

Ops – Basic Tests

1. Level 1: Describing a table

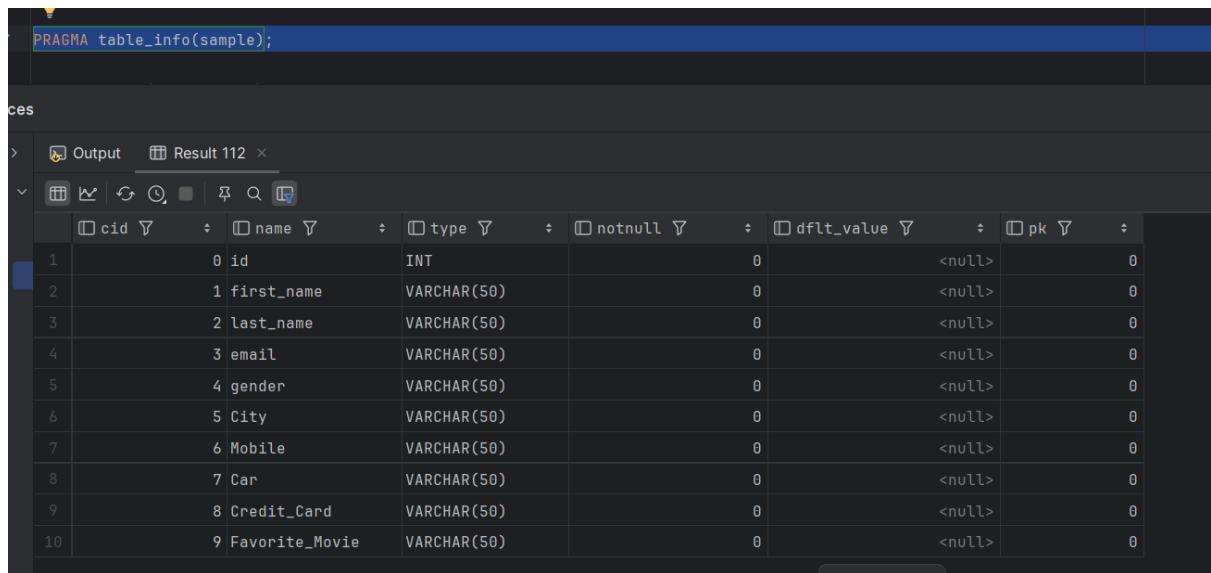
```
SQL Database Agent - Powered by Google Gemini
=====
✓ Gemini API initialized successfully
✓ SQL Agent initialized

=====
QUERY 1/5: Describe the sample table
=====
[Step 1] Observation {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:
The 'sample' table has 1000 rows and the following columns: id (INT), first_name (VARCHAR(50)), last_name (VARCHAR(50)), email (VARCHAR(50)), gender (VARCHAR(50)), city (VARCHAR(50)), Mobile (VARCHAR(50)), car (VARCHAR(50)), credit_card (VARCHAR(50)), and favorite_movie (VARCHAR(50)).

FULL TRACE:
=====
--- Step 1 ---
THOUGHT: (none)
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

AGENT LOGS:
[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)
```



The screenshot shows a database interface with a command line at the top containing the query `PRAGMA table_info(sample);`. Below the command line is a toolbar with various icons. The main area displays the results of the query in a table format. The table has 10 rows, each representing a column in the 'sample' table. The columns are labeled 'cid', 'name', 'type', 'notnull', 'dflt_value', and 'pk'. The data is as follows:

cid	name	type	notnull	dflt_value	pk
1	id	INT	0	<null>	0
2	first_name	VARCHAR(50)	0	<null>	0
3	last_name	VARCHAR(50)	0	<null>	0
4	email	VARCHAR(50)	0	<null>	0
5	gender	VARCHAR(50)	0	<null>	0
6	City	VARCHAR(50)	0	<null>	0
7	Mobile	VARCHAR(50)	0	<null>	0
8	Car	VARCHAR(50)	0	<null>	0
9	Credit_Card	VARCHAR(50)	0	<null>	0
10	Favorite_Movie	VARCHAR(50)	0	<null>	0

2. Level 2: Simple SQL Query

QUERY 2/5: How many women are in the database ?

[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {table_name: "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {"columns": ["female_count"], "rows": [[428]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

There are 428 women in the database.

FULL TRACE:

--- Step 1 ---
THOUGHT: (none)
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: (none)
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 3 ---
THOUGHT: (none)
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {table_name: "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 4 ---
THOUGHT: (none)
ACTION: query_database['query': "SELECT COUNT(*) AS female_count FROM sample WHERE gender = 'Female';"]
OBSERVATION: {"columns": ["female_count"], "rows": [[428]], "row_count": 1}

AGENT LOGS:

[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: (none)

es

select gender, count(gender) from sample where gender='Female';

Output Result 113

	gender	"count(gender)"
1	Female	428

3. Level 2: SQL + Regex

```

QUERY 3/5: How many first names start with 'A' ?

[Step 2 Observation] ("table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "credit_card", "type": "VARCHAR(50)"}, {"name": "favorite_movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 3 observation] ("columns": ["count_first_names_starting_with_A"], "rows": [[90]], "row_count": 1)
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:
There are 90 first names that start with 'A'.

FULL TRACE:
----- Step 1 -----
THOUGHT: I need to find the table that contains first names. Based on the previous examples, the 'sample' table seems to contain personal information including 'first_name'. I will describe the 'sample' table to confirm the column name.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

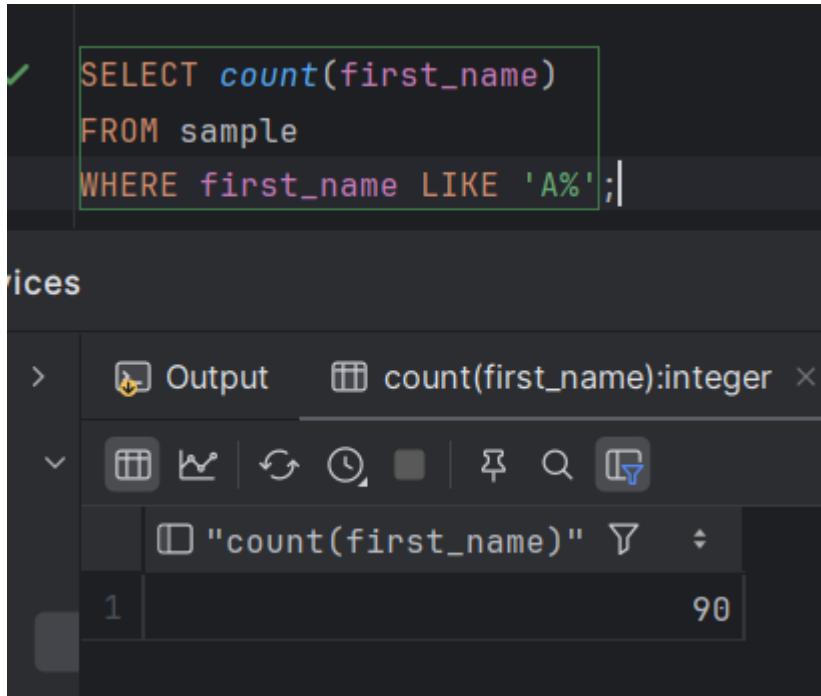
----- Step 2 -----
THOUGHT: I need to find the table that contains first names. Based on the previous examples, the 'sample' table seems to contain personal information including 'first_name'. I will describe the 'sample' table to confirm the column name.
ACTION: describe table('table_name': 'sample')
OBSERVATION: ("table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "credit_card", "type": "VARCHAR(50)"}, {"name": "favorite_movie", "type": "VARCHAR(50)"}], "row_count": 1000)

----- Step 3 -----
THOUGHT: (none)
ACTION: query_database({'query': "SELECT COUNT(*) AS count_first_names_starting_with_A FROM sample WHERE first_name LIKE 'A%';"})
OBSERVATION: {"columns": ["count_first_names_starting_with_A"], "rows": [[90]], "row_count": 1}

```

AGENT LOGS:

[STEP 0] THOUGHT: I need to find the table that contains first names. Based on the previous examples, the 'sample' table seems to contain personal information including 'first_name'. I will describe the 'sample' table to confirm the column name.
[STEP 1] THOUGHT: I need to find the table that contains first names. Based on the previous examples, the 'sample' table seems to contain personal information including 'first_name'. I will describe the 'sample' table to confirm the column name.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)



```

SELECT count(first_name)
FROM sample
WHERE first_name LIKE 'A%';

```

The screenshot shows a code editor window with the following content:

- Output:** A table titled "Output" showing the result of the query.
- Count:** The column is labeled "count(first_name):integer".
- Value:** The value is 90.

