



链滴

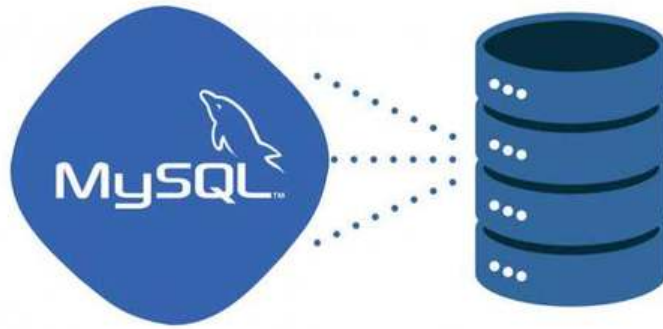
如何在 MySQL / MariaDB 中导入导出数据 ， 导入导出数据库文件、Excel、CSV

作者: [HiJiangChuan](#)

原文链接: <https://ld246.com/article/1636539130941>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



如何在 MySQL 或 MariaDB 中 导入和导出数据 SQL、CSV、EXCEL

卡拉云 kalacloud.com

在日常的数据库维护工作中，经常需要对数据库进行导入导出操作，备份、分析、迁移数据都需要用导入导出功能，在本教程中将详细讲解所有常见的 MySQL 和 MariaDB 中导入导出数据的方法（注：MySQL 和 MariaDB 两个数据库操作命令一样，可以互换。）

本教程将详细讲解

1. MySQL / MariaDB 数据库数据「导出」

- (1) 使用 `mysqldump` 直接导出数据至 SQL 文件
- (2) 阿里云 / 腾讯云远程服务器中的数据库直接导出到本地计算机
- (3) 使用 `into outfile` 命令导出数据至 CSV / Excel

提示：如果你正在寻找数据迁移方案，请查看我写的另一篇专门针对 [MySQL 数据迁移的教程](#)，教中包含腾讯云、阿里云迁移实战。

2. MySQL / MariaDB 数据库数据「导入」

- (1) 将 SQL 文件导入至 MySQL / MariaDB 数据库中
- (2) 使用 `source` 导入数据库 SQL 文件
- (3) 将 CSV / Excel 文件 导入至 MySQL / MariaDB 数据库中

3. 使用「卡拉云」一键导入导出 MySQL / MariaDB 数据

如何使用卡拉云，5分钟搭建一套适应自己工作流的一键导入导出数据库系统。卡拉云无需部署，即

即用，可根据需求灵活调配，适用于后端工程师快速搭建企业内部系统、数据产品经理查看分析数据、数据分析师根据需求快速搭建数据共享平台分享给组内同学协同查看等应用场景。点这里查看详情。

4. 先决条件

跟随本教程学习如何导入导出 MySQL 或 MariaDB 数据库，首先要有

- 一台 Linux 服务器，本文以 Ubuntu 为例
- 已安装 MySQL 或 MariaDB server（还未安装，安装教程请看这篇《[MySQL 安装教程](#)》）
- MySQL 或 MariaDB Server 中有数据库（用于导出）
- 教程使用 MacOS 演示本地计算机操作，此操作同时适用于 Windows 及 Linux

一. 导出 MySQL 或 MariaDB 数据库

1. 如何使用 `mysqldump` 导出数据

`mysqldump` 命令是数据库导出中使用最频繁的一个工具，它可将数据库中的数据备份成已 *.sql 结尾的文本文件，表结构和数据都会存储在其中。

`mysqldump` 命令的原理也很简单，它先把需要备份的表结构查询出来，然后生成一个 `CREATE TABLE 'table'` 语句，最后将表中所有记录转化成一条 `INSERT` 语句。

