NO ACTION,\r\n

NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOI Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n NCREMENT NOT NULL,\r\n FOREIGN KEY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCR EATE INDEX IFK AlbumArtistId ON \"albums\" (ArtistId)\n\n===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 guery with a comment saving intermediate sgl  $\n$ 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 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.Mi lliseconds) AS TotalDuration\nFROM \"playlist track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.Playlis tId\nJOIN \"tracks\" t ON pt.TrackId = t.TrackId\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 pric e):\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 Hint: album quantity is found in invoi Find the top 5 customers who bought the most albums in total quantity (across all inv ce items, \n 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": List all albums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELE CT a.Title, a.ArtistId, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.Ar tistId"}, {"role": "user", "content": " \n Find the customer who bought the most albums in total quant ity (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 1"}, {"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

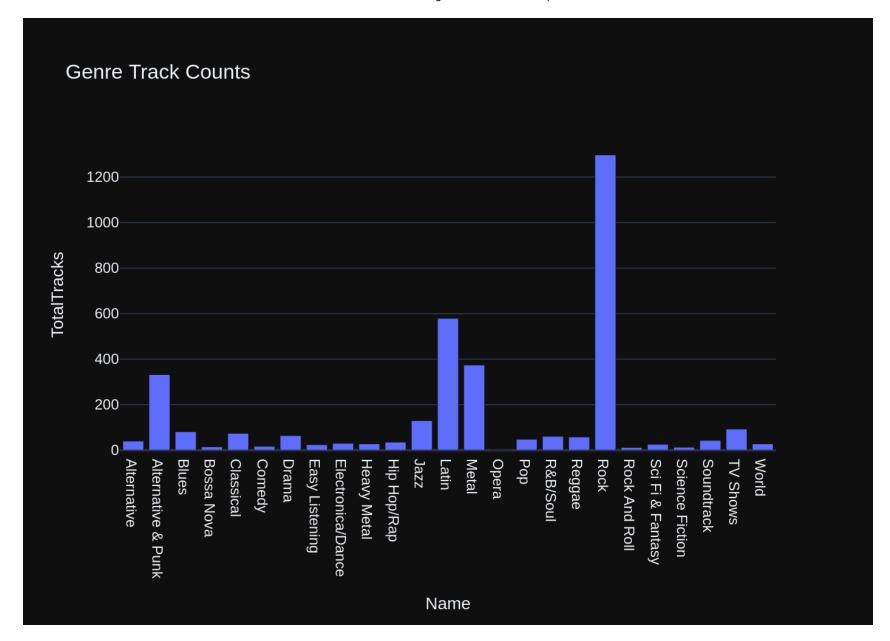
```
items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5"}, {"ro
le": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"},
{"role": "assistant", "content": "SELECT * \nFROM \"tracks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"role":
"user", "content": " \n List all genres and the number of tracks in each genre:\n"}]
```

Add of existing embedding ID: f626b681-4d8f-563a-beee-lea759baaa82-sql Insert of existing embedding ID: f626b681-4d8f-563a-beee-lea759baaa82-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:46:15.309456819Z', 'message': {'role': 'assistan
t', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" g\nJOIN "tracks" t ON g.Genre
Id = t.GenreId\nGROUP BY g.Name'}, 'done reason': 'stop', 'done': True, 'total duration': 81233749327, 'loa
d duration': 702412, 'prompt eval count': 1677, 'prompt eval duration': 69448836000, 'eval count': 44, 'eva
l duration': 11158073000}
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY q.Name
SELECT g.Name, COUNT(t.GenreId) AS TotalTracks
FROM "genres" g
JOIN "tracks" t ON g.GenreId = t.GenreId
GROUP BY g.Name
                  Name TotalTracks
0
           Alternative
                                 40
    Alternative & Punk
                                332
1
2
                 Blues
                                 81
3
                                 15
            Bossa Nova
4
             Classical
                                 74
5
                                 17
                Comedy
6
                                 64
                 Drama
7
                                 24
        Easy Listening
8
     Electronica/Dance
                                 30
9
           Heavy Metal
                                 28
10
                                 35
           Hip Hop/Rap
11
                  Jazz
                                130
12
                                579
                 Latin
13
                                374
                 Metal
14
                 Opera
                                  1
                   Pop
15
                                 48
16
              R&B/Soul
                                 61
17
                                 58
                Reggae
18
                  Rock
                               1297
         Rock And Roll
19
                                 12
20
                                 26
      Sci Fi & Fantasy
21
       Science Fiction
                                 13
22
                                 43
            Soundtrack
23
                                 93
              TV Shows
24
                 World
                                 28
```

Ollama parameters: model=codegemma:latest,

options={}. keep alive=None Prompt Content: [{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n List all genres and the number of tracks in each genr e:\n'\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\n\nThe following is information a bout the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name obiect\nTotalTracks int64\ndtype: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the r esults of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value i n the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- ju st the code."}l Ollama Response: {'model': 'codegemma:latest', 'created at': '2024-06-13T22:46:33.829070714Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='TotalTracks', title ='Genre Track Counts')\n\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total duration': 1849292 3528, 'load duration': 676388, 'prompt eval count': 190, 'prompt eval duration': 8060324000, 'eval count': 41, 'eval duration': 10342276000}



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

```
'Reggae', 'Rock', 'Rock And Roll', 'Sci Fi & Fantasy',
                                   'Science Fiction', 'Soundtrack', 'TV Shows', 'World'], dtype=object),
                        'xaxis': 'x'.
                        'y': array([ 40, 332, 81, 15, 74, 17, 64, 24, 30, 28,
                                                                                              35, 130,
                                    579, 374, 1, 48, 61, 58, 1297, 12, 26, 13, 43, 93,
                                     28]),
                        'yaxis': 'y'}],
              'layout': {'barmode': 'relative',
                        'legend': {'tracegroupgap': 0},
                        'template': '...',
                        'title': {'text': 'Genre Track Counts'},
                         '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 AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n  $(120)\r\n)\n\n\CREATE TABLE "albums"\r\n(\r\n$ 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 List all genres and the number of tracks in each genr e:\n'}, {'role': 'assistant', 'content': 'SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM "genres" q\n JOIN "tracks" t ON q.GenreId = t.GenreId\nGROUP BY q.Name'}, {'role': 'user', 'content': '\n 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 "playl" ist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackI

d\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, {'role': 'user', 'content': ' \n

ind all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'content': 'SELE CT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'role': 'user', 'content': ' \n lbums and their corresponding artist names \n'}, {'role': 'assistant', 'content': 'SELECT a.Title, a.Artis tId, ar.Name AS ArtistName\nFROM "albums" a\nJOIN "artists" ar ON a.ArtistId = ar.ArtistId'}, {'role': 'use Find the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistan r', 'content': ' \n t', 'content': 'SELECT t.TrackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice 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 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.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 Get all genres that do not have any tracks as sociated with them:\n'\l Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 \"tr TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n acks\"\r\n(\r\n Name NVARCHAR(200) NOT NUL L.\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER.\r\n  $ARCHAR(220).\r\n$ Milliseconds INTEGER NOT NULL,\r\n Bvtes INTEGER,\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 0 ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\" (TrackId) \n\nCREATE INDEX IFK TrackMediaTypeId ON \"tracks\" (MediaTypeId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tra cks\" (AlbumId)\n\nCREATE TABLE \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL L.\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN K EY (ArtistId) REFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r ATE TABLE \"playlist track\"\r\n(\r\n

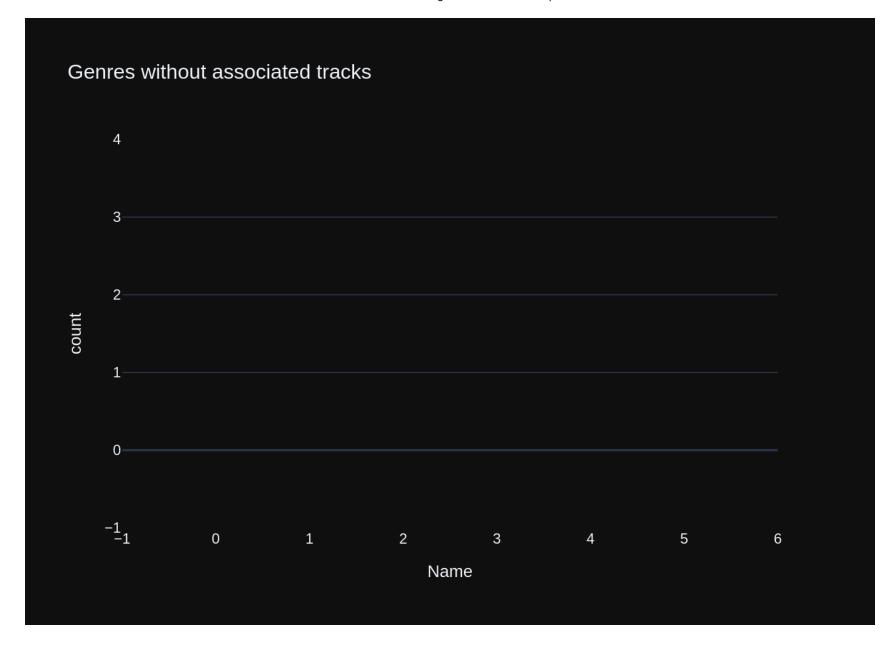
CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFER \n ENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (Track Id) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IF K AlbumArtistId ON \"albums\" (ArtistId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n PlavlistId INTEGER PRIMA Name NVARCHAR(120) $\r\n\n\===Additional$  Context  $\n\n$  the SQLite RY KEY AUTOINCREMENT NOT NULL,\r\n database invoice means order\n\n===Response Guidelines \n1. If the provided context is sufficient, please q enerate a valid SQL query without any explanations for the question. \n2. If the provided context is almost sufficient but requires knowledge of a specific string in a particular column, please generate an intermedi ate SQL query to find the distinct strings in that column. Prepend the query with a comment saying intermed iate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Pleas e use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat th e answer exactly as it was given before. \n"}, {"role": "user", "content": " \n 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 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.Nam e"}, {"role": "user", "content": " \n Get all playlists containing at least 10 tracks and the total du ration of those tracks:\n"}, {"role": "assistant", "content": "SELECT pt.PlaylistId, p.Name AS PlaylistNam e, 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(p t.TrackId) >= 10"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (c ase-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", "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 Find the top 5 m ost expensive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Nam e, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice 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\nJ0IN \"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 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 Get all genres that do not have any tracks associated with them:\n"}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:47:56.878839054Z', 'message': {'role': 'assistan
t', 'content': 'SELECT g.GenreId, g.Name\nFROM "genres" g\nLEFT JOIN "tracks" t ON g.GenreId = t.GenreId\nW
HERE t.GenreId IS NULL;'}, 'done reason': 'stop', 'done': True, 'total duration': 82936127468, 'load durati
on': 695066, 'prompt eval count': 1678, 'prompt eval duration': 71718525000, 'eval count': 42, 'eval durati
on': 10569740000}
SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.GenreId IS NULL;
Output from LLM: SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.GenreId IS NULL;
Extracted SQL: SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.GenreId IS NULL
SELECT g.GenreId, g.Name
FROM "genres" g
LEFT JOIN "tracks" t ON g.GenreId = t.GenreId
WHERE t.GenreId IS NULL
Empty DataFrame
Columns: [GenreId, Name]
Index: []
Ollama parameters:
model=codegemma:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Get all genres that do not have any tracks associated wi
th them:\n'\nThe DataFrame was produced using this query: SELECT q.GenreId, q.Name\nFROM \"genres\" q\nLE
FT JOIN \"tracks\" t ON q.GenreId = t.GenreId\nWHERE t.GenreId IS NULL\n\nThe following is information abou
t the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n GenreId
                                                                              obiect\nName
type: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results o
f the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the da
```

taframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the c ode."}]

Ollama Response:

{'model': 'codegemma:latest', 'created\_at': '2024-06-13T22:48:13.08422513Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', title='Genres without associated tracks')\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 16202935152, 'load\_duration': 768206, 'prompt\_eval\_count': 186, 'prompt\_eval\_duration': 7374055000, 'eval\_count': 37, 'e val duration': 8739586000}