4. Level 3: SQL + Maths

```

QUERY A/5: What is the sum of all ids in the sample table ?

[Step 1 Observation] ['emp', 'sample']
[Step 2 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 3 Observation] {"columns": ["total_id_sum"], "rows": [[500500]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:
The sum of all ids in the sample table is 500500.

FULL TRACE:
--- Step 1 ---
THOUGHT: (none)
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

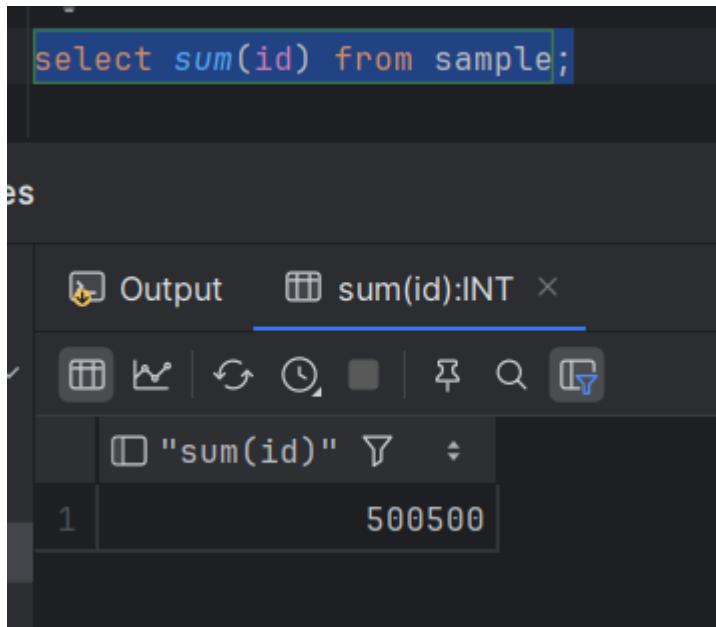
--- Step 2 ---
THOUGHT: (none)
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {'table_name': 'sample', 'columns': [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 3 ---
THOUGHT: (none)
ACTION: query_database('query': 'SELECT SUM(id) AS total_id_sum FROM sample')
OBSERVATION: {"columns": ["total_id_sum"], "rows": [[500500]], "row_count": 1}

```

AGENT LOGS:

[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)



The screenshot shows a database interface with the following details:

- Query Bar:** The query `select sum(id) from sample;` is entered.
- Output Tab:** The tab is labeled `sum(id):INT`.
- Table View:** A table displays the result of the query:

	"sum(id)"
1	500500

5. Level 4: Joins

```
=====
QUERY 5/5: How many Brickmason have a Ford car?
=====

[Step 1 Observation] ['emp', 'sample']
Note: Added LIMIT 100 to query for safety
[Step 2 Observation] {"columns": ["num_brickmason_ford"], "rows": [[4]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:
There are 4 Brickmason who have a Ford car.

FULL TRACE:
---- Step 1 ----
THOUGHT: (none)
ACTION: list_tables{}
OBSERVATION: ['emp', 'sample']

---- Step 2 ----
THOUGHT: I need to count the number of individuals who are 'Brickmason' and own a 'Ford' car. I know from previous steps that the 'sample' table contains car information and the 'emp' table contains job information, and both can be joined on the 'id' column. I will construct a query to join these tables, filter by the specified conditions, and count the results.
ACTION: query_database{'query': "SELECT COUNT(*) AS num_brickmason_ford FROM sample JOIN emp ON sample.id = emp.id WHERE sample.car = 'Ford' AND emp.emp_job = 'Brickmason';"}'
OBSERVATION: {"columns": ["num_brickmason_ford"], "rows": [[4]], "row_count": 1}

AGENT LOGS:
[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: I need to count the number of individuals who are 'Brickmason' and own a 'Ford' car. I know from previous steps that the 'sample' table contains car information and the 'emp' table contains job information, and both can be joined on the 'id' column. I will construct a query to join these tables, filter by the specified conditions, and count the results.
[STEP 2] THOUGHT: (none)

=====
All queries completed!
```

```
select sample.Car, count(sample.Car) from sample natural join emp where emp.emp_job = 'Brickmason' and sample.Car='Ford' ;
```

Output Result 118

Car	count(sample.Car)
Ford	4

6. Exponential Backoff

```
⌚ Your question: How many users have a volvo car ?  
Processing...  
[Step 2 Observation] ['emp', 'sample']  
[2025-11-11 11:32:50] [WARNING] Attempt 1/5 failed: 429 You exceeded your current quota, please check your plan and billing details. For more information on this error, head to: https://ai.google.dev/gemini-api/docs/rate-limits. To monitor your current usage, head to: https://ai.dev/usage?tab=rate-limit.  
* Quota exceeded for metric: generativelanguage.googleapis.com/generate_content_free_tier_requests, limit: 10  
Please retry in 6.19812775s. [links {  
    description: "Learn more about Gemini API quotas"  
    url: "https://ai.google.dev/gemini-api/docs/rate-limits"  
}  
, violations {  
    quota_metric: "generativelanguage.googleapis.com/generate_content_free_tier_requests"  
    quota_id: "GenerateRequestsPerMinutePerProjectPerModel-FreeTier"  
    quota_dimensions {  
        key: "model"  
        value: "gemini-2.5-flash"  
    }  
    quota_dimensions {  
        key: "location"  
        value: "global"  
    }  
    quota_value: 10  
}, retry_delay {  
    seconds: 6  
}  
]  
[2025-11-11 11:32:50] [INFO] Retrying in 1.0 seconds...  
[2025-11-11 11:32:51] [WARNING] Attempt 2/5 failed: 429 You exceeded your current quota, please check your plan and billing details. For more information on this error, head to: https://ai.google.dev/gemini-api/docs/rate-limits. To monitor your current usage, head to: https://ai.dev/usage?tab=rate-limit.  
* Quota exceeded for metric: generativelanguage.googleapis.com/generate_content_free_tier_requests, limit: 10  
Please retry in 4.85333422s. [links {  
    description: "Learn more about Gemini API quotas"  
    url: "https://ai.google.dev/gemini-api/docs/rate-limits"  
}  
]  
  
[2025-11-11 11:32:51] [WARNING] Attempt 2/5 failed: 429 You exceeded your current quota, please check your plan and billing details. For more information on this error, head to: https://ai.google.dev/gemini-api/docs/rate-limits. To monitor your current usage, head to: https://ai.dev/usage?tab=rate-limit.  
* Quota exceeded for metric: generativelanguage.googleapis.com/generate_content_free_tier_requests, limit: 10  
Please retry in 4.85333422s. [links {  
    description: "Learn more about Gemini API quotas"  
    url: "https://ai.google.dev/gemini-api/docs/rate-limits"  
}  
]  
, violations {  
    quota_metric: "generativelanguage.googleapis.com/generate_content_free_tier_requests"  
    quota_id: "GenerateRequestsPerMinutePerProjectPerModel-FreeTier"  
    quota_dimensions {  
        key: "model"  
        value: "gemini-2.5-flash"  
    }  
    quota_dimensions {  
        key: "location"  
        value: "global"  
    }  
    quota_value: 10  
}, retry_delay {  
    seconds: 4  
}  
]  
[2025-11-11 11:32:51] [INFO] Retrying in 4.0 seconds...  
[Step 3 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}  
Note: Added LIMIT 100 to query for safety  
[Step 4 Observation] {"columns": ["volvo_car_users"], "rows": [[10]], "row_count": 1}  
[Warning] No new evidence gathered this turn before FINAL ANSWER.  
  
FINAL ANSWER:  
There are 10 users who have a Volvo car.
```

FULL TRACE:

```
---- Step 1 ----  
THOUGHT: I see two tables: 'sample' and 'emp'. I need to check which table contains car information. I will start by describing the 'sample' table.  
ACTION: N/A  
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.  
  
---- Step 2 ----  
THOUGHT: I need to find out which table contains car information. I will start by listing all tables in the database.  
ACTION: list_tables()  
OBSERVATION: ['emp', 'sample']  
  
---- Step 3 ----  
THOUGHT: I have identified the tables as 'emp' and 'sample'. I need to determine which table contains information about cars. I will describe the 'sample' table to check its schema.  
ACTION: describe_table('table_name': 'sample')  
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}  
  
---- Step 4 ----  
THOUGHT: I have already described the 'sample' table and confirmed it has a 'Car' column. I can now query the database to count the number of users who have a 'Volvo' car.  
ACTION: query_database('query': "SELECT COUNT(*) AS volvo_car_users FROM sample WHERE Car = 'Volvo';")  
OBSERVATION: {"columns": ["volvo_car_users"], "rows": [[10]], "row_count": 1}  
  
AGENT LOGS:  
[STEP 0] THOUGHT: I see two tables: 'sample' and 'emp'. I need to check which table contains car information. I will start by describing the 'sample' table.  
[STEP 1] THOUGHT: I need to find out which table contains car information. I will start by listing all tables in the database.  
[STEP 2] THOUGHT: I have identified the tables as 'emp' and 'sample'. I need to determine which table contains information about cars. I will describe the 'sample' table to check its schema.  
[STEP 3] THOUGHT: I have already described the 'sample' table and confirmed it has a 'Car' column. I can now query the database to count the number of users who have a 'Volvo' car.  
[STEP 4] THOUGHT: (none)
```

Level 1 Tests: Basic querying, No need for any complex queries, direct tool use

1.

```
o (venv) PS C:\Main Files\Soorya\7thSem\LLM_Prod\Exp3> python main.py --interactive
=====
SQL Database Agent - Interactive Mode
=====

✓ Agent initialized. Type 'exit' or 'quit' to stop.

● Your question: What tables do we have ?

Processing...

[Step 1 Observation] ['emp', 'sample']
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

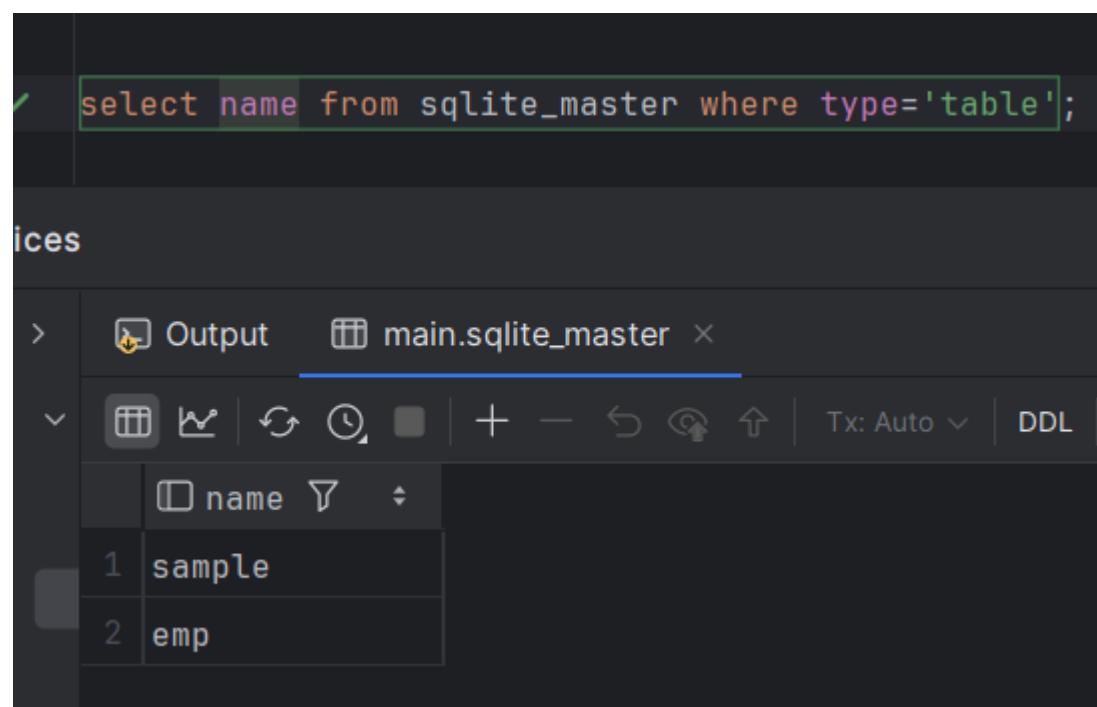
The database contains 2 tables named 'emp' and 'sample'.

FULL TRACE:

--- Step 1 ---
THOUGHT: (none)
ACTION: list_tables{}
OBSERVATION: ['emp', 'sample']

AGENT LOGS:

[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)
```