可以把它理解为一个批量导出导入脚本。数据导入时，按照规范语句导入数据，大幅减少奇怪的未知错误出现。

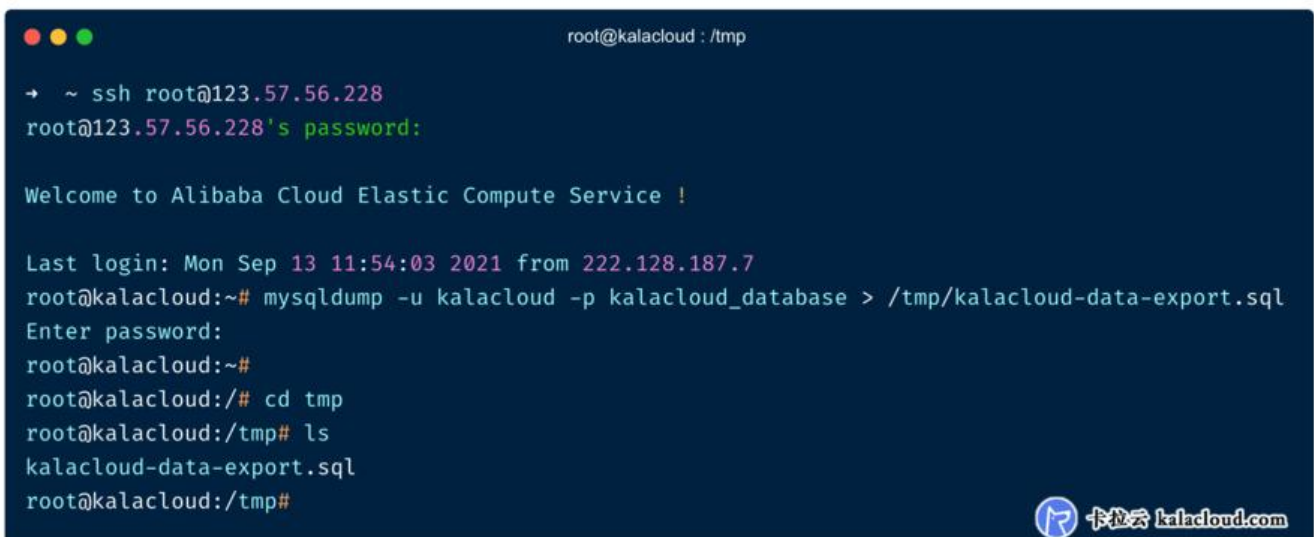
`mysqldump` 的基本命令：

```
$ mysqldump -u username -p database_name > data-dump.sql
```

- `username` 是数据库的登录名
- `database_name` 是需要导出的数据库名称
- `data-dump.sql` 是文件输出目录的文件

导出实战 - 从阿里云服务器中的 MySQL 数据库导出数据

```
$ mysqldump -u kalacloud -p kalacloud_database > /tmp/kalacloud-data-export.sql
```



```
root@kalacloud : /tmp
→ ~ ssh root@123.57.56.228
root@123.57.56.228's password:

Welcome to Alibaba Cloud Elastic Compute Service !

Last login: Mon Sep 13 11:54:03 2021 from 222.128.187.7
root@kalacloud:~# mysqldump -u kalacloud -p kalacloud_database > /tmp/kalacloud-data-export.sql
Enter password:
root@kalacloud:~#
root@kalacloud:~# cd tmp
root@kalacloud:~/tmp# ls
kalacloud-data-export.sql
root@kalacloud:~/tmp#
```

- `kalacloud` : 数据库账号
- `kalacloud_database` : 数据库名
- `/tmp/kalacloud-data-export.sql` : 数据库导出的文件及存放目录

输入数据库 `kalacloud` 账号的密码执行命令，如果执行过程中，没有任何错误，那么命令行不会有任输出。

我们可以 `cd` 到 `tmp` 目录查看结果。上图可以看到，`tmp` 目录下已经生成 `kalacloud-data-export.sql` 的导出文件。

我们在用 `head -n 5 kalacloud-data-export.sql` 命令检查一下。你会看到类似下图的内容。

```
root@kalacloud : /tmp

root@kalacloud:~/tmp# head -n 5 kalacloud-data-export.sql
-- MySQL dump 10.13  Distrib 8.0.26, for Linux (x86_64)
--
-- Host: localhost      Database: kalacloud_database
--
-- Server version      8.0.26-0ubuntu0.20.04.2
root@kalacloud:~/tmp#
```

至此，我们已经将指定数据库导出到 `*.sql` 文件中了，后文我们讲解如何将这些数据导入到数据库。

进阶提示：我们可以使用 `scp` 命令，将导出文件下载至本地计算机。

在本地计算机的命令行终端里，输入：

```
scp root@192.168.180.134:/tmp/kalacloud-data-export.sql /Users/kalacloud/Downloads
```

`root` 远程计算机的登录账号

`192.168.180.134` 为远程计算机的 IP 地址

`/tmp/kalacloud-data-export.sql` 为需要下载到本地的数据库文件在远程计算机上的存储位置

`/Users/kalacloud/Downloads` 为本地计算机的存储位置，远程文件将下载到这个目录中

```
卡拉云 kalacloud.com

→ ~ scp root@123.57.56.228:/tmp/kalacloud-data-export.sql /Users/kalacloud/Downloads
root@123.57.56.228's password:

kalacloud-data-export.sql      100% 2110    876.8KB/s   00:00
→ ~
```