```
Out[31]: ('SELECT q.GenreId, q.Name\nFROM "genres" q\nLEFT JOIN "tracks" t ON q.GenreId = t.GenreId\nWHERE t.GenreI
          d IS NULL',
           Empty DataFrame
           Columns: [GenreId, Name]
           Index: [],
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Name=%{x}<br/>br>count=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                          'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([], dtype=object),
                         'xaxis': 'x',
                         'y': array([], dtype=int64),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Genres without associated tracks'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'count'}}}
           }))
         question = """
In [32]:
             List all customers who have not placed any orders:
         0.00
         vn.ask(question=question)
```

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

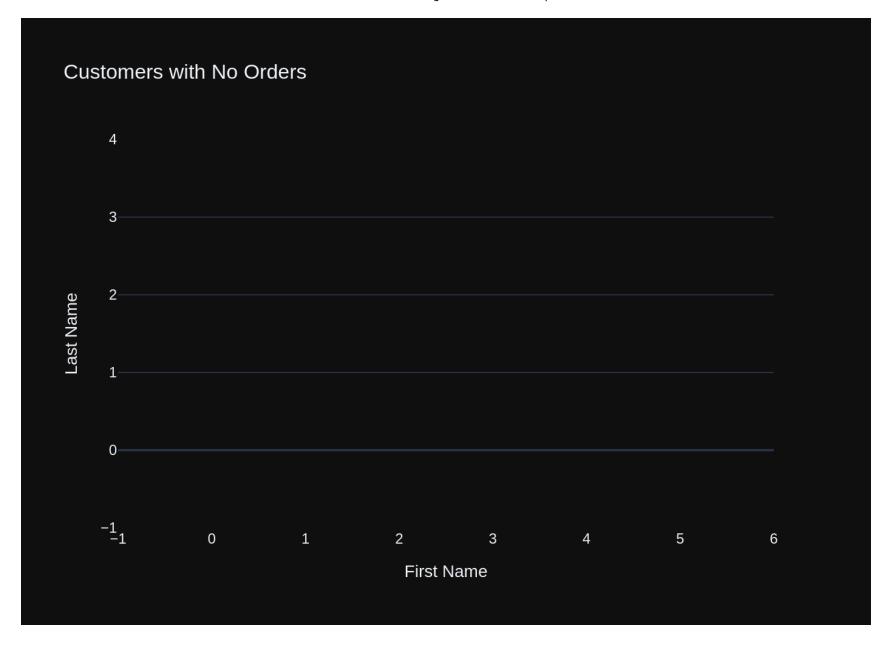
[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE "invoices"\r\n(\r\n InvoiceId INTEGER PRIMARY 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 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 PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(24),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT SupportRepId INTEGER,\r\n FOREIGN KEY (SupportRepId) REFERENCES "employees" (EmployeeId) \r NULL,\r\n \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 LastName NVARCH TABLE "employees"\r\n(\r\n Title NVARCHAR(30),\r\n ReportsTo INTE AR(20) NOT NULL,\r\n FirstName NVARCHAR(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 FOREIGN KEY (ReportsTo) REFERENCES  $NVARCHAR(24), \r\n$ Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n "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 Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT 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 question 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 \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 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', 'content': 'what are the top 5 countries that customers come from?'}, {'role': 'assi stant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nO RDER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Get the total number of invoic es for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS Tota lInvoices\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 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 ontent': '\n List all customers who have not placed any orders:\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN

CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n Billin InvoiceDate DATETIME NOT NULL,\r\n 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 \r\n)\n\nCREATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n LastName NVARCHAR(20) NOT NULL,\r\n FirstName NVARCHAR(40) 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 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 UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL.\r\n R NOT NULL,\r\n FOREIGN KE Y (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\t0N 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 Fax NVARCHAR(24),\r\n FOREIGN KEY (ReportsT Phone NVARCHAR(24),\r\n Email NVARCHAR(60),\r\n 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 C0 NSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums \"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n Title NVARCHAR(160) NOT NULL.\r 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 INDEX IFK CustomerSupportRepId ON \"customers\" (Supp PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL ortRepId)\n\nCREATE TABLE \"playlists\"\r\n(\r\n Name NVARCHAR(120)\r\n)\n\nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n NOT NULL,\r\n GenreId INTEGER.\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 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 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 ist all customers from Canada and their email addresses:\n"}, {"role": "assistant", "content": "SELECT c.Em ail, c.Country\nFROM \"customers\" c\nWHERE c.Country = 'Canada'"}, {"role": "user", "content": "what are t he 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 each customer\n"}, {"role": "assi stant", "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 ma ny customers are there"}, {"role": "assistant", "content": "SELECT COUNT(\*) FROM \"customers\""}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"role": "assistant", "co ntent": "SELECT c.CustomerId, AVG(i.Total) AS AverageInvoiceTotal\nFROM \"customers\" c\nJ0IN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": "\n List all cu stomers who have not placed any orders:\n"}] Ollama Response: {'model': 'codegemma:latest', 'created at': '2024-06-13T22:49:45.187326399Z', 'message': {'role': 'assistan t', 'content': 'SELECT c.CustomerId, c.FirstName, c.LastName\nFROM "customers" c\nLEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId\nWHERE i.InvoiceId IS NULL;'}, 'done reason': 'stop', 'done': True, 'total dura tion': 92006776545, 'load duration': 1103996, 'prompt eval count': 1980, 'prompt eval duration': 8071371200 0, 'eval count': 43, 'eval duration': 10662746000} SELECT c.CustomerId, c.FirstName, c.LastName FROM "customers" c LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId WHERE i.InvoiceId IS NULL;

```
Output from LLM: SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId
WHERE i.InvoiceId IS NULL:
Extracted SQL: SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId
WHERE i.InvoiceId IS NULL
SELECT c.CustomerId, c.FirstName, c.LastName
FROM "customers" c
LEFT JOIN "invoices" i ON c.CustomerId = i.CustomerId
WHERE i.InvoiceId IS NULL
Empty DataFrame
Columns: [CustomerId, FirstName, LastName]
Index: []
Ollama parameters:
model=codegemma:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n List all customers who have not placed any orders:\n'\n
\nThe DataFrame was produced using this query: SELECT c.CustomerId, c.FirstName, c.LastName\nFROM \"custome
rs\" c\nLEFT JOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nWHERE i.InvoiceId IS NULL\n\nThe following
is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n CustomerId
                                      object\ndtype: object"}, {"role": "user", "content": "Can you generat
\nFirstName
                obiect\nLastName
e the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe ca
lled 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do
not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:50:08.734276407Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfiq = px.bar(df, x='FirstName', y='LastName', tit
le='Customers with No Orders')\n\nfig.update layout(xaxis title='First Name', yaxis title='Last Name')\n\nf
ig.show()\n```"}, 'done reason': 'stop', 'done': True, 'total duration': 23544587659, 'load duration': 9208
30, 'prompt eval count': 188, 'prompt eval duration': 7656835000, 'eval count': 63, 'eval duration': 157945
51000}
```



```
Out[32]: ('SELECT c.CustomerId, c.FirstName, c.LastName\nFROM "customers" c\nLEFT JOIN "invoices" i ON c.CustomerId
          = i.CustomerId\nWHERE i.InvoiceId IS NULL',
           Empty DataFrame
           Columns: [CustomerId, FirstName, LastName]
           Index: [],
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'FirstName=%{x}<br>LastName=%{v}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array([], dtype=object),
                         'xaxis': 'x',
                         'y': array([], dtype=object),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Customers with No Orders'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'First Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Last Name'}}}
           }))
         question = """
In [331:
             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 ArtistId INTEGER PRIMARY KEY AUTOINCR ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "artists"\r\n(\r\n 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 PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK P laylistTrack PRIMARY KEY (PlaylistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES "playlists" (Pl aylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES "tracks" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON "tracks"  $(GenreId)\n\nCREATE\ INDEX\ IFK\ PlaylistTrackTrackId\ ON\ "playlist\ track"\ (TrackId)\n\n===Additional\ Context$ \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'}, {'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 c.CustomerId, COU NT(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 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 "artists" 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 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 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': 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.UnitPrice DESC\nLIMIT 5'}, {'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 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': ' \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\nJOIN "invoices" i ON c.Custom erId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'role': 'user', 'conten 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 Tota lDuration\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 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'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"tracks\"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREM ENT NOT NULL.\r\n Name NVARCHAR(200) NOT NULL.\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres \" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums \"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r 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 \"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 PlavlistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(120) $\r\n)\n\n$ CREATE TABLE Name  $NVARCHAR(120)\r\n)\n$ \"genres\"\r\n(\r\n GenreId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\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 (PlaylistId) R EFERENCES \"playlists\" (PlaylistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (T rackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDE X IFK TrackGenreId ON \"tracks\" (GenreId)\n\nCREATE INDEX IFK PlaylistTrackTrackId ON \"playlist track\"  $(TrackId)_n\n\n===Additional\ Context\ \n\nIn\ the\ SQLite\ database\ invoice\ means\ order\n\n===Response\ Guidelin\ \n$ 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 guest 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 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"}, {"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": 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\nJ0IN \"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"}, {"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 \"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 ums and their corresponding artist names \n"}, {"role": "assistant", "content": "SELECT a.Title, a.ArtistI d, ar.Name AS ArtistName\nFROM \"albums\" a\nJOIN \"artists\" ar ON a.ArtistId = ar.ArtistId"}, {"role": "u ser", "content": " \n Hint: album quantity is found in invoice items, \n \n ers who bought the most albums in total quantity (across all invoices):\n"}, {"role": "assistant", "conten t": "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": "assistan t", "content": "SELECT q.Name, COUNT(t.GenreId) AS TotalTracks\nFROM \"genres\" q\nJOIN \"tracks\" t ON q.G enreId = t.GenreId\nGROUP BY q.Name"}, {"role": "user", "content": " \n Find the top 5 most expensive t racks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\n FROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n tomer who bought the most albums in total quantity (across all invoices): \n"}, {"role": "assistant", "cont

