

MySQL语法速查字典(自建)

一、增

1. 创建一个新数据库

```
create database 名字;
```

例:

```
create database employee;
```

2. 创建数据表

```
create table 名字(  
    字段1的名称 字段的数据类型 [指定约束条件],  
    字段2的名称 字段的数据类型,  
    字段3的名称 字段的数据类型  
);
```

2.1 数据类型

2.1.1 数值型:

整数类型: INT;

小数类型: FLOAT, 默认为 (10,2), 总共10位, 小数部分2位

DECIMAL, 特别精确的小数类型;

2.1.2 日期和时间类型:

DATE: YY-MM-DD;

DATETIME: YY-MM-DD HH:MM:SS;

2.1.3 字符串类型:

char(): 固定长度字符串, 少于设定长度的, 自动补空格, 浪费空间;

varchar(): 可变长度字符串;

2.2 指定约束条件

键值	说明
Primary key	主键约束，唯一性，非空（跟在数据类型后）
Constraint 外键名字 foreign key（字段名） references 另外一张表的表名（字段名）	外键约束，本张表的一列，必须来自另外一张表的主键
Not null	非空约束
Unique	唯一性约束，值只能是唯一的
Default 默认值	默认约束，没有赋值时，默认为设定值，语法：字段名 数据类型 default 默认值
auto_increment	自动增长键，跟在数据类型后

注：尽量不要加约束，会降低工作效率

3. 增加新的列

```
Alter table 表名 add 表名 数据类型 ;
```

4. 在表中第一列增加一个字段

```
Alter table 表名 add 新字段名 数据类型 first;
```

在表中某列之后增加一个字段

```
Alter table 表名 add 新字段名 数据类型 after 列名;
```

插入数据

```
insert into 表名 (列1, 列2, ...) values (值1, 值2, ...);
```

注: 如果是字符串需要添加引号

将查询结果插入到另一张表中

```
insert into 新表名 (字段1, 字段2) select 字段1, 字段2 from 旧表; create table *** select * **
```

将外部文件加载到MySQL中

```
load data infile '文件路径'

into table 表名

fields terminated by ',' optionally enclosed by '\ ' # 以, 逗号分隔, \ 闭合

lines terminated by '\r\n' ; # 行区分为 \r\n
```

创建索引

```
create index 索引名 on 表名 (字段名)
```

二、删

删除数据库

```
Drop database 名字;
```

删除表中字段

```
Alter table 表名 drop 字段名;
```

删除表中外键约束

```
Alter table 表名 drop foreign key 外键名称;
```

删除数据

```
delete from 表名 where 条件;
```

删除记录，留下多条日志

```
truncate table 表名;
```

删除索引

```
drop index 索引名 on 表名;
```

三、改

修改数据表名称

```
alter table 表名 rename 新表名;
```

修改表中字段的数据类型（只改数据类型，不改名称）

```
alter table 表名 modify 字段名 新的数据类型;
```

修改表中字段（名字和数据类型都改）

Alter table 表名 change 旧字段名 新字段名 新数据类型;

修改字段的位置

Alter table 表名 modify 字段名 数据类型 first; Alter table 表名 modify 字段名 数据类型 after 列名;

更新表记录

Update 表名 set 字段名2="where 字段名1 ="; 注：一定要有where条件，否则所有信息都被更改。

四、查

查看当前服务器有哪些数据库/数据表

```
show databases; show tables;
```

查看表结构

```
desc 表名;
```

单表查询

查询符合某条件的简单查询

```
select 所需信息 from 表名 where 条件;
```

使用in关键字

查询id为101和102的产品记录

```
select * from 表名 where id in (101,102) ;
```

上例可以使用or运算符

```
select * from 表名 where id =101 or id=102;
```

查询id不等于101也不等于102的记录

```
select * from 表名 where id not in (101,102) ;
```

上例可以使用and运算符

```
select * from 表名 where id!=101 and id!=102;
```

查询某区间内的信息

```
select * from 表名 where 字段名 between x and y; (between包含边界数值)
```

查询某区间外的信息

```
select * from 表名 where 字段名 not between x and y; (not between不包含边界)
```

模糊查询

%: 代表0到n个任意字符; _: 代表单个任意字符

```
select * from 表名 where 字段名 like ' %';
```

查询带有null值的结果

```
select * from 表名 where 字段名 is null; (任何值与null做运算，输出结果都为null，与null的比较要用is)
```

查询多个条件（同时包含and和or）

查询id=101或者102，且price大于5，并且name='Apple'的水果

```
select * from 表名 where (id=101 or id=102) and price>5 and name='apple';
```

```
select * from 表名 where id in (101 , 102) and price>5 and name='apple'; 注：and的优先级高于or。
```

字段去重操作

```
select distinct 字段名 from 表名;
```

统计函数

```
select count(distinct 字段名) from 表名;
```

排序

```
select 字段名 from 表名 order by 字段名; 注：默认为升序，若需要降序，字段名后面加desc。空值最小。
```

先按字段1排序，字段1内再按字段2排序

```
select 字段名 from 表名 order by 字段名1, 字段2;
```

分组查询

找出每个供应商提供的水果数量

```
select 供应商,count(*) from 表名 group by 供应商;
```

连接名称和分组，可显示出每组有哪些水果。

```
select 供应商,count(*) group_concat(水果名称) from 表名 group by 供应商;
```

group by后面加with rollup，可以对分组结果进行求和

对分组统计的结果进行筛选

```
select 供应商,count(*) from 表名 group by 供应商 having 字段<5; python
```

只显示一部分数据

```
select * from 表名 limit(n,m); 从第n+1条记录开始向后显示m条。
```

分页查询

```
select * from 表名 limit 偏移量,页面大小; 偏移量=(n-1)*页面大小
```

case when语句

```
select 字段名,  
case  
    when 条件 1 then 输出结果,  
    when 条件2 then 输出结果,  
    else 输出结果  
end as 别名  
from 表名;
```

查询语句一般格式

```
select 字段名  
from 表名  
where 条件表达式  
group by 字段名  
having 分组筛选条件  
order by 字段  
limit 数量;
```

注：having是在分组查询的前提下。

创建索引

```
create index 索引名 on 表名(字段名);
```

删除索引

```
drop index 索引名 on 表名;
```

多表查询

内连接

```
select 字段名  
From 表名1,表名2  
where 条件 and 1.字段名=2.字段名;
```

```
select 字段
from 表1 inner join 表2
on 1.字段=2.字段
where 条件;
```

外连接

结构同内连接 left join 左边的表全部显示 right join 右边的表全部显示 连接时 小表写在前面，省内存

子查询

对问题进行分层

例：查询 XXXXXXXXX 是 ZZZ 的 YYYYY

```
select YYY from yyy
where XX=(select ZZZ from zzz where 条件);
```