使用 `scp` 将导出的 SQL 文件下载到本地再进行后续处理。当然我们也可以一步导出至本地计算机，面我们继续讲解进阶导出方法。

扩展阅读：《[如何远程连接 MySQL 数据库](#)，[阿里云腾讯云外网连接教程](#)》

2.进阶：将阿里云 / 腾讯云远程服务器中的数据库导出到本地计算机

前文我们讲了如何在远程服务器上操作导出数据库，导出后保存在远程服务器中。有时我们需要把数导出给产品或运营进行数据分析，又或者我们使用的云服务是独立 MySQL 数据库，这时，你需要直接把数据导出到本地计算机中。

```
$ mysqldump -h remote_IP_address -u username -p -P3306 --default-character-set=utf8 --set-gtid-purged=OFF database_name >/Users/kalacloud/Desktop/data-dump.sql
```

- `remote_IP_address` :远程服务器的 IP
- `username` : 拥有远程登录权限的 MySQL 账号
- `3306`: 远程登录的数据库端口，默认是 3306，如果不是可根据情况替换
- `default-character-set=utf8` : 导出时指定字符集
- `set-gtid-purged=OFF` : 全局事务 ID (GTID) 来强化数据库的主备一致性，故障恢复，以及容错力。开启这个功能导入导出时，可能会出错，故关闭。
- `database_name` : 需要导出的数据库名称
- `/Users/kalacloud/Desktop/data-dump.sql` : 本地计算机保存路径及保存文件名

提示：`mysqldump`常见报错：`mysqldump: Couldn't execute 'SELECT COLUMN_NAME, JSON_E TRACT(HISTOGRAM, '$.number-of-buckets-specified')'`

可在命令中添加 `column-statistics=0` 参数。因 MySQL 数据库早期版本 `information_schema` 数据中没有名为 `COLUMN_STATISTICS` 的数据表，新版 `mysqldump` 默认启用，我们可以通过此命令禁它。

导出实战 - 将阿里云服务器中的数据库直接导出到本地计算机

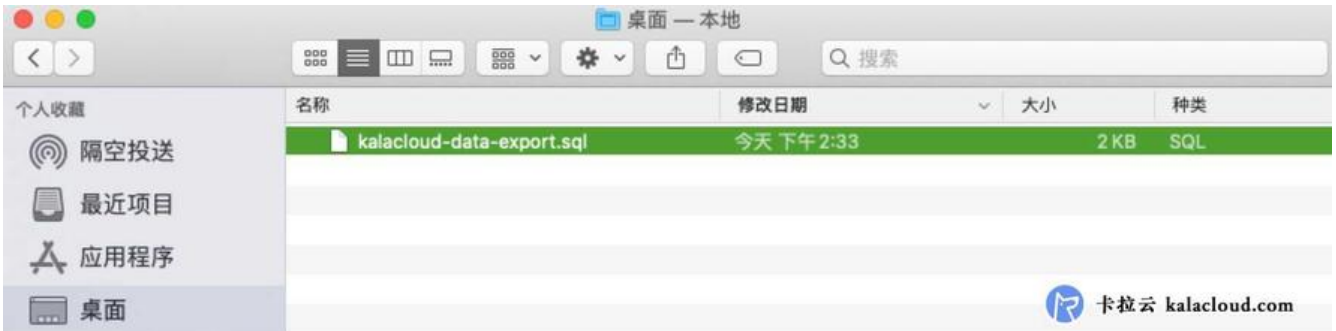
```
$ mysqldump -h123.57.56.228 -ukalacloud-remote -p -P3306 --default-character-set=utf8 --set-gtid-purged=OFF --column-statistics=0 kalacloud_database >/Users/kalacloud/Desktop/kalacloud-data-export.sql
```

- `123.57.56.228`: 远程数据库 ip 地址
- `kalacloud-remote`: 拥有远程访问权限的数据库账号。
- `-P 3306`: 数据库访问端口，可根据自己情况修改。
- `/Users/kalacloud/Desktop/kalacloud-data-export.sql` : 本地计算机保存路径及保存文件名



```
kalacloud@localhost:~  
→ ~ mysqldump -h123.57.56.228 -ukalacloud-remote -p -P3306 --default-character-set=utf8 --set-gtid-purged=OFF --column-statistics=0 kalacloud_database >/Users/kalacloud/Desktop/kalacloud-data-export.sql  
Enter password:  
→ ~
```