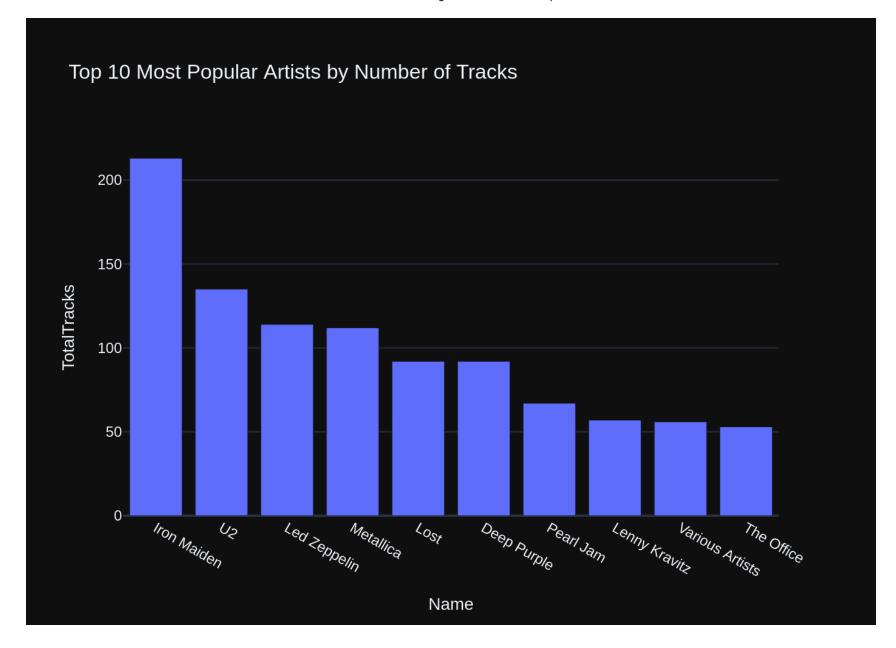
ent": "SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJ0IN \"invoice items\" ii 0N i.InvoiceId = ii.InvoiceId\nGR0UP BY c.CustomerId \nORDER BY TotalAlbums DESC\nLIMIT 1"}, {"role": "user", "content": " \n Find the top 5 customers who \n Hint: order total can be found on invoices table, calculation u spent the most money overall. \n 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 Get all playlis ts containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "conten t": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playlist tr ack\" 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 re are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tr Can you find the top 10 most popular artists based on the number of tracks acks are linked by AlbumId.\n \n"}]

Add of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql Insert of existing embedding ID: 127fd4bd-b9af-539d-9313-1d0234d073b7-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:51:46.333396555Z', 'message': {'role': 'assistan
t', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJ0IN "albums" al ON a.Art
istId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC
\nLIMIT 10'}, 'done reason': 'stop', 'done': True, 'total duration': 97504584705, 'load duration': 678215,
'prompt eval count': 1785, 'prompt eval duration': 78310493000, 'eval count': 71, 'eval duration': 18560680
000}
SELECT a.Name, COUNT(t.TrackId) AS TotalTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.Name
ORDER BY TotalTracks DESC
LIMIT 10
SELECT a.Name, COUNT(t.TrackId) AS TotalTracks
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
JOIN "tracks" t ON al.AlbumId = t.AlbumId
GROUP BY a.Name
ORDER BY TotalTracks DESC
LIMIT 10
             Name TotalTracks
0
      Iron Maiden
                          213
              IJ2
                          135
1
2
     Led Zeppelin
                          114
3
        Metallica
                          112
4
             Lost
                           92
5
      Deep Purple
                           92
6
        Pearl Jam
                           67
7
    Lenny Kravitz
                           57
8 Various Artists
                           56
       The Office
                           53
Ollama parameters:
model=codegemma:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n There are 3 tables: artists, albums and tracks, where alb
ums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n
                                                                                Can you find the top
10 most popular artists based on the number of tracks\n'\n\nThe DataFrame was produced using this query: SE
```

istId\nJOIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n Name object\nTotalTracks 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': 'codegemma:latest', 'created\_at': '2024-06-13T22:52:09.922011487Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Name', y='TotalTracks', title ='Top 10 Most Popular Artists by Number of Tracks')\n\nfig.show()\n``"}, 'done\_reason': 'stop', 'done': Tr ue, 'total\_duration': 23559313956, 'load\_duration': 703828, 'prompt\_eval\_count': 255, 'prompt\_eval\_duration': 10827833000, 'eval count': 49, 'eval duration': 12643232000}



```
Out[33]: ('SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "albums" al ON a.ArtistId = al.Ar
         tistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10',
                         Name TotalTracks
          0
                 Iron Maiden
                                       213
           1
                           IJ2
                                       135
           2
                Led Zeppelin
                                       114
           3
                   Metallica
                                       112
           4
                                        92
                         Lost
           5
                 Deep Purple
                                        92
           6
                                        67
                    Pearl Jam
           7
                                        57
               Lenny Kravitz
            Various Artists
                                        56
                  The Office
                                        53,
           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(['Iron Maiden', 'U2', 'Led Zeppelin', 'Metallica', 'Lost', 'Deep Purple',
                                     'Pearl Jam', 'Lenny Kravitz', 'Various Artists', 'The Office'],
                                    dtype=object),
                         'xaxis': 'x',
                         'y': array([213, 135, 114, 112, 92, 92, 67, 57, 56, 53]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Top 10 Most Popular Artists by Number of Tracks'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Name'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalTracks'}}}
          }))
         question = """
In [34]:
              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 BillingCountry NVARCHAR(40),\r\n 0), r nBillingPostalCode NVARCHAR(10).\r\n Total NUMERIC(10.2) 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': '\nFind the customer with the most invoices \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT

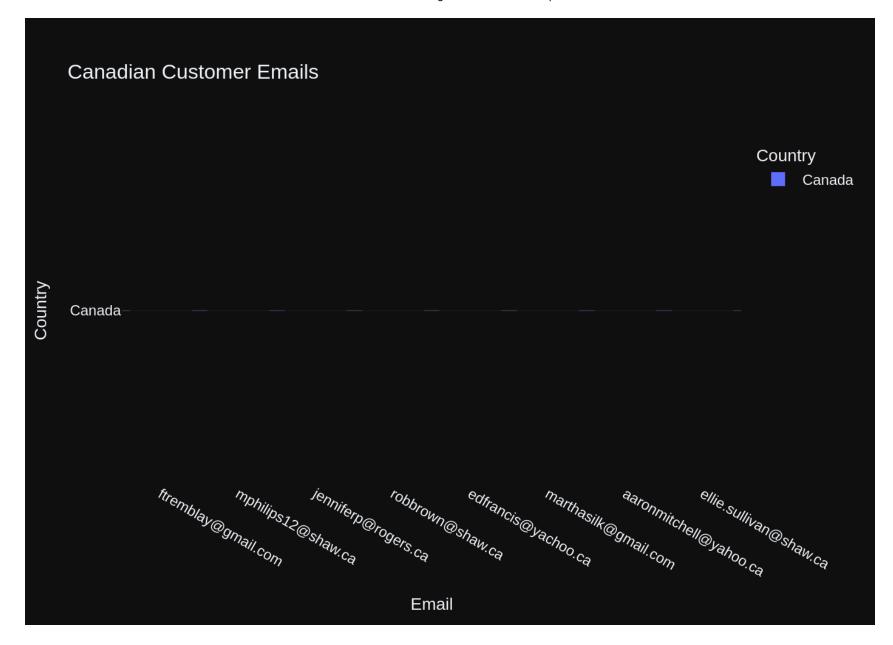
(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, {'role': 'user', 'content': ' \n Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(i.In voiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.Track Id) AS 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 1'}, {'role': 'user', 'content': '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 over 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 TotalSpen t\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY To talSpent DESC\nLIMIT 5'}, {'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 Hint: album quantity is found in in voice 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\nFR OM "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': '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 customers from Canada and their email addresses:\n'}] Ollama parameters: model=codegemma:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE INDEX IFK CustomerSupportRepId ON \"customers\" (SupportRepId)\n\nC REATE TABLE \"customers\"\r\n(\r\n CustomerId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n FirstNa me NVARCHAR(40) NOT NULL,\r\n LastName NVARCHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Addr ess NVARCHAR(70),\r\n City NVARCHAR(40),\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r\n PostalCode NVARCHAR(10).\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 EGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME

BillingCity NVARCHAR(40),\r\n NOT NULL,\r\n BillingAddress NVARCHAR(70).\r\n BillingState NVARCHAR BillingPostalCode NVARCHAR(10),\r\n (40), r nBillingCountry 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 \"employees\"\r\n(\r\n EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n LastName NVARCHAR(2 0) NOT NULL,\r\n FirstName NVARCHAR(20) NOT NULL,\r\n Title NVARCHAR(30),\r\n ReportsTo INTEGE BirthDate DATETIME,\r\n R, r nHireDate 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 tems\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION UPDATE NO ACTION.\r\n ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE sglite sequence(name,seg)\n\nCREATE TABLE \"playlist track\"\r\n  $(\r\n$ PlavlistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) PRIMARY KEY (PlaylistId, TrackId),\r\n \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON \"employee s\" (ReportsTo)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NUL  $L.\r\n$ Title NVARCHAR(160) NOT NULL.\r\n ArtistId INTEGER NOT NULL,\r\n FOREIGN KEY (ArtistId) R EFERENCES \"artists\" (ArtistId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\n===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 SOL 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, 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"}, Find the customer with the most invoices \n"}, {"role": "assistant", {"role": "user", "content": " \n "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 Get the total number of invoices for each customer\n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invo ices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n d 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": "How many customers are ther e"}, {"role": "assistant", "content": "SELECT COUNT(\*) FROM \"customers\""}, {"role": "user", "content": " 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 Find the total number of invoices per country:\n"}, {"role": "assistant", "conten r", "content": " \n t": "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 \n op 5 customers 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 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 quantit y (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.InvoiceLineI 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 List all customer s from Canada and their email addresses:\n"}]

Add of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql Insert of existing embedding ID: 584873f8-1904-50f1-8f80-7ccf08059264-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:53:33.792084896Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\''}, 'done reaso
n': 'stop', 'done': True, 'total duration': 83738425537, 'load duration': 723557, 'prompt eval count': 177
8, 'prompt eval duration': 76853445000, 'eval count': 24, 'eval duration': 6207894000}
SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada'
SELECT c.Email, c.Country
FROM "customers" c
WHERE c.Country = 'Canada'
                    Email Country
0
      ftremblay@gmail.com Canada
1
      mphilips12@shaw.ca Canada
2
      jenniferp@rogers.ca Canada
3
         robbrown@shaw.ca Canada
4
     edfrancis@yachoo.ca Canada
5
    marthasilk@gmail.com Canada
6 aaronmitchell@yahoo.ca Canada
7 ellie.sullivan@shaw.ca Canada
Ollama parameters:
model=codegemma:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n List all customers from Canada and their email addresse
s:\n'\nThe DataFrame was produced using this query: SELECT c.Email, c.Country\nFROM \"customers\" c\nWHER
E c.Country = 'Canada'\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning
                                                object\ndtype: object"}, {"role": "user", "content": "Can
df.dtvpes gives:\n Email
                             object\nCountry
you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas d
ataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Pyth
on code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:53:52.826971592Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='Email', y='Country', color='C
ountry', title='Canadian Customer Emails')\n\nfig.show()\n```"}, 'done reason': 'stop', 'done': True, 'tota
l duration': 19008008243, 'load duration': 759889, 'prompt eval count': 165, 'prompt eval duration': 729427
2000, 'eval count': 44, 'eval duration': 11582078000}
```



```
Out[34]: ('SELECT c.Email, c.Country\nFROM "customers" c\nWHERE c.Country = \'Canada\'',
                               Email Country
                ftremblay@gmail.com Canada
          0
          1
                 mphilips12@shaw.ca Canada
          2
                jenniferp@rogers.ca Canada
          3
                    robbrown@shaw.ca Canada
          4
                edfrancis@yachoo.ca Canada
               marthasilk@gmail.com Canada
          6 aaronmitchell@yahoo.ca Canada
          7 ellie.sullivan@shaw.ca Canada,
          Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'Country=%{y}<br>Email=%{x}<extra></extra>',
                         'legendgroup': 'Canada',
                         'marker': {'color': '#636efa', 'pattern': {'shape': ''}},
                         'name': 'Canada',
                         'offsetgroup': 'Canada',
                         'orientation': 'v',
                         'showlegend': True,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['ftremblay@gmail.com', 'mphilips12@shaw.ca', 'jenniferp@rogers.ca',
                                     'robbrown@shaw.ca', 'edfrancis@yachoo.ca', 'marthasilk@gmail.com',
                                     'aaronmitchell@yahoo.ca', 'ellie.sullivan@shaw.ca'], dtype=object),
                         'xaxis': 'x',
                         'y': array(['Canada', 'Canada', 'Canada', 'Canada', 'Canada', 'Canada',
                                     'Canada'l, dtvpe=object),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'title': {'text': 'Country'}, 'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Canadian Customer Emails'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Email'}},
                          'yaxis': {'anchor': 'x',
                                    'categoryarray': [Canada],
                                    'categoryorder': 'array',
                                    'domain': [0.0, 1.0],
                                    'title': {'text': 'Country'}}}
          }))
         question = """
In [35]:
              Find the customer with the most invoices
```

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

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL 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 Y AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER 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 AlbumId INTEGER,\r\n GenreId INTEGER,\r\n (200) NOT NULL.\r\n MediaTypeId INTEGER NOT NULL.\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 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\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

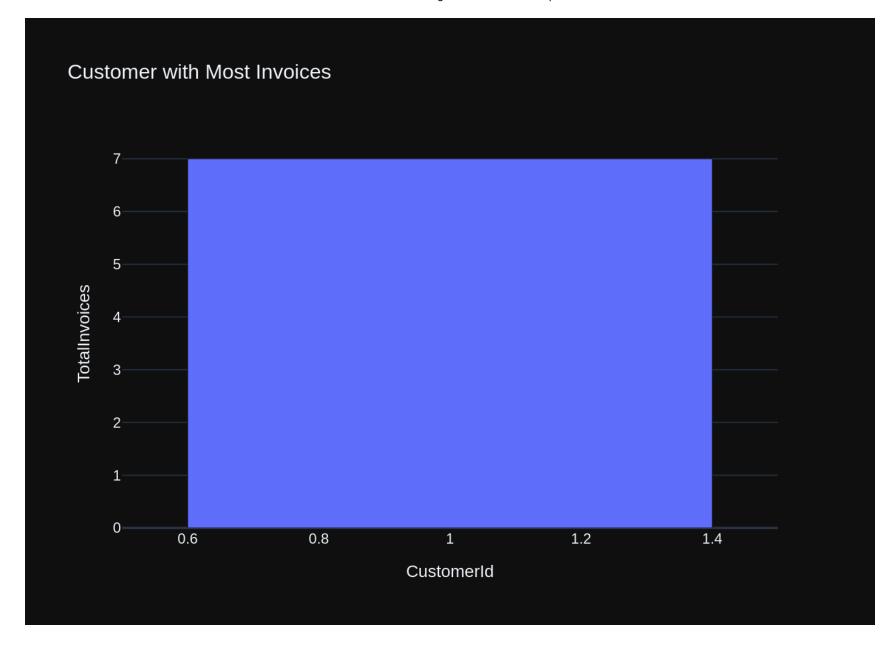
erall, \n Hint: order total can be found on invoices table, calculation using invoice items deta il table is unnecessary \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, SUM(i.Total) AS TotalS pent\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER 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 \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 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 invoices with a total exceeding \$10:\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "invoi ces" \nWHERE Total > 10.00'}, {'role': 'user', 'content': ' \n Find the total number of invoices per co untry:\n'}, {'role': 'assistant', 'content': 'SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM "inv oices" i\nGROUP BY i.BillingCountry'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 an d the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS T otalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'u ser', 'content': ' \n Find the customer with the most invoices \n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN CREMENT NOT NULL,\r\n CustomerId INTEGER NOT NULL,\r\n InvoiceDate DATETIME NOT NULL,\r\n Billin 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 ntrv 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) FOREIGN KEY (Tr REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n 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 Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER,\r\n 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 thDate DATETIME.\r\n HireDate DATETIME,\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40),\r\n St Country NVARCHAR(40),\r\n Phone NVARCHAR(24),\r ate NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n FOREIGN KEY (ReportsTo) REFERENCES \"employees\" Fax NVARCHAR(24),\r\n Email NVARCHAR(60),\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 NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER,\r\n MediaTypeId INTEGER NOT NUL  $L.\r\n$ GenreId INTEGER.\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL.\r\n 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\" FOREIGN KEY (MediaTypeId) REFERENCES \"me (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION.\r\n 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 guery 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 Find the t 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 Get the total number of invoices for each customer\n"}, {"role": "assistan t", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invo ices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"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 \nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n List all invoices with a total exceeding \$1 0:\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"invoices\" \nWHERE Total > 10.00"}, {"role": "u ser", "content": " \n Find the total number of invoices per country:\n"}, {"role": "assistant", "conten t": "SELECT i.BillingCountry, COUNT(\*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.BillingCountry"}, {"role": "user", "content": " \n Find all invoices since 2010 and the total amount invoiced:\n"}, {"rol e": "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": " \n Find the cus tomer with the most invoices \n"}]

Add of existing embedding ID: 3013d1b4-feb2-519d-bfb9-114500436e3d-sql Insert of existing embedding ID: 3013d1b4-feb2-519d-bfb9-114500436e3d-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:55:29.88488508Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM "customers" c\nJOIN "invoice
s" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1'}, 'done r
eason': 'stop', 'done': True, 'total duration': 96948971632, 'load duration': 748683, 'prompt eval count':
1871, 'prompt eval duration': 82554741000, 'eval count': 52, 'eval duration': 13731415000}
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1
SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c.CustomerId
ORDER BY TotalInvoices DESC
LIMIT 1
   CustomerId TotalInvoices
           1
Ollama parameters:
model=codegemma:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Find the customer with the most invoices \n'\n\nThe Dat
aFrame was produced using this query: SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFROM \"cust
omers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoice
s DESC\nLIMIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dty
pes gives:\n CustomerId
                             int64\nTotalInvoices int64\ndtype: object"}, {"role": "user", "content":
"Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pan
das dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only
Python code. Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:55:51.931418411Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfiq = px.bar(\n df,\n x='CustomerId',\n
y='TotalInvoices',\n title='Customer with Most Invoices',\n)\n\fig.show()\n``"}, 'done reason': 'sto
p', 'done': True, 'total duration': 22019543816, 'load duration': 797435, 'prompt eval count': 197, 'prompt
eval duration': 8399675000, 'eval count': 53, 'eval duration': 13489103000}
```



```
Out[35]: ('SELECT 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',
              CustomerId TotalInvoices
           0
                       1
                                      7,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'CustomerId=%{x}<br>TotalInvoices=%{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([7]),
                         'yaxis': 'y'}],
               'layout': {'barmode': 'relative',
                          'legend': {'tracegroupgap': 0},
                          'template': '...',
                          'title': {'text': 'Customer with Most Invoices'},
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                          'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'TotalInvoices'}}}
          }))
 In [ ]:
```

