SQL Agent

This app allows you to interact with an SQL database using a generative AI agent.

Powered by watsonx.ai.

User query

What is the total sales revenue for the top 5 performing dealerships in the year 2022?

Invoke

User Message:

```
What is the total sales revenue for the top 5 performing dealerships in the year 2
```

AI Response:

Tool Name: sql_db_list_tables

Arguments: {'tool_input': ''}

Tool Response:

```
Cars, Dealerships, Sales
```

AI Response:

Tool Name: sql_db_schema

Arguments: {'table_names': 'Cars, Dealerships, Sales'}

Tool Response:

```
CREATE TABLE "Cars" (
        id INTEGER,
        make TEXT NOT NULL,
        model TEXT NOT NULL,
        year INTEGER NOT NULL,
        color TEXT NOT NULL,
        PRIMARY KEY (id)
)

/*
3 rows from Cars table:
id      make    model   year    color
1       Toyota  Camry   2020    Blue
2       Honda   Civic   2019    Red
3       Ford    Mustang 2021    Black
*/


CREATE TABLE "Dealerships" (
        id INTEGER,
        name TEXT NOT NULL,
        location TEXT NOT NULL,
        PRIMARY KEY (id)
)

/*
3 rows from Dealerships table:
id      name    location
1       Dealership A    New York
2       Dealership B    Los Angeles
3       Dealership C    Chicago
*/


CREATE TABLE "Sales" (
        id INTEGER,
        car_id INTEGER NOT NULL,
        dealership_id INTEGER NOT NULL,
        sale_date DATE NOT NULL,
        sale_price REAL NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY(dealership_id) REFERENCES "Dealerships" (id),
        FOREIGN KEY(car_id) REFERENCES "Cars" (id)
)

/*
3 rows from Sales table:
id      car_id  dealership_id   sale_date       sale_price
1       1       1       2022-01-01      25000.0
2       2       2       2022-02-01      20000.0
3       3       3       2022-03-01      30000.0
*/
```

AI Response:

Tool Name: sql_db_query_checker

Arguments: {'query': "SELECT Dealerships.name, SUM(sale_price) AS total_revenue\r\nFROM Sales\r\nJOIN Dealerships on Sales.dealership_id = Dealerships.id\r\nWHERE sale_date BETWEEN '2022-01-01' AND '2022-12-31'\r\nGROUP BY Dealerships.id\r\nORDER BY total_revenue DESC LIMIT 5;\r\n"}

Tool Response:

~~~
The original query looks correct and doesn't have any of the common mistakes menti

```sql
SELECT Dealerships.name, SUM(sale_price) AS total_revenue
FROM Sales
JOIN Dealerships on Sales.dealership_id = Dealerships.id
WHERE sale_date BETWEEN '2022-01-01' AND '2022-12-31'
GROUP BY Dealerships.id
ORDER BY total_revenue DESC LIMIT 5;
```
~~~

AI Response:

Tool Name: sql_db_query

Arguments: {'query': "SELECT Dealerships.name, SUM(sale_price) AS total_revenue\r\nFROM Sales\r\nJOIN Dealerships on Sales.dealership_id = Dealerships.id\r\nWHERE sale_date BETWEEN '2022-01-01' AND '2022-12-31'\r\nGROUP BY Dealerships.id\r\nORDER BY total_revenue DESC LIMIT 5;\r\n"}

Tool Response:

```
[('Dealership A', 65000.0), ('Dealership B', 48000.0), ('Dealership J', 38000.0),
```

AI Response:

```
The top 5 performing dealerships in terms of total sales revenue for the year 2022

1. Dealership A with a total revenue of $65,000.
2. Dealership B with a total revenue of $48,000.
3. Dealership J with a total revenue of $38,000.
4. Dealership G with a total revenue of $35,000.
5. Dealership H with a total revenue of $32,000.
```

✓ Success

Start new chat