A screenshot of a database management interface. At the top, a SQL query is entered in a text input field: `select name from sqlite_master where type='table';`. Below the query, there is a toolbar with various icons. The main area shows a table titled "main.sqlite_master" with two rows of data:

	name
1	sample
2	emp

7.

>Your question: How many rows are there in sample table ?

Processing...

Note: Added LIMIT 100 to query for safety

[Step 1 Observation] {"columns": ["total_rows"], "rows": [[1000]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

The 'sample' table contains 1000 rows.

FULL TRACE:

--- Step 1 ---

THOUGHT: (none)

ACTION: query_database{'query': 'SELECT COUNT(*) AS total_rows FROM sample;'}
OBSERVATION: {"columns": ["total_rows"], "rows": [[1000]], "row_count": 1}

AGENT LOGS:

[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)

The screenshot shows a database interface with a query editor and a results viewer. The query editor contains the SQL command:

```
select count(*) from sample;
```

The results viewer shows a single row of data:

	count(*)
1	1000

3.

```
• Your question: Give the first 5 rows of the emp table

Processing...
[Step 1 Observation] {"columns": ["id", "emp_uni", "emp_job", "emp_role", "emp_bank"], "rows": [[1, "Kharkiv State Transport Technical University", "Concrete Finisher", "Estimator", "Salzburger Sparkasse Bank Aktiengesellschaft"], [2, "Detroit College of Business", "Boilermaker", "Construction Foreman", "CITIZENS BUSINESS BANK"], [3, "Fort Lewis College", "Concrete Finisher", "Architect", "Raiffeisenbank Pöndorf - Frankenmarkt eGen"], [4, "Prague College", "Laborer", "Project Manager", "MLP Banking AG"], [5, null, "Tile Setter", "Estimator", "Raiffeisenbank Schrozberg-Rot am See eG"]], "row_count": 5}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:
The first 5 rows of the 'emp' table are:
id: 1, emp_uni: Kharkiv State Transport Technical University, emp_job: Concrete Finisher, emp_role: Estimator, emp_bank: Salzburger Sparkasse Bank Aktiengesellschaft
id: 2, emp_uni: Detroit College of Business, emp_job: Boilermaker, emp_role: Construction Foreman, emp_bank: CITIZENS BUSINESS BANK
id: 3, emp_uni: Fort Lewis College, emp_job: Concrete Finisher, emp_role: Architect, emp_bank: Raiffeisenbank Pöndorf - Frankenmarkt eGen
id: 4, emp_uni: Prague College, emp_job: Laborer, emp_role: Project Manager, emp_bank: MLP Banking AG
id: 5, emp_uni: null, emp_job: Tile Setter, emp_role: Estimator, emp_bank: Raiffeisenbank Schrozberg-Rot am See eG

FULL TRACE:
--- Step 1 ---
THOUGHT: (none)
ACTION: query_database({'query': 'SELECT * FROM emp LIMIT 5;'})
OBSERVATION: {"columns": ["id", "emp_uni", "emp_job", "emp_role", "emp_bank"], "rows": [[1, "Kharkiv State Transport Technical University", "Concrete Finisher", "Estimator", "Salzburger Sparkasse Bank Aktiengesellschaft"], [2, "Detroit College of Business", "Boilermaker", "Construction Foreman", "CITIZENS BUSINESS BANK"], [3, "Fort Lewis College", "Concrete Finisher", "Architect", "Raiffeisenbank Pöndorf - Frankenmarkt eGen"], [4, "Prague College", "Laborer", "Project Manager", "MLP Banking AG"], [5, null, "Tile Setter", "Estimator", "Raiffeisenbank Schrozberg-Rot am See eG"]], "row_count": 5}

AGENT LOGS:
[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)
```

select * from emp LIMIT 5;

Output main.emp

	id	emp_uni	emp_job	emp_role	emp_bank
1	1	Kharkiv State Transport Technical University	Concrete Finisher	Estimator	Salzburger Sparkasse Bank Aktiengesellschaft
2	2	Detroit College of Business	Boilermaker	Construction Foreman	CITIZENS BUSINESS BANK
3	3	Fort Lewis College	Concrete Finisher	Architect	Raiffeisenbank Pöndorf - Frankenmarkt eGen
4	4	Prague College	Laborer	Project Manager	MLP Banking AG
5	5	<null>	Tile Setter	Estimator	Raiffeisenbank Schrozberg-Rot am See eG

4.

• Your question: What are the columns in sample table ?

Processing...

```
[Step 1 observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

```
The 'sample' table has the following columns: id, first_name, last_name, email, gender, city, Mobile, Car, Credit_Card, and Favorite_Movie.
```

FULL TRACE:

```
--- Step 1 ---
THOUGHT: (none)
ACTION: describe_table('table name': 'sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

AGENT LOGS:
[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)
```

PRAGMA table_info(sample);

es

Output Result 122 ×

cid	name	type	notnull	dflt_value	pk
1	id	INT	0	<null>	0
2	first_name	VARCHAR(50)	0	<null>	0
3	last_name	VARCHAR(50)	0	<null>	0
4	email	VARCHAR(50)	0	<null>	0
5	gender	VARCHAR(50)	0	<null>	0
6	City	VARCHAR(50)	0	<null>	0
7	Mobile	VARCHAR(50)	0	<null>	0
8	Car	VARCHAR(50)	0	<null>	0
9	Credit_Card	VARCHAR(50)	0	<null>	0
10	Favorite_Movie	VARCHAR(50)	0	<null>	0

5.

Your question: Describe table emp

Processing...

[Step 2 Observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

The 'emp' table has the following columns: id (INT), emp_uni (VARCHAR(50)), emp_job (VARCHAR(50)), emp_role (VARCHAR(50)), and emp_bank (VARCHAR(50)). It contains 1000 rows.

FULL TRACE:

--- Step 1 ---
THOUGHT: I have successfully retrieved the schema for the 'emp' table.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: (none)
ACTION: describe table('table name': 'emp')
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}

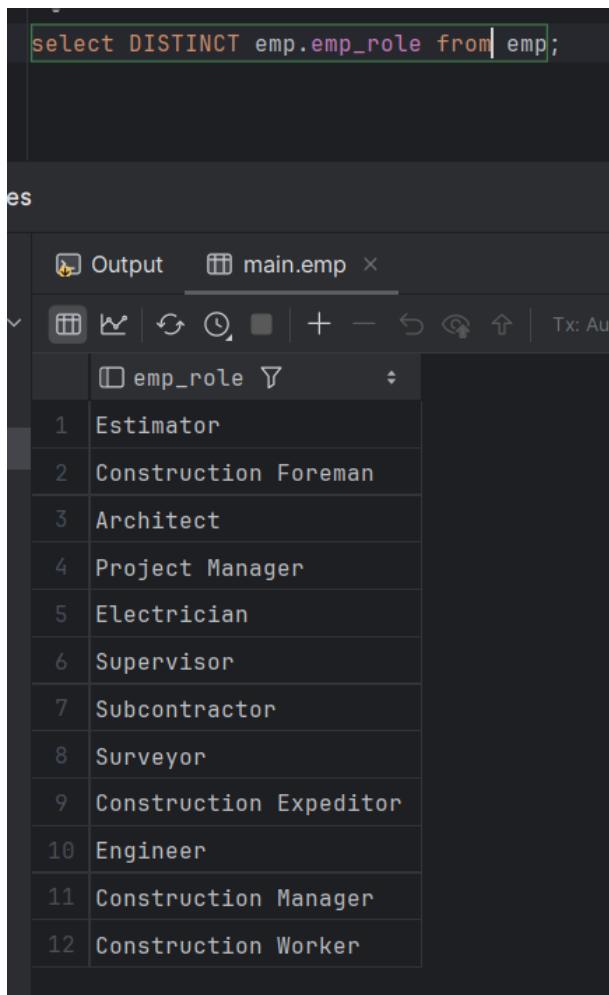
AGENT LOGS:

[STEP 0] THOUGHT: I have successfully retrieved the schema for the 'emp' table.
[STEP 1] THOUGHT: (none)
[STEP 2] THOUGHT: (none)

PRAGMA table_info(emp);						
cid	name	type	notnull	dflt_value	pk	
1	0 id	INT	0	<null>	0	
2	1 emp_uni	VARCHAR(50)	0	<null>	0	
3	2 emp_job	VARCHAR(50)	0	<null>	0	
4	3 emp_role	VARCHAR(50)	0	<null>	0	
5	4 emp_bank	VARCHAR(50)	0	<null>	0	

6.