执行命令后，命令行并没有任何信息输出，但我们已经可以在桌面上看到导出后生成的文件了。



已经导出到本地桌面的远程端数据库

当然，`mysqldump` 也可以分表备份，比较常见的场景有

备份单个库

```
mysqldump -uroot -p -R -E --single-transaction --databases [database_one] > database_one.sql
```

备份部分表

```
mysqldump -uroot -p --single-transaction [database_one] [table_one] [table_two] > database_table12.sql
```

排除某些表

```
mysqldump -uroot -p [database_one] --ignore-table=[database_one.table_one] --ignore-table=[database_one.table_two] > database_one.sql
```

只备份结构

```
mysqldump -uroot -p [database_one] --no-data > [database_one.defs].sql
```

只备份数据

```
mysqldump -uroot -p [database_one] --no-create-info > [database_one.data].sql
```

扩展阅读：有关数据库在两台服务器之间迁移的问题可看我写的《[如何迁移 MySQL 数据库，阿里云腾讯云迁移案例](#)》

3. 使用 `into outfile` 命令导出 MySQL / MariaDB 数据至 CSV / Excel

有时我们需要将数据导出给运营或产品进行数据分析，这时导出 CSV 文件会更加方便使用。

```
mysql> select * from users into outfile '/var/lib/mysql-files/users.csv' FIELDS TERMINATED BY ',';
```

`FIELDS TERMINATED BY ','` 数据以, 进行分隔。

首先我们登录 MySQL shell，选择需要导出的数据库 `use kalacloud_database`；然后执行导出命令。

```
mysql> use kalacloud_database;
Database changed

mysql> select * from users into outfile '/var/lib/mysql-files/kalacloud_data.csv' FIELDS TERMINATED BY
Query OK, 4 rows affected (0.00 sec)

mysql> ^DBye
root@kalacloud:~# cd /var/lib/mysql-files/
root@kalacloud:/var/lib/mysql-files# ls
kalacloud_data.csv
```

导出后会显示成功提示，**CD** 到导出目录可看到 CSV 文件已导出。

提示：**into outfile** 常见报错

ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it can not execute this statement

这是因为你的 MySQL 配置了**--secure-file-priv** 限制了导出文件的存放位置。

你可以使用以下命令来查看具体配置信息

show global variables like '%secure_file_priv%';

```
[
mysql> show global variables like '%secure_file_priv%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| secure_file_priv | /var/lib/mysql-files/ |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

secure_file_priv 为 NULL 时，表示不允许导入或导出。**secure_file_priv** 为路径时 (/var/lib/mysql-files/) 时，表示只允许在路径目录中执行。**secure_file_priv** 没有值时，表示可在任意目录的导入导出。

你可以打开 my.cnf 或 my.ini，添加以下语句，重启 MySQL server 即可

secure_file_priv=""

扩展阅读：有关把 MySQL 查询出来的结果保存到文件可看我写的这篇《[如何在 MySQL 中保存查询果到文件](#)》教程。

二. MySQL 或 MariaDB 数据库导入数据

接着我们讲解如何将 *.sql 导入到数据库中。我们先建一个新数据库用作演示。

我们以 root 或有足够权限的账号登录 MySQL:

```
$ mysql -u root -p
```

输入登录密码后, 进入 MySQL shell 状态。接着我们创建一个新数据库, 在这个例子中, 我们用 `kalacloud_new_database` 作为新数据库名称。

```
mysql> CREATE DATABASE kalacloud_new_database;
```

执行命令后返回内容

```
Query OK, 1 row affected (0.00 sec)
```

用于演示的新数据库创建完成, 我们使用 `CTRL+D` 退出 MySQL shell

1. 直接使用 `mysql` 导入 SQL 文件

在命令行中我们导入上文导出的 `/tmp/kalacloud-data-export.sql` 文件 (注意: 以下命令在**命令行执行**, 不是在 `mysql>` 状态下执行)

```
mysql -u root -p kalacloud_new_database < /tmp/kalacloud-data-export.sql
```

- `root`: 你可以登录数据库的用户名。
- `kalacloud_new_database`: 刚刚新建的空数据库, 这条命令会把数据导入到这其中。
- `/tmp/kalacloud-data-export.sql`: 是上文我们从数据库导出的 sql 文件, 这里我们把它再导入到数据库中。

如果运行成功, 命令行不会有任何提示。如果运行失败, 命令行会提示失败原因。要检测是否导入成功, 我们可以登录到 MySQL 查看并检查数据库中的数据。


