使用速览

# **Start** the interactive **SQL** shell:

cockroach **sql** <flags>

# **Execute** **SQL** **from** the command line:

cockroach **sql** *--execute="<sql statement>;<sql statement>" --execute="<sql-statement>" <flags>*

echo "<sql statement>;<sql statement>" | cockroach **sql** <flags>

cockroach **sql** <flags> < file-containing-statements.**sql**

# **View** **help**:

cockroach **sql** *--help*

命令flags

通用

| **Flag** | **简介** |
| --- | --- |
| --database -d | 指定此次session需要使用的database名称 |
| --echo-sql | （new in v1.1）显示成功发送的sql语句，交互式shell当中则可以使用语句\set echo启动该功能 |
| --execute -e | 直接执行sql语句并返回结果到标注输出，语句之间用分号隔开 |
| --format | 显示指定标准输出的格式，包括：tsv、csv、pretty、raw、records、sql、html |
| --safe-updates | （changed in v2.0）禁用使用高危SQL语句，包括：（1）没有WHERE子句的DELETE、UPDATE语句；（2）ALTER TABLE ... DROP COLUMN  **默认情况:** 使用交互式session时为true，其他情况为false 此外，也可以设置session变量开启safe updates功能，详见：[session variable](http://doc.cockroachchina.baidu.com/#quick-start/learn-cockroachdb-sql/use-the-built-in-sql-client/). |

客户端连接相关

| **Flag** | **简介** |
| --- | --- |
| --host | 指定服务器ip地址，可以是集群的任意节点。  **环境变量**：COCKROACH\_HOST **默认**：localhost |
| --port -p | 指定服务器端口  **环境变量**：COCKROACH\_PORT **默认**：26257 |
| --user -u | 指定[SQL用户](http://doc.cockroachchina.baidu.com/#deploy/access-management/manage-users/)  **环境变量**：COCKROACH\_USER **默认**：root |
| --insecure | 使用非安全模式的连接  **环境变量**：COCKROACH\_INSECURE **默认**：false |
| --certs-dir | 指定存放着[CA证书和客户端证书密钥的目录](http://doc.cockroachchina.baidu.com/#deploy/create-security-certificates/)  **环境变量**：COCKROACH\_CERTS\_DIR **默认**：${HOME}/.cockroach-certs/ |
| --url | 指定[connection url](http://doc.cockroachchina.baidu.com/#develop/client-connection-parameters/#connect-using-a-url)  **环境变量**：COCKROACH\_URL **默认**：空 |

更多细节可以查看：[客户端连接参数](http://doc.cockroachchina.baidu.com/#develop/client-connection-parameters/)

日志相关

cockroach sql默认输出日志错误到stderr，更丰富的日志功能可见：[日志行为](https://www.cockroachlabs.com/docs/v2.0/debug-and-error-logs)

交互式shell

启动输出内容

示例：

**# Welcome to the cockroach SQL interface.**

**# All statements must be terminated by a semicolon.**

**# To exit: CTRL + D.**

**#**

**# Server version: CCL v2.0.5 (darwin amd64, built 2017/07/13 11:43:06, go1.8) (same version as client)**

**# Cluster ID: 7fb9f5b4-a801-4851-92e9-c0db292d03f1**

**#**

**# Enter \? for a brief introduction.**

**#**

* 输出包括服务器cockroachdb版本，当客户端与服务器版本一致时会显示“same version as client”，当出现版本不一致时需要同步版本；
* 输出包括当前session连接服务器的cluster id；

部分参数默认设置

flags:

* --format=pretty

option参数：

* errexit = false
* check\_syntax = true
* smart\_prompt = true

内部操作指令

| **Command** | **Usage** |
| --- | --- |
| \q ctrl-d | 退出shell |
| \! | 执行外部cmd命令，并输出cmd命令运行结果 |
| \| | 执行外部cmd命令，再将输出结果作为sql语句执行，输出sql语句运行结果 |
| \set <option> | 设置客户端option参数，若不带option参数则显示当前所有option参数值 |
| \unset <option> | 使得某一option参数失效 |
| \? help | 获取帮助 |
| \h <statement> \hf <function> | （new in v1.1）针对某些statement语句或是function的使用获取帮助 |

option参数

| **option参数** | **简介** |
| --- | --- |
| display\_format | （updated in v2.0）指定输出格式，默认pretty，可选tsv、csv、pretty、raw、records、sql、html |
| echo | （new in v1.1） 显示成功发送的sql语句，默认false |
| errexit | 是否异常退出，默认false |
| check\_syntax | 检查sql语句语法，默认true |
| show\_times | （new in v1.1）输出sql语句执行时间，默认true |
| smart\_prompt | 输出当前在服务器端执行的事务状态，该option需要echo同时生效，默认true |