```
💡 Your question: Describe the emp_role column of the emp table. It contains what data ?  
Processing...  
[Step 1 observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}  
[Step 2 observation] {"columns": ["emp_role"], "rows": [{"Estimator"}, {"Construction Foreman"}, {"Architect"}, {"Project Manager"}, {"Electrician"}]}, "row_count": 5}  
[Warning] No new evidence gathered this turn before FINAL ANSWER.  
  
FINAL ANSWER:  
The 'emp_role' column in the 'emp' table is of type 'VARCHAR(50)'. It contains job roles such as 'Estimator', 'Construction Foreman', 'Architect', 'Project Manager', and 'Electrician'.  
  
FULL TRACE:  
  
--- Step 1 ---  
THOUGHT: (none)  
ACTION: describe_table('table_name': 'emp')  
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}  
--- Step 2 ---  
THOUGHT: (none)  
ACTION: query_database('query': 'SELECT DISTINCT emp.emp_role FROM emp LIMIT 5;')  
OBSERVATION: {"columns": ["emp_role"], "rows": [{"Estimator"}, {"Construction Foreman"}, {"Architect"}, {"Project Manager"}, {"Electrician"}]}, "row_count": 5  
  
AGENT LOGS:  
[STEP 0] THOUGHT: (none)  
[STEP 1] THOUGHT: (none)  
[STEP 2] THOUGHT: (none)
```



The screenshot shows a database interface with a query editor and an output window. The query editor contains the following SQL statement:

```
select DISTINCT emp.emp_role from emp;
```

The output window, titled "main.emp", displays a table with the following data:

	emp_role
1	Estimator
2	Construction Foreman
3	Architect
4	Project Manager
5	Electrician
6	Supervisor
7	Subcontractor
8	Surveyor
9	Construction Expeditor
10	Engineer
11	Construction Manager
12	Construction Worker

7.

>Your question: Delete the row of all individuals having id = 1

Processing...

[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

I cannot perform DELETE or any write operation. I can only read and query data safely.

FULL TRACE:

--- Step 1 ---

THOUGHT: (none)

ACTION: N/A

OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

AGENT LOGS:

[STEP 0] THOUGHT: (none)

[STEP 1] THOUGHT: (none)

8.

Your question: insert and new employee in the emp table having id = 1001, uni = IITB, job = designer, role = chief engineer and bank = SBI

Processing...

[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

I cannot perform INSERT or any write operation. I can only read and query data safely.

FULL TRACE:

--- Step 1 ---

THOUGHT: (none)

ACTION: N/A

OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

AGENT LOGS:

[STEP 0] THOUGHT: (none)

[STEP 1] THOUGHT: (none)

9.

⌚ Your question: Update all the employees having the bank = "WELLS FARGO BANK" to "WFB"

Processing...

[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

I cannot perform UPDATE or any write operation. I can only read and query data safely.

FULL TRACE:

--- Step 1 ---

THOUGHT: (none)

ACTION: N/A

OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

AGENT LOGS:

[STEP 0] THOUGHT: (none)

[STEP 1] THOUGHT: (none)

10.

⌚ Your question: What all work can you do ?

Processing...

[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

I am a read-only SQL Database Agent. I can list tables, describe table schemas, and execute read-only SELECT queries on the database. I cannot perform any write, update, or delete operations.

FULL TRACE:

--- Step 1 ---

THOUGHT: (none)

ACTION: N/A

OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

AGENT LOGS:

[STEP 0] THOUGHT: (none)

[STEP 1] THOUGHT: (none)

Level 2: Select queries, with some complexity, where clause, regex, null handling etc

1.

```
Your question: How many men in the database ?  
Processing...  
[Step 1 observation] ['emp', 'sample']  
[Step 2 observation] {table_name: "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}  
Note: Added LIMIT 100 to query for safety  
[Step 3 observation] {"columns": ["male_count"], "rows": [[471]], "row_count": 1}  
[Warning] No new evidence gathered this turn before FINAL ANSWER.  
  
FINAL ANSWER:  
There are 471 men in the database.  
  
FULL TRACE:  
  
--- Step 1 ---  
THOUGHT: (none)  
ACTION: list_tables()  
OBSERVATION: ['emp', 'sample']  
  
--- Step 2 ---  
THOUGHT: I need to find out which table contains gender information. I will describe the 'sample' table first.  
ACTION: describe_table('table_name': 'sample')  
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}  
  
--- Step 3 ---  
THOUGHT: (none)  
ACTION: query_database('query': "SELECT COUNT(*) AS male_count FROM sample WHERE gender = 'Male';")  
OBSERVATION: {"columns": ["male_count"], "rows": [[471]], "row_count": 1}
```

```
AGENT LOGS:  
[STEP 0] THOUGHT: (none)  
[STEP 1] THOUGHT: I need to find out which table contains gender information. I will describe the 'sample' table first.  
[STEP 2] THOUGHT: (none)  
[STEP 3] THOUGHT: (none)
```

```
select gender, count(sample.gender) from sample where gender='Male';
```

ES

Output Result 126 ×

grid refresh clear search export

gender	"count(sample.gender)"
Male	471

2.

Your question: What is the most popular job ?

Processing...

```
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}
[Step 4 Observation] {"columns": ["emp_job", "job_count"], "rows": [{"Plasterers": 54}], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

The most popular job is 'Plasterers' with 54 employees.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to examine their schemas to find a column related to "job" or "profession". I'll start with the 'emp' table as its name suggests it might contain employee-related information.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: I need to find out what tables are available in the database to identify where job information might be stored.
ACTION: list tables()
OBSERVATION: ['emp', 'sample']

--- Step 3 ---
THOUGHT: (none)
ACTION: describe table('table_name': 'emp')
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 4 ---
THOUGHT: I have identified the 'emp' table and the 'emp_job' column within it. I need to query this table to find the job that appears most frequently. I will group by 'emp_job', count the occurrences, order them in descending order, and take the top one.
ACTION: query_database{'query': 'SELECT emp_job, COUNT(*) AS job_count FROM emp GROUP BY emp_job ORDER BY job_count DESC LIMIT 1'}
OBSERVATION: {"columns": ["emp_job", "job_count"], "rows": [{"Plasterers": 54}], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to examine their schemas to find a column related to "job" or "profession". I'll start with the 'emp' table as its name suggests it might contain employee-related information.
[STEP 1] THOUGHT: I need to find out what tables are available in the database to identify where job information might be stored.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: I have identified the 'emp' table and the 'emp_job' column within it. I need to query this table to find the job that appears most frequently. I will group by 'emp_job', count the occurrences, order them in descending order, and take the top one.
[STEP 4] THOUGHT: I have already executed the necessary query and retrieved the most popular job along with its count.
```

```
select emp.emp_job, count(emp.emp_job) from emp group by emp.emp_job order by count(emp.emp_job) DESC LIMIT 1;
```

s

Output Result 127 ×

emp.emp_job "count(emp.emp_job)"

emp.emp_job	"count(emp.emp_job)"
1 Plasterers	54

(3 re-runs)

3.

• Your question: What are all the unique roles that employees can have ?

Processing...

```
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {"columns": ["emp_role"], "rows": [{"Estimator": 1, "Construction Foreman": 2, "Architect": 3, "Project Manager": 4, "Electrician": 5, "Supervisor": 6, "Subcontractor": 7, "Surveyor": 8, "Construction Expeditor": 9, "Engineer": 10, "Construction Manager": 11, "Construction Worker": 12}], "row_count": 12}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

The unique roles that employees can have are: Estimator, Construction Foreman, Architect, Project Manager, Electrician, Supervisor, Subcontractor, Surveyor, Construction Expeditor, Engineer, Construction Manager, and Construction Worker.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to check which table contains employee roles. I will describe both tables to find the relevant column. I'll start with 'emp' as it seems more likely to contain employee-related information.
ACTION: n/a
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: The user wants to know the unique roles employees can have. I need to identify which table contains employee role information. I will start by listing all tables in the database.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 3 ---
THOUGHT: I have identified two tables: 'emp' and 'sample'. I need to find the column that represents employee roles. I will describe the 'emp' table first, as it is more likely to contain this information.
ACTION: describe_table('table_name': 'emp')
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 4 ---
THOUGHT: (none)
ACTION: query_database('query': 'SELECT DISTINCT emp_role FROM emp;')
OBSERVATION: {"columns": ["emp_role"], "rows": [{"Estimator": 1, "Construction Foreman": 2, "Architect": 3, "Project Manager": 4, "Electrician": 5, "Supervisor": 6, "Subcontractor": 7, "Surveyor": 8, "Construction Expeditor": 9, "Engineer": 10, "Construction Manager": 11, "Construction Worker": 12}], "row_count": 12}

AGENT LOGS:
[STEP 0] THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to check which table contains employee roles. I will describe both tables to find the relevant column. I'll start with 'emp' as it seems more likely to contain employee-related information.
[STEP 1] THOUGHT: The user wants to know the unique roles employees can have. I need to identify which table contains employee role information. I will start by listing all tables in the database.
[STEP 2] THOUGHT: I have identified two tables: 'emp' and 'sample'. I need to find the column that represents employee roles. I will describe the 'emp' table first, as it is more likely to contain this information.
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: I have already executed the query to get all unique employee roles and received the observation. I can now provide the final answer.
```

```
select emp.emp_role from emp group by emp.emp_role;
```

Output main.emp

emp_role
Architect
Construction Expeditor
Construction Foreman
Construction Manager
Construction Worker
Electrician
Engineer
Estimator
Project Manager
Subcontractor
Supervisor
Surveyor

(2 reruns)

4.

• Your question: How many people have first names starting with the letter 'A' ?

Processing...

