

## 数据库操作

- use 数据库名;
  - select 属性名,... from 表名或视图名,... [where 条件 1]  
[group by 属性名 1 [having 条件 2]] [order by 属性名 2  
[asc | desc]];

属性名, 使用 \* 通配符表示输出所有属性名

- where 内连接条件

同时查询两个或两个以上的表、视图, 根据相同意义的属性名 (包括父表主键和子表外键), 组合显示查询的属性名。 示例: where 表名 1.id =表名 2.id

- select 属性名,... from 表名 1 left | right join 表名 2 on 表名 1.属性名 1=表名 2.属性名 2;

left, 左连接, 可以查出表名 1 的所有记录, 只查出表名 2 中匹配的记录。 right, 右连接, 可以查出表名 2 的所有记录, 只查出表名 1 中匹配的记录。 on, 接匹配条件。

- 子查询 (select 语句)
  - 比较 (select 语句)

=, !=, >, <, >=, <=, <>, 以子查询结果为参考值

- 关键字 (select 语句)

[not] in, [不]在子查询结果中。 any, 通常和比较运算符一起使用, 以子查询中任一结果为参考, 比较为真则执行外层查询, 否则不执行。 all, 与 any 类似, 但以子查询中所有结果为参考。 [not] exists, 返回子查询的真假值, 返回为真则执行外层查询, 否则不执行。

- select 语句 1 union | union all select 语句 2 ...

union, 合并查询结果, 并合并相同记录。 union all, 简单合并查询结果, 可能存在相同记录。

- 表名 别名, ...

主要用于简写表名, 减小 select 语句长度。 注意: 别名只在本次 select 语句中有效。

- 属性名 [as] 别名

注意： 别名只在本次 select 语句的查询结果中显示。 属性别名只用于显示查询结果，不能用于 select 语句中简写。 属性别名不能与已存在的属性名或属性别名重复。

- order by 属性名 [asc | desc]
- group by 属性名 [having 条件] [with rollup]

注意： 单独使用 group by，查询结果只显示每个分组的一条记录。 多个属性名分组，将按属性名的出现顺序，递归分组。 with rollup，将会在所有分组的最后加上合计，显示所有分组的各个属性名的总和。

- select 属性名 集合函数(属性名 1) from 表名视图名 group by 属性名 3;

集合函数，count()，sum()，avg()，max()，min() 能显示每个分组的 属性 1 的统计结果。

- select 属性名 group\_concat(属性名 1) from 表名视图名 group by 属性名 2;

可以使查询结果显示每个分组的 属性名 1 的所有记录。

- limit 初始位置, 限制数

只显示从 起始位置 开始的 限制数 条记录

- select distinct 属性名

distinct 参数表示要消除特定属性名的重复记录

- where 比较/指定范围/指定集合/匹配字符/是否为空/多个条件/正则表达式

比较，=，<，<=，>，>=，!=，不等于<>，不大于!>，不小于!< 指定范围，between ...and...，not between ...and... 指定集合，in，not in 匹配字符，like，not like 是否为空，is null，is not null 多个条件，and，or

- 属性名 [not] between 取值 1 and 取值 2
- 属性名 is [not] null

注意: is、isnot 不能替换为 =、!=, 否则将无匹配输出

- 条件 1 and/or 条件 2 [... and/or 条件 n]

注意: and 优先级高于 or

- 属性名 [not] like '字符串'

字符串通配符: %, 代表任意长度字符串 \_, 代表单个字符 注意: 字符串不含通配符时, 与 = 效果相同。

- 属性名 regexp '正则表达式匹配'

'^参考串', 以参考串开始的字符串。 '\$参考串', 以参考串结束的字符串。 '.', 代替任意一个字符, 包括回车与换行。 '[字符集合]', 匹配字符集合中的任意一个字符, 可以指定范围 '[a-c0-9]'。 '[^字符集合]', 匹配字符集合以外的任意一个字符。 '串 1 | 串 2 | 串 3', 匹配任意一个字符串。 '\*字符', 匹配指定字符前至少 0 个字符。 '+字符', 匹配指定字符前至少 1 个字符。 '字符串 {N}', 字符串连续出现 N 次。 '字符串 {M, N}', 字符串连续出现至少 M 次, 至多 N 次。

- 属性名 [not] in (元素 1,...)