```
root@kalacloud:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Server version: 8.0.26-0ubuntu0.20.04.2 (Ubuntu)

mysql> use kalacloud_new_database;

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_kalacloud_new_database |
+-----+
| users                             |
+-----+
1 row in set (0.00 sec)

mysql> select * from users;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name   | phone   | states | file_size | sale | copyright | homepage | complaint |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | 宋阿美 | 13109591777 | 2     | 100      | 77  | 10        | songamei | 0         |
| 2  | 吕小果 | 13251115861 | 1     | 1        | 15  | 20        | lvxiaoguo | 2         |
| 3  | 谢国庆 | 13374267337 | 2     | 1        | 99  | 1         | xieguoqing | 3         |
| 4  | 蒋小喵 | 13661131133 | 2     | 100      | 99  | 20        | JiangChuan | 1         |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```



登录 MySQL server , 使用 `USE kalacloud_new_database;` 选择刚刚我们导入数据的新建数据库, 后使用 `SHOW TABLES;` 查看数据库中包含的表, 最后用 `SELECT * FROM users;` 打开表查看内容。

扩展阅读: 《[MySQL 配置文件逐行解析](#)》教程

2.使用 `source` 导入 MySQL / MariaDB 数据库 SQL 文件

进入 MySQL shell 状态, 我们还是导入本教程前文导出的 `/tmp/kalacloud-data-export.sql` 文件, 新数据库中。

```
mysql> USE kalacloud_new_database;
```

首先选择需要导入的数据库 `kalacloud_new_database`, 返回结果。

```
Database changed
```

然后使用 `source`

```
mysql> source /tmp/kalacloud_new_database.sql;
```

```
mysql> USE kalacloud_new_database;
Database changed

mysql> source /tmp/kalacloud-datebae.sql;
Query OK, 0 rows affected (0.00 sec)
...
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

执行 `source` 命令后，MySQL 开始执行导入，接着我们使用 `SHOW TABLES` 和 `select` 来查看 SQL 文件是否导入正常。

```
mysql> USE kalacloud_new_database;
Database changed

mysql> source /tmp/kalacloud-datebae.sql;
Query OK, 0 rows affected (0.00 sec)
...
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> SHOW TABLES;
+-----+
| Tables_in_kalacloud_new_database |
+-----+
| users                               |
+-----+
1 row in set (0.00 sec)

mysql> select * from users;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name   | phone      | states | file_size | sale | copyright | homepage | complaint |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | 宋阿美 | 13109591777 | 2      | 100       | 77  | 10        | songamei | 0         |
| 2  | 吕小果 | 13251115861 | 1      | 1         | 15  | 20        | lvxiaoguo | 2         |
| 3  | 谢国庆 | 13374267337 | 2      | 1         | 99  | 1         | xieguoqing | 3         |
| 4  | 蒋小喵 | 13661131133 | 2      | 100       | 99  | 20        | JiangChuan | 1         |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

上图可以看到，数据已经导入成功。

特别提示：`source` 和 `mysql <` 两种导入方式的区别

- **命令执行环境不同**：`source` 在 MySQL shell 里执行，`mysql <` 在终端命令行中执行

- **返回结果的不同**: `source` 会连续返回每一行导入结果, 如果量大可能会影响速度, `mysql <` 全部完成后返回结果。
- **报错是否停止执行**: `source` 遇到报错不会终止执行, `mysql <` 遇到报错会终止执行。

扩展阅读: 《[MySQL 触发器六种情况一次讲透, 应用实战案例](#)》

3.MySQL / MariaDB 数据库中导入 CSV 文件

除了直接导入 sql 类文件外, 有时候我们还会碰到需要导入 CSV 文件。导入 CSV 文件的步骤与直接导入 sql 有很大的不同, 接着我们来讲解如何导入 CSV 文件。