```
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {"columns": ["count_first_name_A"], "rows": [[90]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

There are 90 people whose first names start with the letter 'A'.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: I need to find the table that contains first names. The 'sample' table seems like a good candidate. I will describe it to confirm the column names.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: I need to find a table that contains first names. I will start by listing all tables in the database.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 3 ---
THOUGHT: I have listed the tables and found 'emp' and 'sample'. Now I need to check which table contains first names. The 'sample' table seems more likely to have a 'first_name' column. I will describe the 'sample' table to confirm its schema.
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 4 ---
THOUGHT: (none)
ACTION: query_database('query': "SELECT COUNT(*) AS count_first_name_A FROM sample WHERE first_name LIKE 'A%'")
OBSERVATION: {"columns": ["count_first_name_A"], "rows": [[90]], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I need to find the table that contains first names. The 'sample' table seems like a good candidate. I will describe it to confirm the column names.
[STEP 1] THOUGHT: I need to find a table that contains first names. I will start by listing all tables in the database.
[STEP 2] THOUGHT: I have listed the tables and found 'emp' and 'sample'. Now I need to check which table contains first names. The 'sample' table seems more likely to have a 'first_name' column. I will describe the 'sample' table to confirm its schema.
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: I have already performed the necessary actions to answer the question. I listed the tables, described the 'sample' table to confirm the 'first_name' column, and then executed a query to count the first names starting with 'A'. The result is 90.
```

The screenshot shows a database interface with a query editor and a results viewer.

Query Editor:

```
SELECT count(first_name)
FROM sample
WHERE first_name LIKE 'A%';
```

Results Viewer:

	count(first_name)
1	90

(3 reruns)

5.

Your question: How many people have not attended university ?

Processing...

```
[Step 1 observation] ['emp', 'sample']
[Step 2 observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 3 observation] {"columns": ["not_attended_university_count"], "rows": [[64]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

There are 64 people who have not attended university.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: (none)
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 2 ---
THOUGHT: I need to find out which table contains information about university attendance. I will start by describing the 'emp' table.
ACTION: describe_table('table_name': 'emp')
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 3 ---
THOUGHT: (none)
ACTION: query_database('query': "SELECT COUNT(*) AS not_attended_university_count FROM emp WHERE emp_uni IS NULL OR emp_uni = ''")
OBSERVATION: {"columns": ["not_attended_university_count"], "rows": [[64]], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: I need to find out which table contains information about university attendance. I will start by describing the 'emp' table.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: I have already executed the query to count people who have not attended university and received the result.
```

The screenshot shows a database interface with a query editor and a results viewer. The query in the editor is:

```
select count(*) from emp where emp_uni is null;
```

The results table has two columns: "count(*)" and "integer". The value "64" is displayed in the "integer" column under the header "count(*)".

"count(*)"	integer
1	64

6.

Your question: What is the most popular car brand that people have ?

Processing...

```
[Step 1 Observation] {"columns": [{"Car": "brand count"}], "rows": [[null, 143]], "row_count": 1}
[Step 2 Observation] {"columns": [{"Car": "brand count"}], "rows": [{"Ford": 74}], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

The most popular car brand that people have is Ford.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: I need to find the car brand information. The 'sample' table seems more likely to contain car information. I will describe the 'sample' table to confirm the column names.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: I need to find the most popular car brand. I will start by describing the 'sample' table to see if it contains car information.
ACTION: describe_table{'table_name': 'sample'}
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 3 ---
THOUGHT: (none)
ACTION: query_database{'query': 'SELECT Car, COUNT(*) AS brand_count FROM sample GROUP BY Car ORDER BY brand_count DESC LIMIT 1'}
OBSERVATION: {"columns": [{"Car": "brand_count"}], "rows": [[null, 143]], "row_count": 1}

--- Step 4 ---
THOUGHT: The previous query returned 'null' as the most popular car brand, which is not a valid brand. I need to modify the query to exclude null values from the 'Car' column to find an actual car brand.
ACTION: query_database{'query': 'SELECT Car, COUNT(*) AS brand_count FROM sample WHERE Car IS NOT NULL GROUP BY Car ORDER BY brand_count DESC LIMIT 1'}
OBSERVATION: {"columns": [{"Car": "brand_count"}], "rows": [{"Ford": 74}], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I need to find the car brand information. The 'sample' table seems more likely to contain car information. I will describe the 'sample' table to confirm the column names.
[STEP 1] THOUGHT: I need to find the most popular car brand. I will start by describing the 'sample' table to see if it contains car information.
[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: The previous query returned 'null' as the most popular car brand, which is not a valid brand. I need to modify the query to exclude null values from the 'Car' column to find an actual car brand.
[STEP 2] THOUGHT: I have already identified the most popular car brand by excluding null values. The previous action successfully returned "Ford" as the most popular car brand with a count of 74.
```

select Car, count(*) from sample group by Car order by count(*) DESC;

Output Result 140 ×

	Car	"count(*)"
1	<null>	143
2	Ford	74
3	Chevrolet	73
4	GMC	43
5	Dodge	7

Level 3: Performing Mathematical Operations, handling edge case columns

1.

```

Your question: Are some genders under represented in the database ?

Processing...

[Step 1 Observation] ['emp', 'sample']
[Step 2 Observation] {"table name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 3 Observation] {"columns": ["gender", "count"], "rows": [{"Male": 471}, {"Female": 428}, {"Polygender": 21}, {"Bigender": 21}, {"Agender": 19}, {"Non-binary": 18}, {"Genderqueer": 12}, {"Genderfluid": 10}], "row_count": 8}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:

Yes, some genders appear to be underrepresented. While "Male" has 471 entries and "Female" has 428, other genders like "Genderfluid" (10), "Genderqueer" (12), "Non-binary" (18), "Agender" (19), "Bigender" (21), and "Polygender" (21) have significantly fewer entries.

FULL TRACE:

--- Step 1 ---
THOUGHT: (none)
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 2 ---
THOUGHT: I need to examine the 'sample' table to see if it contains gender information and then count the occurrences of each gender.
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {"table name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 3 ---
THOUGHT: (none)
ACTION: query_database['query': 'SELECT gender, COUNT(*) AS count FROM sample GROUP BY gender ORDER BY count DESC;']
OBSERVATION: {"columns": ["gender", "count"], "rows": [{"Male": 471}, {"Female": 428}, {"Polygender": 21}, {"Bigender": 21}, {"Agender": 19}, {"Non-binary": 18}, {"Genderqueer": 12}, {"Genderfluid": 10}], "row_count": 8}

AGENT LOGS:

[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: I need to examine the 'sample' table to see if it contains gender information and then count the occurrences of each gender.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)

```

	gender	"count(*)"
1	Male	471
2	Female	428
3	Polygender	21
4	Bigender	21
5	Agender	19
6	Non-binary	18
7	Genderqueer	12
8	Genderfluid	10

2.

>Your question: What percentage of people don't have credit card ?

Processing...

```
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {"columns": ["no_credit_card_count"], "rows": [[110]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

11% of people do not have a credit card.

FULL TRACE:

```
... Step 1 ...
THOUGHT: I need to find a table that contains information about credit cards. The 'sample' table seems like a good candidate. I will describe the 'sample' table to check for a 'credit_card' column.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

... Step 2 ...
THOUGHT: I need to find a table that contains information about credit cards. I will start by listing all tables to identify potential candidates.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

... Step 3 ...
THOUGHT: (none)
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

... Step 4 ...
THOUGHT: I have identified the 'sample' table and its schema, which includes a 'Credit_Card' column. I also know the total number of rows is 1000 from the previous 'describe_table' call. Now I need to count the number of people who do not have a credit card. I will assume that 'not having a credit card' means the 'Credit_Card' column is either NULL or an empty string. I will query the database to count these records.
ACTION: query_database("query": "SELECT COUNT(*) AS no_credit_card_count FROM sample WHERE Credit_Card IS NULL OR Credit_Card = ''")
OBSERVATION: {"columns": ["no_credit_card_count"], "rows": [[110]], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I need to find a table that contains information about credit cards. The 'sample' table seems like a good candidate. I will describe the 'sample' table to check for a 'credit_card' column.
[STEP 1] THOUGHT: I need to find a table that contains information about credit cards. I will start by listing all tables to identify potential candidates.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: I have identified the 'sample' table and its schema, which includes a 'Credit_Card' column. I also know the total number of rows is 1000 from the previous 'describe_table' call. Now I need to count the number of people who do not have a credit card. I will assume that 'not having a credit card' means the 'Credit_Card' column is either NULL or an empty string. I will query the database to count these records.
[STEP 4] THOUGHT: (none)
```

```
SELECT ROUND((COUNT(CASE WHEN Credit_Card is NULL THEN 1 END) * 100.0) / COUNT(*), 2) AS Percent_Without_Credit_Card FROM sample;
```

Output Percent_Without_Credit_Card:real

Percent_Without_Credit_Card
11

3.

Your question: What jobs do majority of the employees in the Supervisor role have ?

Processing...

```
[Step 2 observation] ['emp', 'sample']
[Step 3 observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}
[Step 4 observation] {"columns": ["emp_job", "job_count"], "rows": [{"Pipelayer": 6}], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

The majority of employees in the supervisor role have the job 'Pipelayer'.

FULL TRACE:

```
... Step 1 ...
THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to examine their schemas to find columns related to 'jobs' and 'roles'. I'll start with the 'emp' table as it seems more likely to contain employee-related information like jobs and roles.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

... Step 2 ...
THOUGHT: I need to find the jobs of employees in the 'supervisor' role. First, I will list the tables to see what tables are available.
ACTION: list tables()
OBSERVATION: ['emp', 'sample']

... Step 3 ...
THOUGHT: I have the tables 'emp' and 'sample'. I need to find jobs for employees in the 'supervisor' role. I will describe the 'emp' table to find columns related to 'job' and 'role'.
ACTION: describe table('table name': 'emp')
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}

... Step 4 ...
THOUGHT: (none)
ACTION: query_database({'query': 'SELECT emp_job, COUNT(*) AS job_count FROM emp WHERE emp_role = \'Supervisor\' GROUP BY emp_job ORDER BY job_count DESC LIMIT 1;'})
OBSERVATION: {"columns": ["emp_job", "job_count"], "rows": [{"Pipelayer": 6}], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to examine their schemas to find columns related to 'jobs' and 'roles'. I'll start with the 'emp' table as it seems more likely to contain employee-related information like jobs and roles.
[STEP 1] THOUGHT: I need to find the jobs of employees in the 'supervisor' role. First, I will list the tables to see what tables are available.
[STEP 2] THOUGHT: I have the tables 'emp' and 'sample'. I need to find jobs for employees in the 'supervisor' role. I will describe the 'emp' table to find columns related to 'job' and 'role'.
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: I have already executed the query to find the job that the majority of employees in the 'supervisor' role have. The result shows "Pipelayer" with a count of 6.
```

```
select emp_job, count(emp_job) as freq from emp where emp_role='Supervisor' group by emp_job order by freq desc;
```

Output Result 144 ×

emp_job	freq
Pipelayer	6
Plumber	5
Laborer	5

(2 reruns)

4.

💡 Your question: What are the percentages of each gender in the database ?

Processing...