## Advanced SQL questions

[{'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\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  $L,\r\n$ TrackId INTEGER NOT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Ouantity INTEGER NOT FOREIGN KEY (InvoiceId) REFERENCES "invoices" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDAT NULL,\r\n 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 nd 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 5 customers who spent the most money overall, \n \n Hint: order total can be found on invoices tabl 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 re are 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tr acks 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 Find all invoices since 2010 and the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS Tot alAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'use r', 'content': '\n Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'cont 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 ost expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Nam e, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'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 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'}, {'ro le': 'user', 'content': ' \n Find the customer who bought the most albums in total quantity (across al l invoices): \n'}] Ollama parameters: model=codegemma:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 TABLE \"invoic InvoiceLineId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n e items\"\r\n(\r\n InvoiceId INTEGER N

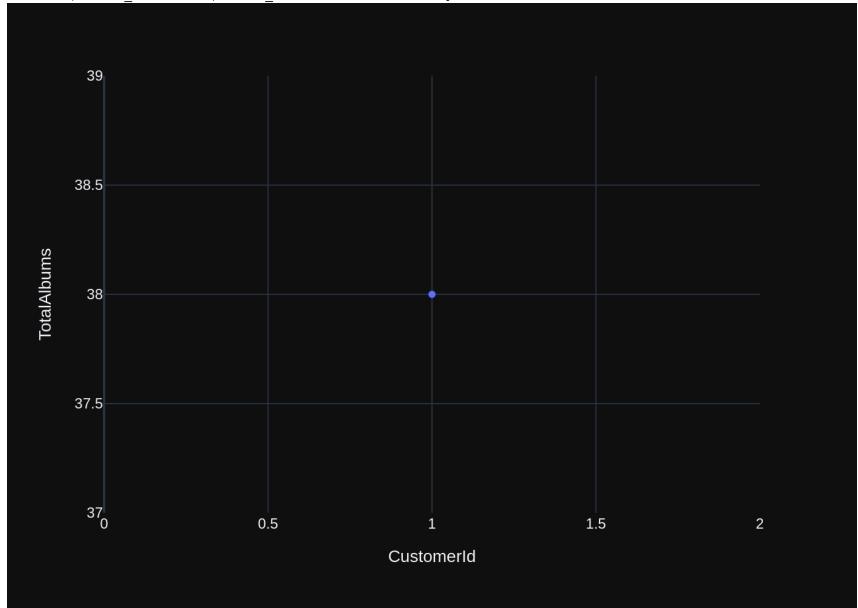
TrackId INTEGER NOT NULL,\r\n OT NULL,\r\n UnitPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGE FOREIGN KEY (InvoiceId) REFERENCES \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION R NOT NULL,\r\n ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO 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 BillingState NVARCHAR(4 LL,\r\n BillingAddress NVARCHAR(70).\r\n BillingCity NVARCHAR(40),\r\n 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 SOLite 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 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 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 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 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 \"c ustomers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpen t DESC\nLIMIT 5"}, {"role": "user", "content": "\n There are 3 tables: artists, albums and tracks, wher

e 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.N ame, COUNT(t.TrackId) AS TotalTracks\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nJ OIN \"tracks\" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY TotalTracks DESC\nLIMIT 10"}, {"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.In voiceDate >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"role": "user", "content": " \n Get the total num ber of invoices for each customer\n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.Invoi ceId) AS TotalInvoices\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId"}, {"role": "user", "content": " \n Find the top 5 most expensive tracks (based on unit pr ice):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDE R BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n Identify artists who have albums wi th tracks appearing in multiple genres:\n\n\n"}, {"role": "assistant", "content": "SELECT a.ArtistId, a.Nam e 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 q 2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name"}, {"role": "user", "content": " Find the customer who bought the most albums in total quantity (across all invoices): \n"}]

Add of existing embedding ID: d1d70c18-f5d9-5970-a32c-914deeca1087-sql Insert of existing embedding ID: d1d70c18-f5d9-5970-a32c-914deeca1087-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:57:29.060958875Z', 'message': {'role': 'assistan
t', '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.Custome
rId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, 'done reason': 'stop', 'done': True, 'total duration': 970389169
04, 'load duration': 832273, 'prompt eval count': 1820, 'prompt eval duration': 78017885000, 'eval count':
70, 'eval duration': 18332291000}
SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums
FROM "customers" c
JOIN "invoices" i ON c.CustomerId = i.CustomerId
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 1
   CustomerId TotalAlbums
            1
Ollama parameters:
model=codegemma:latest,
options={}.
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Find the customer who bought the most albums in total q
uantity (across all invoices): \n'\n\nThe DataFrame was produced using this guery: SELECT c.CustomerId, COU
NT(ii.TrackId) AS TotalAlbums\nFROM \"customers\" c\nJOIN \"invoices\" i ON c.CustomerId = i.CustomerId\nJO
IN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLI
MIT 1\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes give
                                         int64\ndtype: object"}, {"role": "user", "content": "Can you gene
s:\n CustomerId
                    int64\nTotalAlbums
rate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe
called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code.
Do not answer with any explanations -- just the code."}]
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T22:57:59.45195276Z', 'message': {'role': 'assistan
t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='CustomerId', y='TotalAlbums',
title='Customer Album Purchases')\n\nif len(df) == 1:\n fig.add trace(px.indicator(value=df['TotalAlbum
```

s'].iloc[0], title='Total Albums Purchased'))\n\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 't otal\_duration': 30363457561, 'load\_duration': 757730, 'prompt\_eval\_count': 224, 'prompt\_eval\_duration': 9353209000, 'eval count': 80, 'eval duration': 20880106000}



```
Out[36]: ('SELECT c.CustomerId, COUNT(ii.TrackId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.Custom
          erId = i.CustomerId\nJOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER B
          Y TotalAlbums DESC\nLIMIT 1'.
              CustomerId TotalAlbums
           0
                       1
                                   38,
           Figure({
               'data': [{'hovertemplate': 'CustomerId=%{x}<br>TotalAlbums=%{y}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': '#636efa', 'symbol': 'circle'},
                         'mode': 'markers',
                         'name': '',
                         'orientation': 'v',
                         'showlegend': False,
                         'type': 'scatter',
                         'x': array([1]),
                         'xaxis': 'x',
                         'y': array([38]),
                         'yaxis': 'y'}],
               'layout': {'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
                          'template': '...',
                          'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'CustomerId'}},
                          'vaxis': {'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):
         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 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 GenreId INTEGER,\r\n bumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n Composer NVARCHAR(22 Milliseconds INTEGER NOT NULL,\r\n 0), r nBytes 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 $ION, \r\n$ FOREIGN KEY (MediaTypeId) REFERENCES "media types" (MediaTypeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE "albums"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMENT NOT NU 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 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 \$ 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 er', 'content': ' \n Find the customer who bought the most albums in total quantity (across all invoic es): \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': ' \n \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 unnecessary \n'}, {'role': 'assistant', 'co ntent': 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent\nFROM "customers" c\nJOIN "invoices" i ON c.Custom erId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalSpent DESC\nLIMIT 5'}, {'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\nFR0 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 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 Find the top 5 most expensiv e tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.TrackId, t.Name, t.UnitPric e\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Identify ar tists 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\nW HERE a.ArtistId IN (\n SELECT q2.GenreId\n FROM "genres" q2\n JOIN "tracks" t2 ON q2.GenreId = t2.GenreI d\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'use r', 'content': '\n Get the total number of invoices for each customer\n'}, {'role': 'assistant', 'cont 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 Find the total n umber 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 lbum 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'}] Ollama parameters: model=codegemma:latest, options={}, keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo rmat instructions. \n===Tables \nCREATE TABLE \"invoice items\"\r\n(\r\n InvoiceLineId INTEGER PRIMARY K EY AUTOINCREMENT NOT NULL,\r\n InvoiceId INTEGER NOT NULL,\r\n
TrackId INTEGER NOT NULL,\r\n tPrice NUMERIC(10,2) NOT NULL,\r\n Quantity INTEGER NOT NULL,\r\n FOREIGN KEY (InvoiceId) REFERENCE S \"invoices\" (InvoiceId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (TrackId) RE FERENCES \"tracks\" (TrackId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"tracks \"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(200) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(22 Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n 0),\r\n UnitPrice NUMERIC(10,2) NOT NUL

FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO AC L.\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO  $TION, r\n$ ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeId) \r\n\t\tON DELETE NO ACTI ON ON UPDATE NO ACTION\r\n)\n\nCREATE TABLE \"albums\"\r\n(\r\n AlbumId INTEGER PRIMARY KEY AUTOINCREMEN T NOT NULL,\r\n Title NVARCHAR(160) NOT NULL,\r\n ArtistId INTEGER NOT NULL,\r\n 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 nInvoiceDate DATETIME NOT NULL,\r\n BillingAddress NVARCHAR(70),\r\n BillingCity NVARCHAR(4 BillingCountry NVARCHAR(40),\r\n BillingPostalCode NVARCHAR 0), r nBillingState NVARCHAR(40),\r\n (10), r nTotal 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 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 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 guestion 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 Find the customer who bought the most albums in tota l quantity (across all invoices): \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(ii.Tra ckId) 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 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 There are 3 tables: artists, albums and tracks, where albu ms and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 1 0 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"}, {"role": "u
ser", "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.C
ustomerId = i.CustomerId\nGROUP BY c.CustomerId\nORDER BY TotalInvoices DESC\nLIMIT 1"}, {"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.UnitPrice DESC\nLIMIT 5"},
{"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple ge
nres:\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 \"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"}, {"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.CustomerI
d"}, {"role": "user", "content": " \n Find the total number of invoices per country:\n"}, {"role": "ass
istant", "content": "SELECT i.BillingCountry, COUNT(*) AS TotalInvoices\nFROM \"invoices\" i\nGROUP BY i.Bi
llingCountry"}, {"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': 'codegemma:latest', 'created at': '2024-06-13T22:59:35.63141622Z', 'message': {'role': 'assistan
t', 'content': 'SELECT c.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums\nFROM "invoices" i\nJOIN "invoi
ce items" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, 'd
one reason': 'stop', 'done': True, 'total duration': 96082450718, 'load duration': 829857, 'prompt eval cou
nt': 1808, 'prompt eval duration': 80020504000, 'eval count': 57, 'eval duration': 15375800000}
SELECT c.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums
FROM "invoices" i
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
SELECT c.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums
FROM "invoices" i
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5
Couldn't run sql: Execution failed on sql 'SELECT c.CustomerId, COUNT(ii.InvoiceLineId) AS TotalAlbums
FROM "invoices" i
JOIN "invoice items" ii ON i.InvoiceId = ii.InvoiceId
GROUP BY c.CustomerId
ORDER BY TotalAlbums DESC
LIMIT 5': no such column: c.CustomerId
```

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

In [38]: question = """
    Find the top 5 customers who spent the most money overall,

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

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 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 Phone NV  $0), \r\n$ State NVARCHAR(40),\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 Bytes INTEGER,\r\n  $0), \r\n$ Milliseconds INTEGER NOT NULL,\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  $\n)\n\nCREATE INDEX IFK EmployeeReportsTo ON "employees" (ReportsTo)\n\n\n===Additional Context <math>\n\n$  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.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 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 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': ' \n Get the average invoice total for each customer:\n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerI d, AVG(i.Total) AS AverageInvoiceTotal\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerI d\nGROUP BY c.CustomerId'}, {'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\nF ROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nGROUP BY c.CustomerId'}, {'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': 'what are the top 5 countries that customers come from?'}, {'role': 'assis tant', 'content': 'SELECT c.Country, COUNT(\*) AS TotalCustomers\nFROM "customers" c\nGROUP BY c.Country\nOR DER BY TotalCustomers DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find all invoices since 2010 an d the total amount invoiced:\n'}, {'role': 'assistant', 'content': 'SELECT i.InvoiceDate, SUM(i.Total) AS T otalAmount\nFROM "invoices" i\nWHERE i.InvoiceDate >= \'2010-01-01\'\nGROUP BY i.InvoiceDate'}, {'role': 'u ser', 'content': '\n Find the top 5 customers who spent the most money overall, \n rder total can be found on invoices table, calculation using invoice items detail table is unnecessary \n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL query to answer the question. Your response should ONLY be based on the given context and follow the response guidelines and fo

rmat instructions. \n===Tables \nCREATE TABLE \"invoices\"\r\n(\r\n InvoiceId INTEGER PRIMARY KEY AUTOIN 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 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 \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 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 LastName NVAR CHAR(20) NOT NULL,\r\n Company NVARCHAR(80),\r\n Address NVARCHAR(70),\r\n City NVARCHAR(40).\r\n State NVARCHAR(40),\r\n Country NVARCHAR(40),\r\n PostalCode NVARCHAR(10),\r\n Phone NVARCHAR(2 4),\r\n Fax NVARCHAR(24),\r\n Email NVARCHAR(60) NOT NULL,\r\n SupportRepId INTEGER.\r\n FOREI GN KEY (SupportRepId) REFERENCES \"employees\" (EmployeeId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION EmployeeId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL.\r\n \r\n)\n\nCREATE TABLE \"employees\"\r\n(\r\n FirstName NVARCHAR(20) NOT NULL,\r\n LastName 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 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\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) NOT NULL.\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL.\r\n GenreId INTEGER.\r\n Comp Milliseconds INTEGER NOT NULL,\r\n oser NVARCHAR(220),\r\n Bytes INTEGER,\r\n UnitPrice NUMERIC(1 0,2) NOT NULL,\r\n FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DELETE NO ACTION 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 TrackId INTEGER NOT NULL.\r\n 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 query with a comment saying intermediate sql \n3. If the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and 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 \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 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 0N 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.InvoiceLineId) AS TotalAlbum s\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 Find the customer with the most in voices \n"}, {"role": "assistant", "content": "SELECT c.CustomerId, COUNT(i.InvoiceId) AS TotalInvoices\nFR OM \"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\n JOIN \"invoice items\" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\n LIMIT 1"}, {"role": "user", "content": " \n Get the average invoice total for each customer:\n"}, {"rol e": "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": 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 most expensi ve tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitPri ce\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": "what are the top 5 countries that customers come from?"}, {"role": "assistant", "content": "SELECT c.Country, COUNT(\*) AS To talCustomers\nFROM \"customers\" c\nGROUP BY c.Country\nORDER BY TotalCustomers DESC\nLIMIT 5"}, {"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.Invoice Date >= '2010-01-01'\nGROUP BY i.InvoiceDate"}, {"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"}] Ollama Response: {'model': 'codegemma:latest', 'created at': '2024-06-13T23:01:14.386175427Z', 'message': {'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'}, 'done reason': 'stop', 'done': True, 'total duration': 987134927 56, 'load duration': 924926, 'prompt eval count': 1962, 'prompt eval duration': 87403438000, 'eval count': 39, 'eval duration': 10639920000} SELECT c.CustomerId, SUM(i.Total) AS TotalSpent FROM "customers" c JOIN "invoices" i ON c.CustomerId = i.CustomerId

```
GROUP BY c

SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c

JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c

Couldn't run sql: Execution failed on sql 'SELECT c.CustomerId, SUM(i.Total) AS TotalSpent
FROM "customers" c

JOIN "invoices" i ON c.CustomerId = i.CustomerId
GROUP BY c': no such column: c

In [39]: question = """

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

vn.ask(question=question)

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

[{'role': 'system', 'content': 'You are a SQLite expert. Please help to generate a SQL 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 PlaylistId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n ATE TABLE "plavlists"\r\n(\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\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCRE ATE TABLE "tracks"\r\n(\r\n TrackId INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,\r\n Name NVARCHAR(20 0) NOT NULL,\r\n AlbumId INTEGER.\r\n MediaTypeId INTEGER NOT NULL,\r\n GenreId INTEGER,\r\n Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER,\r\n UnitPrice NUMER IC(10,2) NOT NULL,\r\n FOREIGN KEY (Albumid) REFERENCES "albums" (Albumid) \r\n\t\tON DELETE NO ACTION 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 e 3 tables: artists, albums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId.\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', 'content': 'SELECT a.Name, COUNT(t.TrackId) AS TotalTracks\nFROM "artists" a\nJOIN "a lbums" al ON a.ArtistId = al.ArtistId\nJOIN "tracks" t ON al.AlbumId = t.AlbumId\nGROUP BY a.Name\nORDER BY Identify artists who have albums with t TotalTracks DESC\nLIMIT 10'}, {'role': 'user', 'content': ' \n racks 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 JOIN "tracks" t2 ON q2.GenreId = t2.GenreId\n GROUP BY q2.GenreId\n HAV

customer who bought the most albums in total quantity (across all invoices): \n'}, {'role': 'assistant', 'c

ING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'}, {'role': 'user', 'content': ' \n

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 Find the top 5 customers who bought the most albums in total quantity (across al invoice items. \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.InvoiceI d = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': Find all tracks with a name containing "What" (case-insensitive)\n'}, {'role': 'assistant', 'conte nt': 'SELECT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%what%\''}, {'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 d the top 5 most expensive tracks (based on unit price):\n'}, {'role': 'assistant', 'content': 'SELECT t.Tr ackId, t.Name, t.UnitPrice\nFROM "tracks" t\nORDER BY t.UnitPrice DESC\nLIMIT 5'}, {'role': 'user', 'conten 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.A rtistId'}, {'role': 'user', 'content': '\n Get all playlists containing at least 10 tracks and the to tal duration of those tracks:\n'\\ Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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\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  $R.\r\n$ Composer NVARCHAR(220),\r\n Milliseconds INTEGER NOT NULL,\r\n Bytes INTEGER.\r\n UnitP FOREIGN KEY (AlbumId) REFERENCES \"albums\" (AlbumId) \r\n\t\tON DELET rice NUMERIC(10,2) NOT NULL,\r\n E NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (GenreId) REFERENCES \"genres\" (GenreId) \r\n\t\tON DE LETE NO ACTION ON UPDATE NO ACTION,\r\n FOREIGN KEY (MediaTypeId) REFERENCES \"media types\" (MediaTypeI d) \r\n\t\tON DELETE NO ACTION ON UPDATE NO ACTION\r\n)\n\nCREATE INDEX IFK TrackGenreId ON \"tracks\" (Gen reId)\n\nCREATE INDEX IFK TrackAlbumId ON \"tracks\" (AlbumId)\n\nCREATE INDEX IFK TrackMediaTypeId ON \"tr

Find the

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 $\nFROM \genres \ggnares \ggna$ e"}, {"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", "c Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}, {"ro le": "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.GenreId\n HAVING COUNT(g2.GenreId) >  $1\n$ )\nGROUP B Y a.ArtistId, a.Name"}, {"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 \"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 \"inv oices\" 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 all tracks wi th a name containing \"What\" (case-insensitive)\n"}, {"role": "assistant", "content": "SELECT \* \nFROM \"t racks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"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 i.CustomerId, COUNT(ii.InvoiceLineId) 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 Find the top 5 most expen

sive tracks (based on unit price):\n"}, {"role": "assistant", "content": "SELECT t.TrackId, t.Name, t.UnitP rice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n List al 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 Get all playlists containing at least 10 tracks and the total duration of those tracks:\n"}]