我们先进入 MySQL Shell:

```
$ mysql -uroot -p
```

然后新建一个空数据库:

```
mysql> CREATE DATABASE kalacloud_new_database;
```

与导入 *.sql 不同, 导入 CSV 文件需要先创建「表」, 我们需要根据 CSV 文件中包含的列, 使用 **CREATE TABLE** 创建表。

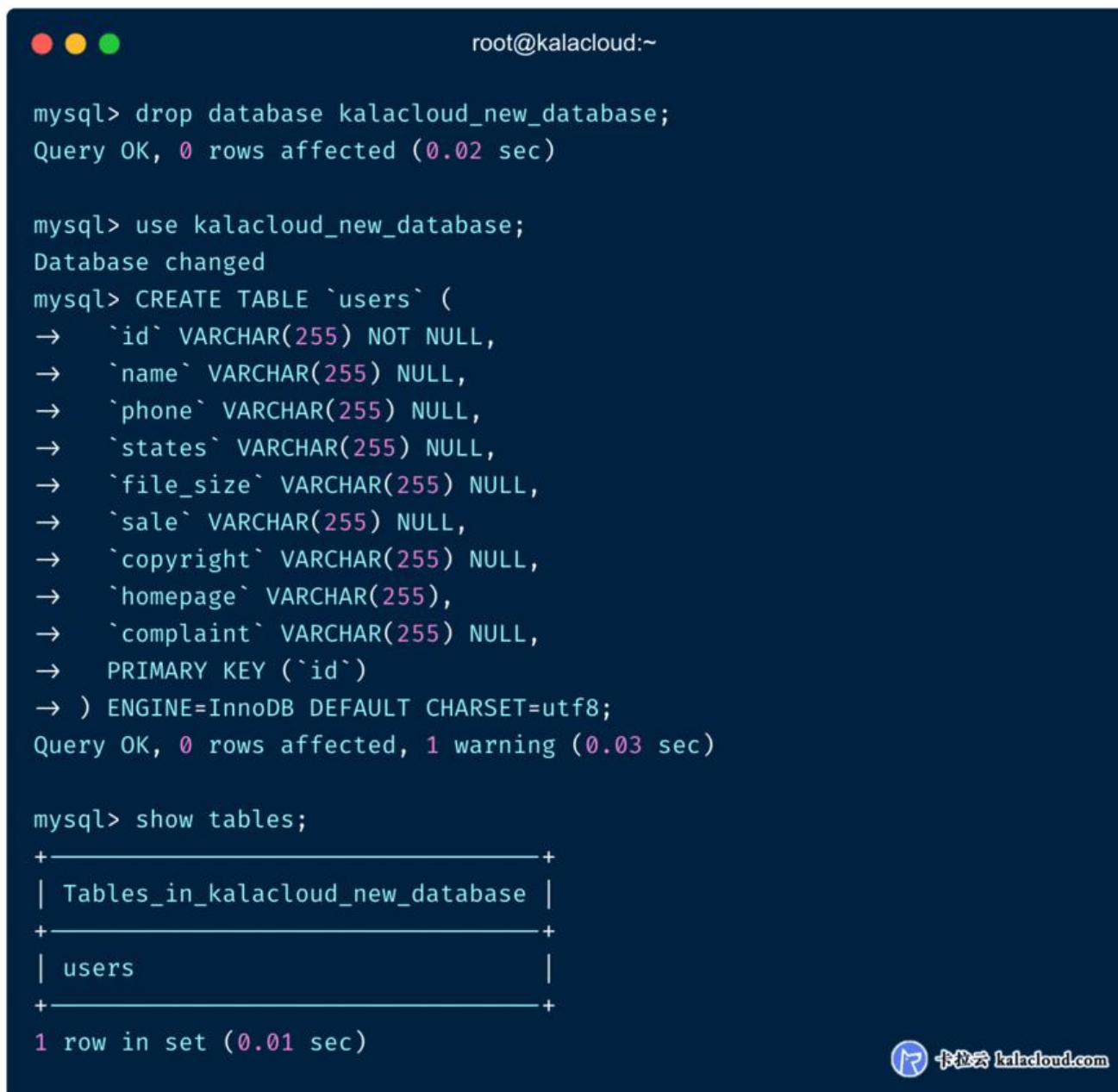
```
CREATE TABLE `users` (  
  `id` VARCHAR(255) NULL,  
  `name` VARCHAR(255) NULL,  
  `phone` VARCHAR(255) NULL,  
  `states` VARCHAR(255) NULL,  
  `file_size` VARCHAR(255) NULL,  
  `sale` VARCHAR(255) NULL,  
  `copyright` VARCHAR(255) NULL,  
  `homepage` VARCHAR(255),  
  `complaint` VARCHAR(255) NULL,  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

建议所有字段都设为接纳 **NULL** 值, 也暂时不要设置主键。因为我们并不知道即将导入的 CSV 文件的数据是否完整和规范。

建议即便是数字, 也先使用 **VARCHAR** 字段, 以防止文件中的数据格式不正确导致的奇怪错误。

我们可以在数据导入后, 在对数据库进行验证、清理和修正。