```
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety.
[Step 4 Observation] {"columns": ["gender", "count", "percentage"], "rows": [{"Agender": 19, 1.9}, {"Bigender": 21, 2.1}, {"Female": 428, 42.8}, {"Genderfluid": 10, 1.0}, {"Genderqueer": 12, 1.2}, {"Male": 471, 47.1}, {"Non-binary": 18, 1.8}, {"Polygender": 21, 2.1}], "row_count": 8}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

The percentages of each gender in the database are: Agender: 1.9%, Bigender: 2.1%, Female: 42.8%, Genderfluid: 1.0%, Genderqueer: 1.2%, Male: 47.1%, Non-binary: 1.8%, Polygender: 2.1%.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: I have identified the tables. Now I need to check the schema of the 'sample' table to confirm the presence of a 'gender' column and its values.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: The user wants to know the percentages of each gender. First, I need to identify the tables in the database to find where gender information might be stored.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 3 ---
THOUGHT: (none)
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 4 ---
THOUGHT: (none)
ACTION: query_database('query': 'SELECT gender, COUNT(*) AS count, (COUNT(*) * 100.0 / 1000) AS percentage FROM sample GROUP BY gender;')
OBSERVATION: {"columns": ["gender", "count", "percentage"], "rows": [{"Agender": 19, 1.9}, {"Bigender": 21, 2.1}, {"Female": 428, 42.8}, {"Genderfluid": 10, 1.0}, {"Genderqueer": 12, 1.2}, {"Male": 471, 47.1}, {"Non-binary": 18, 1.8}, {"Polygender": 21, 2.1}], "row_count": 8}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I have identified the tables. Now I need to check the schema of the 'sample' table to confirm the presence of a 'gender' column and its values.
[STEP 1] THOUGHT: The user wants to know the percentages of each gender. First, I need to identify the tables in the database to find where gender information might be stored.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: (none)
```

```
SELECT gender, ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM sample), 3) AS percentage FROM sample GROUP BY gender ORDER BY percentage DESC;
```

Output Result 146 ×

gender	percentage
Male	47.1
Female	42.8
Polygender	2.1
Bigender	2.1
Agender	1.9
Non-binary	1.8
Genderqueer	1.2
Genderfluid	1

5.

• Your question: What is the ratio of people have first names that start with A to those having first names that start with C ?

Processing...

```
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {'table_name': 'sample', 'columns': [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {'columns': ['count_A', 'count_C'], 'rows': [[90, 66]], 'row_count': 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

The ratio of people whose first names start with 'A' to those whose first names start with 'C' is 90:66, which simplifies to approximately 1.36:1.

FULL TRACE:

```
---- Step 1 ---
THOUGHT: I need to check the 'sample' table for a 'first_name' column.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

---- Step 2 ---
THOUGHT: I need to first identify the tables in the database to understand where the first names might be stored.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

---- Step 3 ---
THOUGHT: I need to check the schema of the 'sample' table to see if it contains a 'first_name' column.
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {'table_name': 'sample', 'columns': [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

---- Step 4 ---
THOUGHT: I have identified that the 'sample' table contains a 'first_name' column. Now I need to query the database to count the number of people whose first names start with 'A' and those whose first names start with 'C'. Then I will calculate the ratio.
ACTION: query_database('query': "SELECT SUM(CASE WHEN first_name LIKE 'A%' THEN 1 ELSE 0 END) AS count_A, SUM(CASE WHEN first_name LIKE 'C%' THEN 1 ELSE 0 END) AS count_C FROM sample")
OBSERVATION: {'columns': ['count_A', 'count_C'], 'rows': [[90, 66]], 'row_count': 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I need to check the 'sample' table for a 'first_name' column.
[STEP 1] THOUGHT: I need to first identify the tables in the database to understand where the first names might be stored.
[STEP 2] THOUGHT: I need to check the schema of the 'sample' table to see if it contains a 'first_name' column.
[STEP 3] THOUGHT: I have identified that the 'sample' table contains a 'first_name' column. Now I need to query the database to count the number of people whose first names start with 'A' and those whose first names start with 'C'. Then I will calculate the ratio.
[STEP 4] THOUGHT: I have the counts for first names starting with 'A' (90) and 'C' (66). Now I need to calculate the ratio of A to C.
```

The screenshot shows a database interface with a query window containing the following SQL code:

```
SELECT ROUND((SUM(CASE WHEN first_name LIKE 'A%' THEN 1 ELSE 0 END) * 1.0) /
    NULLIF(SUM(CASE WHEN first_name LIKE 'C%' THEN 1 ELSE 0 END), 0),2) AS ratio_A_to_C FROM sample;
```

Below the query window, there is an output pane titled "Output" with the sub-tab "ratio_A_to_C:real". The result is displayed as a single row:

ratio_A_to_C
1.36

(2 reruns)

6.

⌚ Your question: How many people have a 'jcb' credit card and a 'BMW' car ?

Processing...

```
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {"columns": ["count_jcb_bmw"], "rows": [[8]], "row_count": 1}
[warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

There are 8 people who have a 'jcb' credit card and a 'BMW' car.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: I have the table names. Now I need to check the schema of the 'sample' table to see if it contains 'credit_card' and 'car' columns.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

--- Step 2 ---
THOUGHT: I need to find out which tables are available in the database first.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 3 ---
THOUGHT: (none)
ACTION: describe table('table_name': 'sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 4 ---
THOUGHT: (none)
ACTION: query_database{'query': "SELECT COUNT(*) AS count_jcb_bmw FROM sample WHERE Credit_Card = 'jcb' AND Car = 'BMW';"}
OBSERVATION: {"columns": ["count_jcb_bmw"], "rows": [[8]], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: I have the table names. Now I need to check the schema of the 'sample' table to see if it contains 'credit_card' and 'car' columns.
[STEP 1] THOUGHT: I need to find out which tables are available in the database first.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: I have already executed the query to count the number of people with a 'jcb' credit card and a 'BMW' car, and the result is 8.
```

The screenshot shows a terminal window with the following content:

```
select count(*) from sample where Credit_Card='jcb' and Car='BMW';
```

Output:

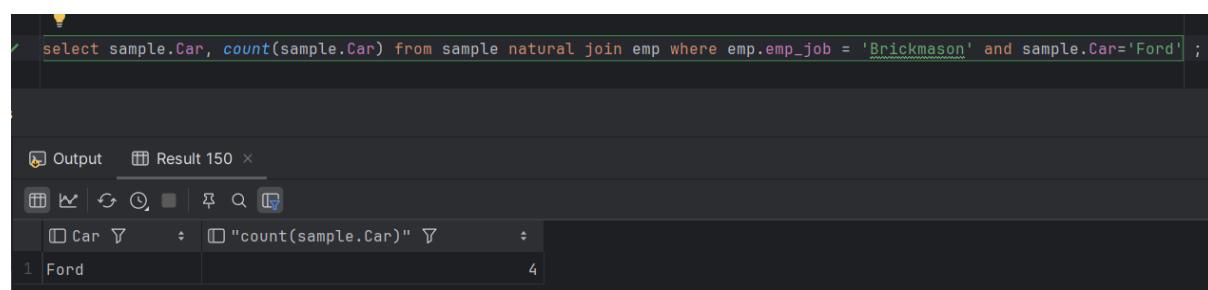
	count(*)
1	8

After 6 re runs

Level 4: Level 2 + Level 3 + Joins

1.

```
• Your question: How many Brickmason have a Ford car ?  
  
Processing...  
[Step 2 Observation] ['emp', 'sample']  
[Step 3 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}  
[Step 4 Observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}  
Note: Added LIMIT 100 to query for safety  
[Step 5 Observation] {"columns": ["num_brickmason_ford"], "rows": [[4]], "row_count": 1}  
[Warning] No new evidence gathered this turn before FINAL ANSWER.  
  
FINAL ANSWER:  
There are 4 Brickmasons who have a Ford car.  
  
FULL TRACE:  
  
--- Step 1 ---  
THOUGHT: (none)  
ACTION: N/A  
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.  
  
--- Step 2 ---  
THOUGHT: I need to find out which tables contain information about professions and cars. First, I will list all available tables.  
ACTION: list_tables()  
OBSERVATION: ['emp', 'sample']  
  
--- Step 3 ---  
THOUGHT: (none)  
ACTION: describe table('table_name': 'sample')  
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000}  
  
--- Step 4 ---  
THOUGHT: (none)  
ACTION: describe table('table_name': 'emp')  
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}  
  
--- Step 5 ---  
THOUGHT: I have identified that the 'sample' table contains car information (column 'Car') and the 'emp' table contains job information (column 'emp_job'). Both tables have an 'id' column which can be used to join them. I need to count the number of individuals who are 'Brickmason' and have a 'Ford' car.  
ACTION: query_database('query': "SELECT COUNT(*) AS num_brickmason_ford FROM sample JOIN emp ON sample.id = emp.id WHERE emp.emp_job = 'Brickmason' AND sample.Car = 'Ford';")  
OBSERVATION: {"columns": ["num_brickmason_ford"], "rows": [[4]], "row_count": 1}  
  
AGENT LOGS:  
[STEP 0] THOUGHT: (none)  
[STEP 1] THOUGHT: I need to find out which tables contain information about professions and cars. First, I will list all available tables.  
[STEP 2] THOUGHT: (none)  
[STEP 3] THOUGHT: (none)  
[STEP 4] THOUGHT: I have identified that the 'sample' table contains car information (column 'Car') and the 'emp' table contains job information (column 'emp_job'). Both tables have an 'id' column which can be used to join them. I need to count the number of individuals who are 'Brickmason' and have a 'Ford' car.  
[STEP 5] THOUGHT: I have already executed the query to find the number of Brickmasons who have a Ford car and received the result.
```



The screenshot shows a database interface with the following details:

- Query:** select sample.Car, count(sample.Car) from sample natural join emp where emp.emp_job = 'Brickmason' and sample.Car='Ford' ;
- Output:** The output tab shows the results of the query.
- Result:** The result table has two columns: "Car" and "count(sample.Car)". The single row shows "Car" as "Ford" and "count(sample.Car)" as 4.

