

LOAD DATA 语句

聚集函数

AVG()

COUNT() COUNT(DISTINCT)

MAX() MIN()

SUM()

日期和时间格式相关函数

```
mysql -u root --local-infile=1 -p
```

```
create database lab02;  
use lab02
```

## LOAD DATA 语句

<https://dev.mysql.com/doc/refman/8.0/en/load-data.html>

```
LOAD DATA  
  [LOW_PRIORITY | CONCURRENT] [LOCAL]  
  INFILE 'file_name'  
  [REPLACE | IGNORE]  
  INTO TABLE tbl_name  
  [PARTITION (partition_name [, partition_name] ...)]  
  [CHARACTER SET charset_name]  
  [{FIELDS | COLUMNS}  
    [TERMINATED BY 'string']  
    [[OPTIONALLY] ENCLOSED BY 'char']  
    [ESCAPED BY 'char']  
  ]  
  [LINES  
    [STARTING BY 'string']  
    [TERMINATED BY 'string']  
  ]  
  [IGNORE number {LINES | ROWS}]  
  [(col_name_or_user_var  
    [, col_name_or_user_var] ...)]  
  [SET col_name={expr | DEFAULT}  
    [, col_name={expr | DEFAULT}] ...]
```

可以从 txt、csv 等文本文件中导入数据到指定表，默认设置为：

- 换行符 (\n) 分隔每一行
- 制表符 (\t) 分隔每一列

```
create table 学生 (  
    学号 char(20) primary key,  
    姓名 varchar(30),  
    班级 char(20)  
);
```

~/Desktop/stu.txt 文件的内容如下:

```
17373001    abc 172111  
17373002    def 172112  
17373003    ghi 172113
```

由于安全性设置, 导入前先启用本地文件导入:

```
set global local_infile = 1;  
load data local infile '~/Desktop/stu.txt' into table 学生;
```

```
mysql> select * from 学生;  
+-----+-----+-----+  
| 学号      | 姓名    | 班级    |  
+-----+-----+-----+  
| 17373001 | abc     | 172111  |  
| 17373002 | def     | 172112  |  
| 17373003 | ghi     | 172113  |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```

~/Desktop/stu.csv 文件的内容如下:

```
17373004,jkl,172114  
17373005,mno,172115  
17373006,pqr,172111
```

```
load data local infile '~/Desktop/stu.csv' into table 学生 fields terminated by  
',';
```

```
mysql> select * from 学生;
+-----+-----+-----+
| 学号   | 姓名   | 班级   |
+-----+-----+-----+
| 17373001 | abc    | 172111 |
| 17373002 | def    | 172112 |
| 17373003 | ghi    | 172113 |
| 17373004 | jkl    | 172114 |
| 17373005 | mno    | 172115 |
| 17373006 | pqr    | 172111 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

[mysqlimport](#) 命令行工具封装了 LOAD DATA 语句，可以直接通过命令行调用。

## 聚集函数

<https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions-and-modifiers.html>

```
create table 商品 (
    商品编号 char(30) primary key,
    名称 varchar(50),
    类别 varchar(30),
    单位 char(10),
    单价 dec(10, 2)
);

insert into 商品
values ('00001', '农夫山泉', '矿泉水', '元', '2.00');

insert into 商品
values ('00012', '怡宝', '矿泉水', '元', '2.50');

insert into 商品
values ('00123', '数值分析', '书籍', '元', '35.00');

insert into 商品
values ('01234', 'C++ Primer', '书籍', '元', '128.00');

insert into 商品
values ('12345', '树莓派4B', '数码产品', '元', '300.00');

insert into 商品
values ('23456', '1660SUPER', '数码产品', '元', NULL);
```

## AVG()

统计每种类别商品的平均价格：

```
AVG([DISTINCT] expr) [over_clause]
```

```
mysql> select 类别, avg(单价) as 平均价格 from 商品 group by 类别;
```

```
+-----+-----+
| 类别      | 平均价格      |
+-----+-----+
| 矿泉水    | 2.250000      |
| 书籍      | 81.500000     |
| 数码产品  | 300.000000    |
+-----+-----+
3 rows in set (0.00 sec)
```

除非特殊说明，聚集函数会跳过 NULL 值。

## COUNT() COUNT(DISTINCT)

统计每种类别商品的数量：

```
COUNT(expr) [over_clause]
COUNT(DISTINCT expr,[expr...])
```

```
mysql> select 类别, count(*) as 包含NULL的数量, count(单价) as 不包含NULL的数量 from
商品 group by 类别;
```

```
+-----+-----+-----+
| 类别      | 包含NULL的数量 | 不包含NULL的数量 |
+-----+-----+-----+
| 矿泉水    | 2              | 2                |
| 书籍      | 2              | 2                |
| 数码产品  | 2              | 1                |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

COUNT(\*) 会包含 NULL 的值。

## MAX() MIN()

统计每种类别商品的最高价格和最低价格：

```
MAX([DISTINCT] expr) [over_clause]
MIN([DISTINCT] expr) [over_clause]
```

```
mysql> select 类别, max(单价) as 最高价格, min(单价) as 最低价格 from 商品 group by 类别;
```

类别	最高价格	最低价格
矿泉水	2.50	2.00
书籍	128.00	35.00
数码产品	300.00	300.00

3 rows in set (0.00 sec)

## SUM()

统计每种类别商品的单价总和：

```
SUM([DISTINCT] expr) [over_clause]
```

```
mysql> select 类别, sum(单价) as 单价总和 from 商品 group by 类别;
```

类别	单价总和
矿泉水	4.50
书籍	163.00
数码产品	300.00

3 rows in set (0.00 sec)

## 日期和时间格式相关函数

<https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>

日期、时间的占位符见这个 [表格](#)。

把字符串转化为 datetime 类型：

```
STR_TO_DATE(str,format)
```

```
mysql> select str_to_date('2021-3-29', '%Y-%m-%d');
```

str_to_date('2021-3-29', '%Y-%m-%d')
2021-03-29

1 row in set (0.00 sec)

把 datetime 转化为字符串：

DATE\_FORMAT([date](#),format)

```
mysql> select date_format(str_to_date('2021-3-29', '%Y-%m-%d'), '%Y/%m/%d');
+-----+
| date_format(str_to_date('2021-3-29', '%Y-%m-%d'), '%Y/%m/%d') |
+-----+
| 2021/03/29 |
+-----+
1 row in set (0.00 sec)
```