```
root@kalacloud:~  
  
mysql> drop database kalacloud_new_database;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> use kalacloud_new_database;  
Database changed  
mysql> CREATE TABLE `users` (  
→ `id` VARCHAR(255) NOT NULL,  
→ `name` VARCHAR(255) NULL,  
→ `phone` VARCHAR(255) NULL,  
→ `states` VARCHAR(255) NULL,  
→ `file_size` VARCHAR(255) NULL,  
→ `sale` VARCHAR(255) NULL,  
→ `copyright` VARCHAR(255) NULL,  
→ `homepage` VARCHAR(255),  
→ `complaint` VARCHAR(255) NULL,  
→ PRIMARY KEY (`id`)  
→ ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
Query OK, 0 rows affected, 1 warning (0.03 sec)  
  
mysql> show tables;  
+-----+  
| Tables_in_kalacloud_new_database |  
+-----+  
| users |  
+-----+  
1 row in set (0.01 sec)
```



上文中我们从 `kalacloud_database` 中导出了表 `users` 存放到了 `/var/lib/mysql-files/users.csv` 里，面我们使用 `LOAD DATA INFILE SQL` 语句把这个 CSV 文件导入新建的表中。

```
load data infile '/var/lib/mysql-files/users.csv'  
into table users  
FIELDS TERMINATED BY ','  
ENCLOSED BY '"';
```

字段使用逗号分隔，字符串用双引号括起来。如果你的 CSV 第一行是标题而非数据，那么还可以添加 `IGNORE 1 ROWS`；导入时，忽略第一行。

```
root@kalacloud:~  
mysql> load data infile '/var/lib/mysql-files/users.csv'  
  → into table users  
  → FIELDS TERMINATED BY ','  
  → ENCLOSED BY '';  
Query OK, 4 rows affected (0.01 sec)  
Records: 4 Deleted: 0 Skipped: 0 Warnings: 0  
  
mysql> select * from users;  
+----+-----+-----+-----+-----+-----+-----+-----+-----+  
| id | name  | phone | states | file_size | sale | copyright | homepage | complaint |  
+----+-----+-----+-----+-----+-----+-----+-----+-----+  
| 1  | 宋阿美 | 13109591726 | 2 | 100 | 77 | 10 | songamei | 0 |  
| 2  | 吕小果 | 13251115861 | 1 | 1 | 15 | 20 | lvxiaoguo | 2 |  
| 3  | 谢国庆 | 13374267337 | 2 | 1 | 99 | 1 | xieguoqing | 3 |  
| 4  | 蒋小喵 | 13661131133 | 2 | 100 | 99 | 20 | JiangChuan | 1 |  
+----+-----+-----+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```