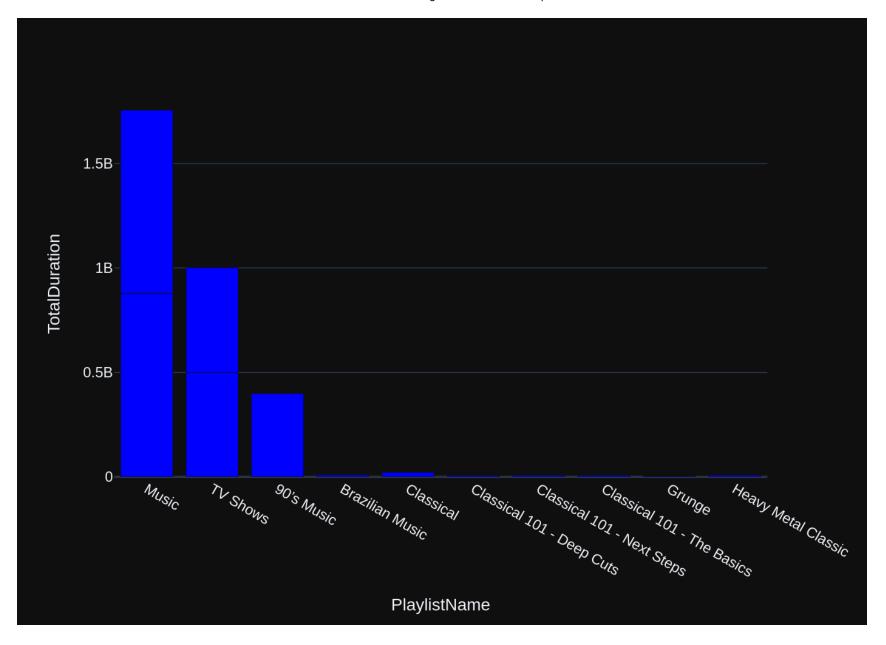
Add of existing embedding ID: e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql Insert of existing embedding ID: e7c4b3aa-664f-5f87-8b25-449a4482f3fd-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T23:02:57.347300303Z', 'message': {'role': 'assistan
t', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM "p
laylist track" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.Tr
ackId\nGROUP BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10'}, 'done reason': 'stop', 'done': Tru
e, 'total duration': 102920738722, 'load duration': 821971, 'prompt eval count': 1683, 'prompt eval duration'
n': 77459932000, 'eval count': 88, 'eval duration': 24780736000}
SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration
FROM "playlist track" pt
JOIN "playlists" p ON pt.PlaylistId = p.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY pt.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10
SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration
FROM "playlist track" pt
JOIN "playlists" p ON pt.PlaylistId = p.PlaylistId
JOIN "tracks" t ON pt.TrackId = t.TrackId
GROUP BY pt.PlaylistId, p.Name
HAVING COUNT(pt.TrackId) >= 10
                              PlaylistName TotalDuration
    PlavlistId
0
             1
                                     Music
                                                877683083
             3
1
                                  TV Shows
                                                501094957
2
             5
                                90's Music
                                                398705153
3
             8
                                    Music
                                                877683083
4
            10
                                  TV Shows
                                                501094957
5
                           Brazilian Music
            11
                                                 9486559
6
            12
                                 Classical
                                                21770592
7
            13 Classical 101 - Deep Cuts
                                                 6755730
8
            14 Classical 101 - Next Steps
                                                 7575051
9
            15 Classical 101 - The Basics
                                                  7439811
                                                 4122018
10
           16
                                    Grunge
            17
11
                       Heavy Metal Classic
                                                  8206312
Ollama parameters:
model=codegemma:latest,
options={},
keep alive=None
Prompt Content:
[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query
that answers the question the user asked: '\n Get all playlists containing at least 10 tracks and the
total duration of those tracks:\n'\nThe DataFrame was produced using this query: 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.N
```

ame\nHAVING COUNT(pt.TrackId) >= 10\n\nThe following is information about the resulting pandas DataFrame 'd f': \nRunning df.dtypes gives:\n PlaylistId int64\nPlaylistName object\nTotalDuration int64 \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': 'codegemma:latest', 'created\_at': '2024-06-13T23:03:24.715454197Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as px\n\nfig = px.bar(df, x='PlaylistName', y='TotalDurati on', hover\_name='PlaylistName')\nfig.update\_traces(marker\_color='blue')\nfig.show()\n```"}, 'done\_reason': 'stop', 'done': True, 'total\_duration': 27340345217, 'load\_duration': 700342, 'prompt\_eval\_count': 248, 'pr ompt\_eval\_duration': 11523796000, 'eval\_count': 56, 'eval\_duration': 15684832000}



```
Out[39]: ('SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM "playlist trac
          k" pt\nJOIN "playlists" p ON pt.PlaylistId = p.PlaylistId\nJOIN "tracks" t ON pt.TrackId = t.TrackId\nGROU
          P BY pt.PlaylistId, p.Name\nHAVING COUNT(pt.TrackId) >= 10',
               PlaylistId
                                         PlaylistName TotalDuration
          0
                                                Music
                                                           877683083
                        1
                        3
           1
                                             TV Shows
                                                           501094957
           2
                        5
                                           90's Music
                                                           398705153
           3
                       8
                                                Music
                                                           877683083
           4
                       10
                                             TV Shows
                                                           501094957
           5
                       11
                                      Brazilian Music
                                                             9486559
           6
                       12
                                            Classical
                                                            21770592
           7
                       13 Classical 101 - Deep Cuts
                                                             6755730
           8
                       14 Classical 101 - Next Steps
                                                             7575051
           9
                       15 Classical 101 - The Basics
                                                             7439811
           10
                       16
                                               Grunge
                                                             4122018
           11
                       17
                                  Heavy Metal Classic
                                                             8206312,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': ('<b>%{hovertext}</b><br>Pla' ... 'alDuration=%{y}<extra></extra>'),
                         'hovertext': array(['Music', 'TV Shows', '90's Music', 'Music', 'TV Shows',
                                             'Brazilian Music', 'Classical', 'Classical 101 - Deep Cuts',
                                             'Classical 101 - Next Steps', 'Classical 101 - The Basics', 'Grunge',
                                             'Heavy Metal Classic'], dtype=object),
                         'legendgroup': '',
                         'marker': {'color': 'blue', 'pattern': {'shape': ''}},
                         'name': '',
                         'offsetgroup': ''
                         'orientation': 'v',
                         'showlegend': False,
                         'textposition': 'auto',
                         'type': 'bar',
                         'x': array(['Music', 'TV Shows', '90's Music', 'Music', 'TV Shows',
                                     'Brazilian Music', 'Classical', 'Classical 101 - Deep Cuts',
                                     'Classical 101 - Next Steps', 'Classical 101 - The Basics', 'Grunge',
                                     'Heavy Metal Classic'], dtype=object),
                         'xaxis': 'x',
                         'y': array([877683083, 501094957, 398705153, 877683083, 501094957,
                                                                                              9486559,
                                      21770592, 6755730, 7575051, 7439811, 4122018,
                                                                                              8206312]),
                         'vaxis': 'v'}l.
               'lavout': {'barmode': 'relative'.
                          'legend': {'tracegroupgap': 0},
                          'margin': {'t': 60},
```

[{'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 ArtistId INTEGER NOT NULL,\r\n NOT NULL,\r\n Title NVARCHAR(160) 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 TrackId INTEGER NOT NULL,\r\n CONSTRAINT PK PlaylistTrack PRIMARY KEY (PlaylistId, TrackI ULL,\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\t0N 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 There are 3 tables: artists, alb ums and tracks, where albums and artists are linked by ArtistId, albums and tracks are linked by AlbumId,\n Can you find the top 10 most popular artists based on the number of tracks\n'}, {'role': 'assistant', '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 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 List all genres and the number of tracks in each genre:\n'}, {'role': 'assistant', 'content': 'SELECT g.Name, COUNT(t.GenreId) AS TotalTra cks\nFROM "genres" g\nJOIN "tracks" t ON g.GenreId = t.GenreId\nGROUP BY g.Name'}, {'role': 'user', 'conten Hint: album quantity is found in invoice items, \n \n Find the top 5 customers who bough t the most albums in total quantity (across all invoices):\n'}, {'role': 'assistant', 'content': 'SELECT c.

CustomerId, 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 TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find the customer who bought the most albums in tota l quantity (across all invoices): \n'}, {'role': 'assistant', 'content': 'SELECT c.CustomerId, COUNT(ii.Tra ckId) AS TotalAlbums\nFROM "customers" c\nJOIN "invoices" i ON c.CustomerId = i.CustomerId\nJOIN "invoice i tems" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY c.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 1'}, {'rol Hint: album quantity is found in invoice items, \n e': 'user', 'content': ' \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 ite ms" ii ON i.InvoiceId = ii.InvoiceId\nGROUP BY i.CustomerId\nORDER BY TotalAlbums DESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Get all playlists containing at least 10 tracks and the total duration of thos e tracks:\n'}, {'role': 'assistant', 'content': 'SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Millis econds) 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': ' \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 D ESC\nLIMIT 5'}, {'role': 'user', 'content': ' \n Find all tracks with a name containing "What" (case-in sensitive)\n'}, {'role': 'assistant', 'content': 'SELECT \* \nFROM "tracks" \nWHERE LOWER(Name) LIKE \'%wha t%\''}, {'role': 'user', 'content': ' \n Identify artists who have albums with tracks appearing in mul tiple genres:\n\n\n'}] Ollama parameters: model=codegemma:latest, options={}. keep alive=None Prompt Content: [{"role": "system", "content": "You are a SQLite expert. Please help to generate a SQL 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 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 PlaylistId INTEGER NOT NULL,\r\n TrackId INTEGER NOT NULL.\r\n KEY (PlavlistId, TrackId),\r\n FOREIGN KEY (PlaylistId) REFERENCES \"playlists\" (PlaylistId) \r\n\t\t0 N DELETE NO ACTION ON UPDATE NO ACTION.\r\n FOREIGN KEY (TrackId) REFERENCES \"tracks\" (TrackId) \r\n\t  $\t 0N DELETE NO ACTION ON UPDATE NO ACTION \( r \n \) \n \( n = = Additional Context \) \n \n In 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 query with a comment saying intermediate sql \n3. I f the provided context is insufficient, please explain why it can't be generated. \n4. Please use the most relevant table(s). \n5. If the question has been asked and answered before, please repeat the answer exactl y as it was given before. \n"}, {"role": "user", "content": " \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 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 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", "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 g.Name"}, {"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 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": " 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 quantity is found in invoice items, \n \n Find the top 5 customers who bought the most albums in tota l quantity (across all invoices):\n"}, {"role": "assistant", "content": "SELECT i.CustomerId, COUNT(ii.Invo iceLineId) 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 aylists containing at least 10 tracks and the total duration of those tracks:\n"}, {"role": "assistant", "c ontent": "SELECT pt.PlaylistId, p.Name AS PlaylistName, SUM(t.Milliseconds) AS TotalDuration\nFROM \"playli st track\" pt\nJOIN \"playlists\" p ON pt.PlaylistId = p.PlaylistId\nJOIN \"tracks\" t ON pt.TrackId = t.Tr ackId\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 t.TrackId, t.Name, t.UnitPrice\nFROM \"tracks\" t\nORDER BY t.UnitPrice DESC\nLIMIT 5"}, {"role": "user", "content": " \n Find all tracks with a name containing \"What\" (case-insensitive)\n"}, {"role": "assis tant", "content": "SELECT \* \nFROM \"tracks\" \nWHERE LOWER(Name) LIKE '%what%'"}, {"role": "user", "content": " \n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n"}]

Add of existing embedding ID: 9a9c970b-b94c-5f22-b54c-b86921a38b65-sql Insert of existing embedding ID: 9a9c970b-b94c-5f22-b54c-b86921a38b65-sql