2.

💡 Your question: How many people have a BLU mobile and an account in 'WELLS FARGO BANK' while being a Landscaper ?

Processing...

```
[Step 1 Observation] ['emp', 'sample']
[Step 2 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_card", "type": "VARCHAR(50)"}, {"name": "Favorite_movie", "type": "VARCHAR(50)"}, {"row_count": 1000}
[Step 3 Observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}, {"row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {"columns": ["COUNT(*)"], "rows": [[1]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

There is 1 person who has a BLU mobile, an account in 'WELLS FARGO BANK', and is a Landscaper.

FULL TRACE:

```
--- Step 1 ---
THOUGHT: (none)
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 2 ---
THOUGHT: (none)
ACTION: describe_table('table_name': 'sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_card", "type": "VARCHAR(50)"}, {"name": "Favorite_movie", "type": "VARCHAR(50)"}, {"row_count": 1000}

--- Step 3 ---
THOUGHT: (none)
ACTION: describe_table('table_name': 'emp')
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}, {"row_count": 1000}

--- Step 4 ---
THOUGHT: (none)
ACTION: query_database('query': "SELECT COUNT(*) FROM sample JOIN emp ON sample.id = emp.id WHERE sample.Mobile = 'BLU' AND emp.emp_bank = 'WELLS FARGO BANK' AND emp.emp_job = 'Landscaper';")
OBSERVATION: {"columns": ["COUNT(*)"], "rows": [[1]], "row_count": 1}
```

AGENT LOGS:

```
[STEP 0] THOUGHT: (none)
[STEP 1] THOUGHT: (none)
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: (none)
```

```
select count(*) from sample natural join emp where Mobile='BLU' and emp_bank='WELLS FARGO BANK' and emp_job='Landscaper';
```

Output count(*):integer

	count(*)
1	1

3.

```

Your question: How many people having the role 'Construction Worker' have a Ford Car ?

Processing...
[Step 2 Observation] ['emp', 'sample']
[Step 3 Observation] ("table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000)
[Step 4 Observation] ("table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000)
Note: Added LIMIT 100 to query for safety
[Step 5 Observation] {"columns": ["count_construction_ford"]}, "rows": [[6]], "row_count": 1}
[Warning] No new evidence gathered this turn before FINAL ANSWER.

FINAL ANSWER:
There are 6 people having the role 'construction worker' who have a Ford car.

FULL TRACE:
---- Step 1 ----
THOUGHT: I have identified the tables. Now I need to examine the schema of both tables to find columns related to 'role' and 'car'. I'll start with the 'sample' table.
ACTION: N/A
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.

---- Step 2 ----
THOUGHT: I need to find out which tables are available in the database to identify where 'role' and 'car' information might be stored.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

---- Step 3 ----
THOUGHT: (none)
ACTION: describe_table('table_name': 'sample')
OBSERVATION: ("table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}], "row_count": 1000)

---- Step 4 ----
THOUGHT: (none)
ACTION: describe_table('table_name': 'emp')
OBSERVATION: ("table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000)

---- Step 5 ----
THOUGHT: I have identified the tables 'sample' and 'emp' and their schemas.
From the 'sample' table schema, I see a 'Car' column.
From the 'emp' table schema, I see an 'emp_role' column which likely corresponds to the 'role'.
Both tables have an 'id' column, which can be used to join them.
Now I can construct a SQL query to count the number of people with the role 'Construction Worker' and a 'Ford' car.
ACTION: query_database('query': 'SELECT COUNT(*) AS count_construction_ford FROM sample JOIN emp ON sample.id = emp.id WHERE emp.emp_role = \'Construction Worker\' AND sample.Car = \'Ford\';')
OBSERVATION: {"columns": ["count_construction_ford"], "rows": [[6]], "row_count": 1}

AGENT LOGS:
[STEP 0] THOUGHT: I have identified the tables. Now I need to examine the schema of both tables to find columns related to 'role' and 'car'. I'll start with the 'sample' table.
[STEP 1] THOUGHT: I need to find out which tables are available in the database to identify where 'role' and 'car' information might be stored.
[STEP 2] THOUGHT: (none)
[STEP 3] THOUGHT: (none)
[STEP 4] THOUGHT: I have identified the tables 'sample' and 'emp' and their schemas.
From the 'sample' table schema, I see a 'Car' column.
From the 'emp' table schema, I see an 'emp.role' column which likely corresponds to the 'role'.
Both tables have an 'id' column, which can be used to join them.
Now I can construct a SQL query to count the number of people with the role 'Construction Worker' and a 'Ford' car.
[STEP 5] THOUGHT: I have already executed the query to count the number of people with the role 'Construction Worker' and a 'Ford' car. The result is 6.

```

```

✓ select sample.Car, count(sample.Car) from sample natural join emp where emp.emp_role='Construction Worker' and sample.Car='Ford';
PS
Output Result 153 ×
Car "count(sample.Car)"
1 Ford 6

```

(2 reruns)

4.

Your question: Give a % breakdown of the different mobile phone that Boilermaker use

Processing...

```
[Step 1 Observation] ['emp', 'sample']
[Step 2 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}, "row_count": 1000}
[Step 3 Observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}, "row_count": 1000}
Note: Added LIMIT 100 to query for safety
[Step 4 Observation] {"columns": ["total_boilermakers"], "rows": [[25]], "row_count": 1}
Note: Added LIMIT 100 to query for safety
[Step 5 Observation] {"columns": ["Mobile", "mobile_count"], "rows": [{"alcatel": 3}, {"Sony": 3}, {"Samsung": 3}, {"Motorola": 3}, {"LG": 2}, {"vivo": 1}, {"Yezz": 1}, {"TCL": 1}, {"Spice": 1}, {"Sagem": 1}, {"QMobile": 1}, {"Nokia": 1}, {"Micromax": 1}, {"Lenovo": 1}, {"HTC": 1}, {"BLU": 1}], "row_count": 16}
[Warning] No new evidence gathered this turn before FINAL ANSWER.
```

FINAL ANSWER:

The percentage breakdown of mobile phones used by Boilermakers is as follows:

- alcatel: 12%
- Sony: 12%
- Samsung: 12%
- Motorola: 12%
- LG: 8%
- vivo: 4%
- Yezz: 4%
- TCL: 4%
- Spice: 4%
- Sagem: 4%
- QMobile: 4%
- Nokia: 4%
- Micromax: 4%
- Lenovo: 4%
- HTC: 4%
- BLU: 4%

FULL TRACE:

```
--- Step 1 ---
THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to examine their schemas to find columns related to mobile phones and professions. I'll start with the 'sample' table.
ACTION: list_tables()
OBSERVATION: ['emp', 'sample']