导入成功后，使用 `select * from users;` 初步检查表中数据是否正确。

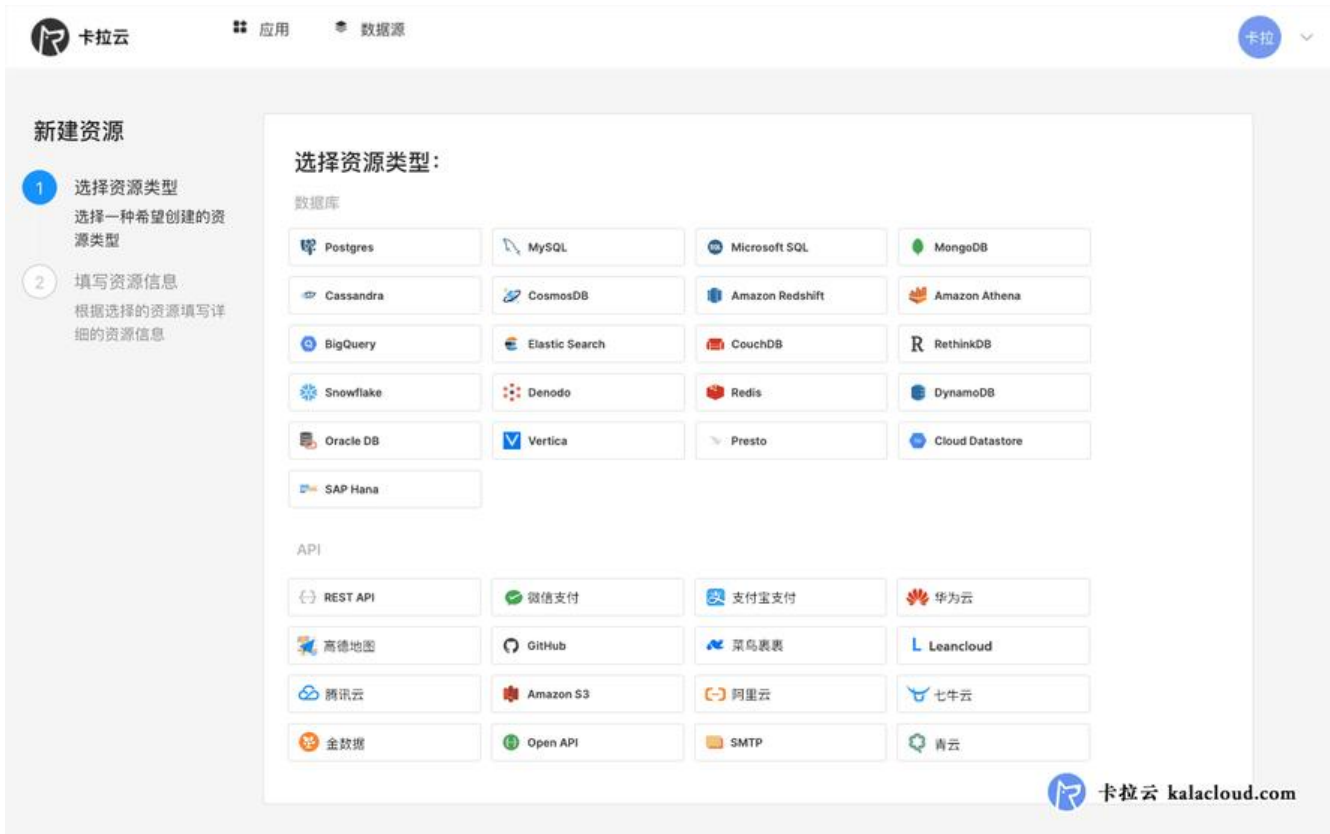
扩展阅读：我们也可以使用 Workbench 这种免费的 MySQL 图形管理工具来操作，了解更多可看我的这篇《[MySQL Workbench 中文指南](#)》教程。

三. 使用「卡拉云」一键导入导出数据

除了 MySQL / MariaDB 数据迁移这类适合使用终端命令操作外，大多数对 MySQL / MariaDB 数据导入导出操作还是为了数据展示、分析、协同共享等产品和运营层面的应用场景。

比如后端工程师接到产品需求，协助导出某类数据等场景，如果这类需求频繁出现，推荐使用卡拉云。卡拉云是新一代低代码开发工具，免安装部署，可一键接入包括 MySQL 在内的常见数据库及 API。

不仅可以像命令行一样灵活，还可根据自己的工作流，定制开发。无需繁琐的前端开发，只需要简单拽，即可快速搭建企业内部工具。数月的开发工作量，使用卡拉云后可缩减至数天，欢迎试用我们开的卡拉云。



卡拉云可快速接入的常见数据库及 API

卡拉云可根据公司 workflow 需求，轻松搭建数据看板，并且可分享给组内的小伙伴共享数据



仅需拖拽一键生成前端代码，简单一行代码即可映射数据到指定组件中。

卡拉云 用户留存对比 已保存 保存 预览

新用户当日完成新手任务后，次日及七日留存比

name	age	phone	gender	user_wechat_name	stats
宁静	26	13135801177	女	苏打绿	1
金牛	28	15001131177	男	一片云	2
陈工	25	13235643637	男	可爱的河马	2
张三丰	19	17263746473	女	小火花	1
廖小nga	23	13265883839	女	小章鱼	3

5 结果 每页显示 10 < 1 / 1 >

发帖留存 点赞留存 评论留存

选择用户注册日期: 2021-11-19

当日新注册用户: 3678 发帖新用户次日留存: 35 发帖新用户七日留存: 35
当日发帖新用户: 415 未发帖新用户次日留存: 35 未发帖新用户七日留存: 35
当日点赞新用户: 1785

新增事件响应

事件	点击
行动	导出数据
导出文件名	新用户行为留存对比
导出文件类型	JSON格式
导出数据值	CSV格式
	Excel格式
	JSON格式

卡拉云 kalacloud.com

卡拉云可直接添加导出按钮，导出适用于各类分析软件的数据格式，方便快捷。[立即开通卡拉云](#)，导出你的数据

四.总结

在本教程中，我们讲解了如何导入导出数据库至 SQL 文件和 CSV 文件。mysqldump 还有很多使用化，你可以参考 [mysqldump 官方文档](#) 了解更多。更多数据库相关教程可访问 [卡拉云](#) 查看。