注意: 字符型元素使用单引号

- create [algorithm={undefined | merge | temptable}] view 视图名 [(属性名, ...)] as select \* | 属性名,... from 表名,... where 过滤条件 [with [cascaded | local] check option];

创建子视图时, 使用 with [cascaded] check option 参数, 要求更新新视图需要考虑父视图的约束条件, 可以保证数据的安全性。

- create or replace 后续同 create 语法

存在即替换, 不存在则创建

- alter 后续同 create 语法
- drop view [if exists] 视图名,... [restrict | cascade];

- `select drop_priv from mysql.user where user='用户名';`

从 MySQL 数据库下的 user 表中查询是否有 drop 权限

- `update 视图名 set 属性名=值,... where 过滤条件;`

不可更新情况: 1 2 3 4 5 6 7

- `desc 视图名;`
- `show create view 视图名;`
- `show table status like '视图名';`
- `select * from information_schema.views;`

所有视图都存储在 information\_schema 数据库下的 views 表中。 \*表示查询所有的列的信息。

- `select select_priv, create_view_priv from 数据库名.表名 where user='用户名';`

查询某个用户在某个数据库的特定表上是否具有 select 权限和 creat view 权限

- `drop table 表名;`

注意: 若为父类表, 需先删除子类表或删除子类表的外键约束, 然后才能删除父类表

- `create table 表名(属性名 类型 [约束条件], ...);`
  - `constraint 外键名 foreign key(属性名 1.1, 1.2, ...) references 表名(属性名 2.1, 2.2, ...)`
  - `not null`
  - `unique`
  - `[unique | fulltext | spatial] index|key [索引名] (属性名 1 [长度] [asc | desc], ...)`

unique, 创建唯一性索引, item 也具唯一性才能起作用。  
fulltext, 创建全文索引, 存储引擎需要是 MyISAM, 否则出错。  
spatial, 创建空间索引, item 必须 not null 并且是 geometry/point/linestring/polygon 类型。不带 unique/fulltext/spatial, 即是普通索引。index/key, 效果一样。多个 item 即是多列索引, 只有第一个 item 被用于查询时, 索引才生效。

- `drop index 索引名 on 表名;`

- `alter table 表名 add [unique | fulltext | spatial] index 索引名 (属性名 [(长度)] [asc | desc], ...);`
- `explain select * from 表名 where 属性名='关键字';`

查看搜索是否使用了索引键

- `create [unique | fulltext | spatial] index 索引名 on 表名 (属性名 [(长度)] [asc | desc], ...);`
- `auto_increment`
- `default` 默认值
- `primary key`
- `alter table 表名` 修改操作;
  - `add 属性名 数据类型 [约束条件] [first | after 属性名 2]`
  - `modify 属性名 新数据类型`
  - `change 属性名 新属性名 新数据类型`
  - `drop foreign key 外键名`
  - `drop 属性名`
  - `engine=innodb | myisam | memory`
  - `modify 属性名 数据类型 first | after item2`
  - `rename [to] 新表名`
- `insert into 表名[(属性名列表)] values(值列表 1)[, (值列表 2), ...];`

不指定属性名，表示插入所有属性名的列的值。指定属性名，代表只插入指定属性名的列的值。值列表可以后延，一次插入多个值列表。注意：值列表的个数、顺序、类型范围必须与表中属性名列表一致。自增属性赋值，跳过该属性，或使用 `null`

- `insert into 表名 1(属性名列表 1) select 属性名列表 2 from 表名 2 where 条件;`

把查询结果当作多个值列表，插入行

- `update 表名 set 属性名 1=值 1, ... where 条件;`

对符合条件的所有，指定的属性名更新为指定值

- `delete from 表名 [where 条件];`

不指定条件，则删除指定表中的所有行。

- describe | desc 表名;
  - show create table 表名;
  - show tables;
- create database 数据库名;
- show databases;