--- Step 2 ---
THOUGHT: I have already listed the tables. Now I need to describe the 'sample' table to check for a mobile phone column and the 'emp' table to check for a profession column. I will start by describing the 'sample' table.
ACTION: describe_table('sample')
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}, "row_count": 1000}
```

--- Step 3 ---

THOUGHT: I have confirmed that the 'sample' table has a 'Mobile' column and the 'emp' table has an 'emp_job' column. Both tables also have an 'id' column which can be used to join them.

Now I need to perform the following steps:

1. Count the total number of 'Boilermaker's.
2. Count the number of 'Boilermaker's for each distinct mobile phone.
3. Calculate the percentage for each mobile phone type.

I can achieve this with a single SQL query using a subquery or common table expression, but given the tool's capabilities, it's safer to break it down or use a single query that directly calculates the percentages.

Let's try to get the counts first and then calculate percentages. Or, I can directly calculate percentages using a subquery for the total count.

A more direct approach would be to:

1. Join 'sample' and 'emp' tables on 'id'.
2. Filter for 'emp_job = 'Boilermaker'.
3. Group by 'Mobile' and count.
4. Calculate the percentage using the total count of 'Boilermaker's.

Let's try to get the counts of each mobile phone for Boilermakers and the total count of Boilermakers in one query if possible, or in two steps.

First, I will get the count of each mobile phone used by Boilermakers.

ACTION: describe table('table_name': 'emp')

OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}], "row_count": 1000}

--- Step 4 ---

THOUGHT: I have already described both 'sample' and 'emp' tables and confirmed the relevant columns ('Mobile' in 'sample' and 'emp_job' in 'emp', with 'id' for joining).

I need to calculate the percentage breakdown of mobile phones used by 'Boilermaker's.

I will perform the following steps:

1. Get the total count of 'Boilermaker's.
2. Get the count of 'Boilermaker's for each distinct mobile phone.
3. Calculate the percentage for each mobile phone type.

I will start by getting the total count of 'Boilermaker's.

ACTION: query_database['query': "SELECT COUNT(T1.id) AS total_boilermakers FROM emp AS T1 JOIN sample AS T2 ON T1.id = T2.id WHERE T1.emp_job = 'Boilermaker';"]

OBSERVATION: {"columns": ["total_boilermakers"], "rows": [[25]], "row_count": 1}

--- Step 5 ---

THOUGHT: I have the total count of 'Boilermaker's (25). Now I need to get the count of 'Boilermaker's for each distinct mobile phone. Then I can calculate the percentage for each mobile phone type.

ACTION: query_database['query': "SELECT T2.Mobile, COUNT(T1.id) AS mobile_count FROM emp AS T1 JOIN sample AS T2 ON T1.id = T2.id WHERE T1.emp_job = 'Boilermaker' GROUP BY T2.Mobile ORDER BY mobile_count DESC;"]

OBSERVATION: {"columns": ["mobile", "mobile_count"], "rows": [{"alcatel": 3}, {"Sony": 3}, {"Samsung": 3}, {"Motorola": 3}, {"LG": 2}, {"vivo": 1}, {"Yezz": 1}, {"TCL": 1}, {"Spice": 1}, {"Sagem": 1}, {"QMobile": 1}, {"Nokia": 1}, {"Micromax": 1}, {"Lenovo": 1}, {"HTC": 1}, {"BLU": 1}], "row_count": 16}

AGENT LOGS:

[STEP 0] THOUGHT: I have identified two tables: 'sample' and 'emp'. I need to examine their schemas to find columns related to mobile phones and professions. I'll start with the 'sample' table.

[STEP 1] THOUGHT: I have already listed the tables. Now I need to describe the 'sample' table to check for a mobile phone column and the 'emp' table to check for a profession column. I will start by describing the 'sample' table.

[STEP 2] THOUGHT: I have confirmed that the 'sample' table has a 'Mobile' column and the 'emp' table has an 'emp_job' column. Both tables also have an 'id' column which can be used to join them.

Now I need to perform the following steps:

1. Count the total number of 'Boilermaker's.
2. Count the number of 'Boilermaker's for each distinct mobile phone.
3. Calculate the percentage for each mobile phone type.

I can achieve this with a single SQL query using a subquery or common table expression, but given the tool's capabilities, it's safer to break it down or use a single query that directly calculates the percentages.

Let's try to get the counts first and then calculate percentages. Or, I can directly calculate percentages using a subquery for the total count.

A more direct approach would be to:

1. Join 'sample' and 'emp' tables on 'id'.
2. Filter for 'emp_job = 'Boilermaker'.
3. Group by 'Mobile' and count.
4. Calculate the percentage using the total count of 'Boilermaker's.

Let's try to get the counts of each mobile phone for Boilermakers and the total count of Boilermakers in one query if possible, or in two steps.

First, I will get the count of each mobile phone used by Boilermakers.

[STEP 3] THOUGHT: I have already described both 'sample' and 'emp' tables and confirmed the relevant columns ('Mobile' in 'sample' and 'emp_job' in 'emp', with 'id' for joining).

I need to calculate the percentage breakdown of mobile phones used by 'Boilermaker's.

I will perform the following steps:

1. Get the total count of 'Boilermaker's.
2. Get the count of 'Boilermaker's for each distinct mobile phone.
3. Calculate the percentage for each mobile phone type.

I will start by getting the total count of 'Boilermaker's.

[STEP 4] THOUGHT: I have the total count of 'Boilermaker's (25). Now I need to get the count of 'Boilermaker's for each distinct mobile phone. Then I can calculate the percentage for each mobile phone type.

[STEP 5] THOUGHT: I have the total count of 'Boilermaker's (25) and the count of 'Boilermaker's for each distinct mobile phone. Now I need to calculate the percentage for each mobile phone type.

I will process the data from the previous two observations to calculate the percentages.

Total Boilermakers: 25

Mobile phone counts:

- alcatel: 3
- Sony: 3
- Samsung: 3
- Motorola: 3
- LG: 2
- vivo: 1
- Yezz: 1
- TCL: 1
- Spice: 1
- Sagem: 1
- QMobile: 1
- Nokia: 1
- Micromax: 1
- Lenovo: 1
- HTC: 1
- BLU: 1

```

Now, I will calculate the percentage for each:
- alcatel: (3 / 25) * 100 = 12%
- Sony: (3 / 25) * 100 = 12%
- Samsung: (3 / 25) * 100 = 12%
- Motorola: (3 / 25) * 100 = 12%
- LG: (2 / 25) * 100 = 8%
- vivo: (1 / 25) * 100 = 4%
- Yezz: (1 / 25) * 100 = 4%
- TCL: (1 / 25) * 100 = 4%
- Spice: (1 / 25) * 100 = 4%
- Sagem: (1 / 25) * 100 = 4%
- QMobile: (1 / 25) * 100 = 4%
- Nokia: (1 / 25) * 100 = 4%
- Micromax: (1 / 25) * 100 = 4%
- Lenovo: (1 / 25) * 100 = 4%
- HTC: (1 / 25) * 100 = 4%
- BLU: (1 / 25) * 100 = 4%

```

```

14 ✓ ✓ select sample.Mobile, (count(sample.Mobile) * 100 / (select Count(*) from emp where emp_job='Boilermaker'))
15     as mob_freq from sample natural join emp where emp_job='Boilermaker' group by sample.Mobile order by mob_freq desc;
16
17

```

services

	Mobile	mob_freq
1	alcatel	12
2	Sony	12
3	Samsung	12
4	Motorola	12
5	LG	8
6	vivo	4
7	Yezz	4
8	TCL	4
9	Spice	4
10	Sagem	4
11	QMobile	4
12	Nokia	4
13	Micromax	4
14	Lenovo	4
15	HTC	4
16	BLU	4

16 rows | :

(10 – 12 reruns)

5.

```
● Your question: What is the ratio of employees having a Boilermaker job who drive a BMW to those who drive a Kia ?  
Processing...  
[Step 2 Observation] ['emp', 'sample']  
[Step 3 Observation] {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}, {"row_count": 1000}  
[Step 4 Observation] {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}, {"row_count": 1000}  
Note: Added LIMIT 100 to query for safety  
[Step 5 Observation] {"columns": ["bmw_drivers", "kia_drivers"], "rows": [[1, 1]], "row_count": 1}  
[Warning] No new evidence gathered this turn before FINAL ANSWER.  
  
FINAL ANSWER:  
The ratio of employees having a Boilermaker job who drive a BMW to those who drive a Kia is 1:1.  
  
FULL TRACE:  
  
--- Step 1 ---  
THOUGHT: I have identified the tables. Now I need to examine the schema of both tables to find columns related to 'job' and 'car'. I'll start with the 'sample' table.  
ACTION: N/A  
OBSERVATION: You are answering without running any tool yet. Please gather evidence using one ACTION before concluding.  
  
--- Step 2 ---  
THOUGHT: I need to find out which tables are available in the database to understand where employee job and car information might be stored.  
ACTION: list_tables()  
OBSERVATION: ['emp', 'sample']  
  
--- Step 3 ---  
THOUGHT: (none)  
ACTION: describe_table('table name': 'emp')  
OBSERVATION: {"table_name": "emp", "columns": [{"name": "id", "type": "INT"}, {"name": "emp_uni", "type": "VARCHAR(50)"}, {"name": "emp_job", "type": "VARCHAR(50)"}, {"name": "emp_role", "type": "VARCHAR(50)"}, {"name": "emp_bank", "type": "VARCHAR(50)"}, {"row_count": 1000}  
  
--- Step 4 ---  
THOUGHT: I have already described the 'emp' table and found the 'emp_job' column. Now I need to describe the 'sample' table to find car information and a common column for joining.  
ACTION: describe_table('table_name': 'sample')  
OBSERVATION: {"table_name": "sample", "columns": [{"name": "id", "type": "INT"}, {"name": "first_name", "type": "VARCHAR(50)"}, {"name": "last_name", "type": "VARCHAR(50)"}, {"name": "email", "type": "VARCHAR(50)"}, {"name": "gender", "type": "VARCHAR(50)"}, {"name": "city", "type": "VARCHAR(50)"}, {"name": "Mobile", "type": "VARCHAR(50)"}, {"name": "Car", "type": "VARCHAR(50)"}, {"name": "Credit_Card", "type": "VARCHAR(50)"}, {"name": "Favorite_Movie", "type": "VARCHAR(50)"}, {"row_count": 1000}  
  
--- Step 5 ---  
THOUGHT: I need to calculate two counts:  
1. The number of employees with the job 'Boilermaker' who drive a 'BMW'.  
2. The number of employees with the job 'Boilermaker' who drive a 'Kia'.  
Then, I will calculate the ratio of these two numbers. I will use a single query to get both counts.  
ACTION: query_database('query': "SELECT SUM(CASE WHEN T2.Car = 'BMW' THEN 1 ELSE 0 END) AS bmw_drivers, SUM(CASE WHEN T2.Car = 'Kia' THEN 1 ELSE 0 END) AS kia_drivers FROM emp AS T1 JOIN sample AS T2 ON T1.id = T2.id WHERE T1.emp_job = 'Boilermaker';")  
OBSERVATION: {"columns": ["bmw_drivers", "kia_drivers"], "rows": [[1, 1]], "row_count": 1}  
  
AGENT LOGS:  
[STEP 0] THOUGHT: I have identified the tables. Now I need to examine the schema of both tables to find columns related to 'job' and 'car'. I'll start with the 'sample' table.  
[STEP 1] THOUGHT: I need to find out which tables are available in the database to understand where employee job and car information might be stored.  
[STEP 2] THOUGHT: (none)  
[STEP 3] THOUGHT: I have already described the 'emp' table and found the 'emp_job' column. Now I need to describe the 'sample' table to find car information and a common column for joining.  
[STEP 4] THOUGHT: I need to calculate two counts:  
1. The number of employees with the job 'Boilermaker' who drive a 'BMW'.  
2. The number of employees with the job 'Boilermaker' who drive a 'Kia'.  
Then, I will calculate the ratio of these two numbers. I will use a single query to get both counts.  
[STEP 5] THOUGHT: (none)
```

```
select Car, COUNT(Car) from sample natural join emp where emp_job = 'Boilermaker'  
and (Car='Kia' or car='BMW') group by Car ;
```

Output Result 177 ×

Car	COUNT(Car)
BMW	1
Kia	1

(6-7 reruns)