更多更丰富的快捷键

更多更丰富的快捷键，用户可查看文档：[更多快捷键](https://github.com/chzyer/readline/blob/master/doc/shortcut.md)

示例：

启动交互式shell

# Using standard connection flags:

cockroach sql *--certs-dir=certs --user=maxroach --host=12.345.67.89 --port=26257 --database=critterdb*

# Using the *--url flag:*

cockroach sql *--url="postgresql://maxroach@12.345.67.89:26257/critterdb?sslcert=certs/client.maxroach.crt&sslkey=certs/client.maxroach.key&sslmode=verify-full&sslrootcert=certs/ca.crt"*

# Using standard connection flags:

cockroach sql *--insecure --user=maxroach --host=12.345.67.89 --port=26257 --database=critterdb*

# Using the *--url flag:*

$ cockroach sql *--url="postgresql://maxroach@12.345.67.89:26257/critterdb?sslmode=disable"*

在交互式shell当中执行sql语句

**CREATE** **TABLE** animals (id SERIAL **PRIMARY** **KEY**, name **STRING**);

**INSERT** **INTO** animals (name) **VALUES** ('bobcat'), ('乌龟'), ('barn owl');

**SELECT** \* **FROM** animals;

+*--------------------+----------+*

| id | name |

+*--------------------+----------+*

| 148899952591994881 | bobcat |

| 148899952592060417 | 乌龟 |

| 148899952592093185 | barn owl |

+*--------------------+----------+*

在cmd界面中执行sql语句

# Statements with a single *--execute flag:*

cockroach sql *--insecure --execute="CREATE TABLE roaches (name STRING, country STRING); INSERT INTO roaches VALUES ('American Cockroach', 'United States'), ('Brownbanded Cockroach', 'United States')" --user=maxroach --host=12.345.67.89 --port=26257 --database=critterdb*

# Statements with multiple *--execute flags:*

cockroach sql *--insecure --execute="CREATE TABLE roaches (name STRING, country STRING)" --execute="INSERT INTO roaches VALUES ('American Cockroach', 'United States'), ('Brownbanded Cockroach', 'United States')" --user=maxroach --host=12.345.67.89 --port=26257 --database=critterdb*

# Statements with the echo command:

echo "**SHOW** **TABLES**; **SELECT** \* **FROM** roaches;" | cockroach sql *--insecure --user=maxroach --host=12.345.67.89 --port=26257 --database=critterdb*

+*----------+*

| Table |

+*----------+*

| roaches |

+*----------+*

+*-----------------------+---------------+*

| name | country |

+*-----------------------+---------------+*

| American Cockroach | United States |

| Brownbanded Cockroach | United States |

+*-----------------------+---------------+*

控制输出格式

cockroach sql *--insecure --format=html --execute="SELECT 'chicken' AS chick, 'dark' AS turtle" --user=maxroach --host=12.345.67.89 --port=26257 --database=critterdb*

<table>

<thead><tr><th>chick</th><th>turtle</th></tr></head>

<tbody>

<tr><td>chicken</td><td>dark</td></tr>

</tbody>

</table>

执行文件当中的sql语句

cockroach sql *--insecure --user=maxroach --host=12.345.67.89 --port=26257 --database=critterdb < statements.sql*

交互式shell - 执行外部cmd命令

\! cat test.csv

交互式shell - 执行外部cmd命令，并将输出结果作为sql语句运行

**CREATE** **TABLE** for\_loop (x INT);

\| for ((i=0;i<10;++i)); **do** echo "INSERT INTO for\_loop VALUES ($i);"; done

**SELECT** \* **FROM** for\_loop;

+*---+*

| x |

+*---+*

| 0 |

| 1 |

| 2 |

| 3 |

| 4 |

| 5 |

| 6 |

| 7 |

| 8 |

| 9 |

+*---+*

safe-update相关

阻止不安全sql语句执行，默认为true

cockroach sql --insecure --execute="DELETE FROM db1.t1"

Error: pq: rejected: DELETE without WHERE clause (sql\_safe\_updates = true)

Failed running "sql"

显示成功发送的sql语句

cockroach sql --insecure --execute="CREATE TABLE t1 (id INT PRIMARY KEY, name STRING)" --execute="INSERT INTO t1 VALUES (1, 'a'), (2, 'b'), (3, 'c')" --user=maxroach --host=12.345.67.89 --port=26257 --database=db1 --echo-sql

*# Server version: CockroachDB CCL f8f3c9317 (darwin amd64, built 2017/09/13 15:05:35, go1.8) (same version as client)*

*# Cluster ID: 847a4ba5-c78a-465a-b1a0-59fae3aab520*

> SET sql\_safe\_updates = TRUE

> CREATE TABLE t1 (id INT PRIMARY KEY, name STRING)

CREATE TABLE

> INSERT INTO t1 VALUES (1, 'a'), (2, 'b'), (3, 'c')

INSERT 3