```
Ollama Response:
{'model': 'codegemma:latest', 'created at': '2024-06-13T23:05:17.391736048Z', 'message': {'role': 'assistan
t', '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.Genre
Id = t2.GenreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name'},
'done reason': 'stop', 'done': True, 'total duration': 112573084080, 'load duration': 784922, 'prompt eval
count': 1676, 'prompt eval duration': 78939042000, 'eval count': 116, 'eval duration': 32953725000}
SELECT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
WHERE a.ArtistId IN (
  SELECT q2.GenreId
  FROM "genres" q2
  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId
  GROUP BY q2.GenreId
  HAVING COUNT(g2.GenreId) > 1
GROUP BY a.ArtistId, a.Name
SELECT a.ArtistId, a.Name AS ArtistName
FROM "artists" a
JOIN "albums" al ON a.ArtistId = al.ArtistId
WHERE a.ArtistId IN (
  SELECT q2.GenreId
  FROM "genres" g2
  JOIN "tracks" t2 ON g2.GenreId = t2.GenreId
  GROUP BY q2.GenreId
  HAVING COUNT(q2.GenreId) > 1
GROUP BY a.ArtistId, a.Name
    ArtistId
                                   ArtistName
0
           1
                                        AC/DC
           2
1
                                       Accept
2
           3
                                    Aerosmith
3
                            Alanis Morissette
4
           5
                              Alice In Chains
5
           6
                         Antônio Carlos Jobim
6
           7
                                 Apocalyptica
7
           8
                                   Audioslave
8
           9
                                     BackBeat
9
          10
                                 Billy Cobham
          11
10
                          Black Label Society
11
          12
                                Black Sabbath
```

```
12
          13
                                   Body Count
13
          14
                              Bruce Dickinson
14
          15
                                    Buddy Guy
15
                               Caetano Veloso
          16
16
          17
                                Chico Buarque
                  Chico Science & Nação Zumbi
17
          18
18
          19
                                 Cidade Negra
          20
19
                                 Cláudio Zoli
20
          21
                              Various Artists
21
          22
                                 Led Zeppelin
22
          23 Frank Zappa & Captain Beefheart
23
          24
                                 Marcos Valle
```

Ollama parameters:

model=codegemma:latest,

options={},

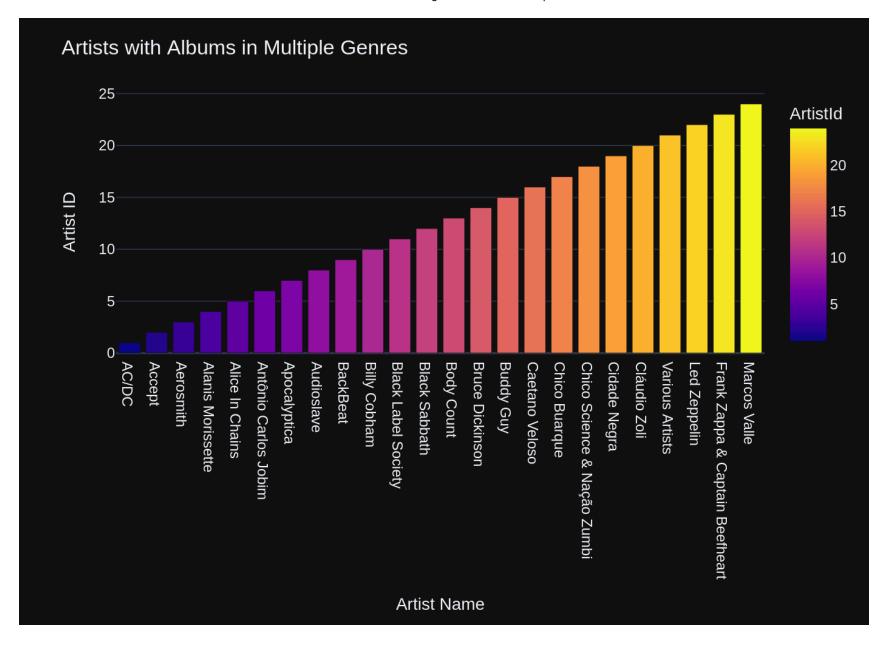
keep alive=None

Prompt Content:

[{"role": "system", "content": "The following is a pandas DataFrame that contains the results of the query that answers the question the user asked: '\n Identify artists who have albums with tracks appearing in multiple genres:\n\n\n\n\n\nThe DataFrame was produced using this query: SELECT a.ArtistId, a.Name AS Art istName\nFROM \"artists\" a\nJOIN \"albums\" al ON a.ArtistId = al.ArtistId\nWHERE a.ArtistId IN (\n SELEC T 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\n\nThe following is information about the resulting pandas DataFrame 'df': \nRunning df.dtypes gives:\n ArtistId int64\nArtistName object\nd type: object"}, {"role": "user", "content": "Can you generate the Python plotly code to chart the results of the dataframe? Assume the data is in a pandas dataframe called 'df'. If there is only one value in the dataframe, use an Indicator. Respond with only Python code. Do not answer with any explanations -- just the code."}]

#### Ollama Response:

{'model': 'codegemma:latest', 'created\_at': '2024-06-13T23:05:47.300414838Z', 'message': {'role': 'assistan t', 'content': "```python\nimport plotly.express as  $px\n = px.bar(df, x='ArtistName', y='ArtistId', color='ArtistId')\n = px.bar(df, x='ArtistName', y='ArtistId', color='ArtistId', color='ArtistId')\n = px.bar(df, x='ArtistName', y='ArtistId', color='ArtistId', color='ArtistId')\n = px.bar(df, x='ArtistName', y='ArtistId', color='ArtistId', color='ArtistId',$ 



```
Out[40]: ('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 JOIN "tracks" t2 ON q2.GenreId = t2.Ge
          nreId\n GROUP BY q2.GenreId\n HAVING COUNT(q2.GenreId) > 1\n)\nGROUP BY a.ArtistId, a.Name',
               ArtistId
                                               ArtistName
           0
                      1
                                                    AC/DC
           1
                      2
                                                   Accept
           2
                      3
                                                Aerosmith
           3
                      4
                                        Alanis Morissette
           4
                      5
                                          Alice In Chains
           5
                      6
                                    Antônio Carlos Jobim
           6
                      7
                                             Apocalyptica
           7
                      8
                                               Audioslave
           8
                      9
                                                 BackBeat
           9
                                             Billy Cobham
                     10
                     11
           10
                                      Black Label Society
           11
                     12
                                            Black Sabbath
                     13
           12
                                               Body Count
           13
                     14
                                          Bruce Dickinson
                     15
           14
                                                Buddy Guy
           15
                     16
                                           Caetano Veloso
           16
                     17
                                            Chico Buarque
                             Chico Science & Nação Zumbi
           17
                     18
           18
                     19
                                             Cidade Negra
           19
                     20
                                             Cláudio Zoli
           20
                     21
                                          Various Artists
                     22
           21
                                             Led Zeppelin
                         Frank Zappa & Captain Beefheart
           22
           23
                     24
                                             Marcos Valle,
           Figure({
               'data': [{'alignmentgroup': 'True',
                         'hovertemplate': 'ArtistName=%{x}<br/>br>ArtistId=%{marker.color}<extra></extra>',
                         'legendgroup': '',
                         'marker': {'color': 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]),
                                     'coloraxis': 'coloraxis',
                                     'pattern': {'shape': ''}},
                          'name': '',
                         'offsetgroup': '',
                         'orientation': 'v'.
                         'showlegend': False,
                         'textposition': 'auto',
```

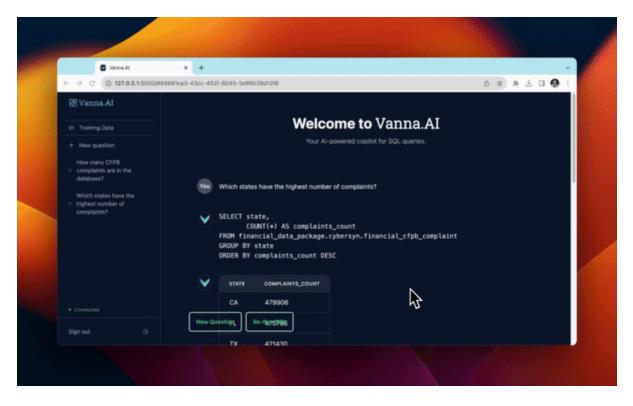
```
'tvpe': 'bar',
             'x': array(['AC/DC', 'Accept', 'Aerosmith', 'Alanis Morissette', 'Alice In Chains',
                        'Antônio Carlos Jobim', 'Apocalyptica', 'Audioslave', 'BackBeat',
                        'Billy Cobham', 'Black Label Society', 'Black Sabbath', 'Body Count',
                        'Bruce Dickinson', 'Buddy Guy', 'Caetano Veloso', 'Chico Buarque',
                        'Chico Science & Nação Zumbi', 'Cidade Negra', 'Cláudio Zoli',
                        'Various Artists', 'Led Zeppelin', 'Frank Zappa & Captain Beefheart',
                        'Marcos Valle', dtvpe=object),
             'xaxis': 'x'.
             'y': 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]),
             'yaxis': 'y'}],
    'layout': {'barmode': 'relative',
              'coloraxis': {'colorbar': {'title': {'text': 'ArtistId'}},
                           'colorscale': [[0.0, '#0d0887'], [0.1111111111111111,
                                         '#46039f'], [0.22222222222222,
                                         '#bd3786'], [0.555555555555556,
                                         '#ed7953'], [0.7777777777778,
                                         '#fb9f3a'], [0.888888888888888,
                                         '#fdca26'], [1.0, '#f0f921']]},
              'legend': {'tracegroupgap': 0},
              'margin': {'t': 60},
              'template': '...',
              'title': {'text': 'Artists with Albums in Multiple Genres'},
              'xaxis': {'anchor': 'y', 'domain': [0.0, 1.0], 'title': {'text': 'Artist Name'}},
              'yaxis': {'anchor': 'x', 'domain': [0.0, 1.0], 'title': {'text': 'Artist ID'}}}
}))
```

### 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 'codegemma' LLM took : 2572.97 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