# 操作表结构

## 创建表结构

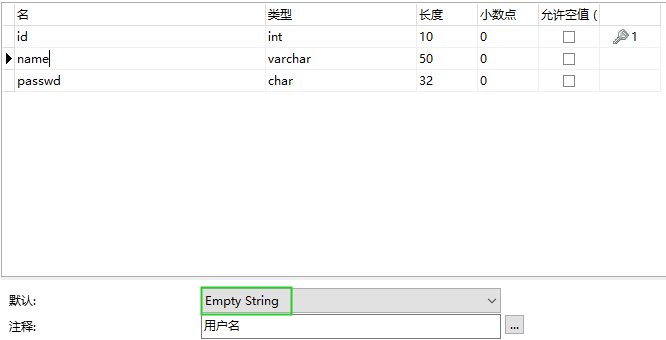
### 常见字段设置

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表结构常见字段 | | | | | | |
| 注释 | 列名 | 类型 | 长度 | 默认值 | 说明 |
| 文章ID | article\_id | mediumint | 8 | '0' | 主键一般设置为自增长 |
| 分类ID | type\_id | smallint | 5 | '0' |  |
| 标题 | title | varchar | 255 | '' |  |
| 描述 | description | varchar | 255 | '' |  |
| 正文 | body | mediumtext |  |  |  |
| 用户IP | user\_ip | char | 15 | '' | IP长度是固定的 |
| 时间戳 | pubdate | int | 10 | '0' |  |
| 是否可见 | is\_show | tinyint | 1 | '0' | “是否”一般长度为1位 |
| MD5密码 | passwd | char | 32 |  | MD5一般长度为32位 |

在设计字段时，所有字段（**text字段除外**）都设置为**NOT NULL**并设置**默认值**。

主键ID、文章ID、分类ID、时间戳、分页大小、长度、高度、宽度、访问次数等基本上都**不可能**为负数，所以可以设置为**无符号**。

比如设置“用户名”字段，如下截图所示：



以后设置其它字段时，可进行参考。

# MySQL管理

# MySQL操作

## 表结构

### 创建表

|  |
| --- |
| CREATE TABLE user\_tbl(name VARCHAR(20), signup\_date DATE); |

### 修改表

#### 修改表名

|  |
| --- |
| ALTER TABLE user\_tbl RENAME TO backup\_tbl; |

#### 修改栏目名

|  |
| --- |
| ALTER TABLE user\_tbl RENAME COLUMN signup\_date TO signup; |

#### 增加栏位

|  |
| --- |
| ALTER TABLE user\_tbl ADD email VARCHAR(40); |

#### 修改栏位属性

|  |
| --- |
| ALTER TABLE user\_tbl ALTER COLUMN signup\_date SET NOT NULL; |

#### 删除栏位

|  |
| --- |
| ALTER TABLE user\_tbl DROP COLUMN email; |

### 删除表

|  |
| --- |
| DROP TABLE IF EXISTS backup\_tbl; |

## 表记录

### 插入记录

|  |
| --- |
| INSERT INTO user\_tbl(name, signup\_date) VALUES('张三', '2013-12-22'); |

### 选择记录

|  |
| --- |
| SELECT \* FROM user\_tbl; |

### 更新记录

|  |
| --- |
| UPDATE user\_tbl set name = '李四' WHERE name = '张三'; |

### 删除记录

|  |
| --- |
| DELETE FROM user\_tbl WHERE name = '李四'; |

# MySQL经验

## 设置最大连接数的方法

方法一：

进入MYSQL安装目录 打开MYSQL配置文件 my.ini 或 my.cnf查找 max\_connections=100 修改为 max\_connections=1000 服务里重起MYSQL即可

方法二：MySQL的最大连接数默认是100

客户端登录：mysql -uusername -ppassword

设置新的最大连接数为200：mysql> set GLOBAL max\_connections=200

显示当前运行的Query：mysql> show processlist

显示当前状态：mysql> show status

退出客户端：mysql> exit

查看当前最大连接数：mysqladmin -uusername -ppassword variables

方法三：

## 表结构增加字段并设置默认值

|  |
| --- |
| ALTER TABLE zlszzc ADD is\_fumian tinyint(1) unsigned not null DEFAULT '0' COMMENT '是否负面: 0 不是负面, 1 是负面';  ALTER TABLE zlszzc ADD is\_hidden tinyint(1) unsigned not null DEFAULT '0' COMMENT '是否隐藏: 0 不隐藏, 1 隐藏'; |

参考网址: <http://blog.csdn.net/kimgoo/article/details/54630257>

## mysql where in 性能优化

<https://blog.csdn.net/pengyufight/article/details/77523404>

<https://blog.csdn.net/hotdust/article/details/51451300>

## MySQL查询包含汉字的记录

|  |
| --- |
| SELECT \* FROM `table\_test` WHERE textname REGEXP '[^ -~]'=1  列出英文字段记录  SELECT \* FROM `table\_test` WHERE textname REGEXP '[^ -~]'!=1  列出中文字段记录  SELECT username FROM cdb\_members WHERE username REGEXP '[/u4e00-/u9fa5]+'!=1 |

## 高效查询随机记录

在PHP等编程语言里面定义好代码，再查询MySQL数据库，示例如下：

|  |
| --- |
| <?php  $num = 20;  $dsql->SetQuery("SELECT MAX(id), MIN(id) FROM `#@\_\_archives`");  $dsql->Execute();  while ($row = $dsql->GetArray()) {  $max\_id = $row['MAX(id)'];  $min\_id = $row['MIN(id)'];  }  $ids = array();  $index = 0;  while (true) {  $rn = rand($min\_id, $max\_id);  if (!in\_array($rn, $ids)) {  $ids[] = $rn;  $index++;  }  if ($index >= $num + 20) {  break;  }  }  shuffle($ids);  $in\_ids = array\_slice($ids, 0, $num);  $in\_ids = implode(',', $in\_ids);  $query = "id IN ($in\_ids)";  ?> |

## Limit查询优化

Limit 用于分页查询时越往后翻性能越差，解决的原则：缩小扫描范围，如下所示：

|  |
| --- |
| select \* from orders where id > (select id from orders order by id desc limit 1000000, 1) order by id desc limit 0,10 |

如果以上方案依然很慢呢？只好用游标了，感兴趣的朋友阅读 JDBC 使用游标实现分页查询的方法。

# MySQL优化

盘点开发中那些常用的MySQL优化

<https://blog.csdn.net/lxw1844912514/article/details/106632325>

# MySQL错误

## Incorrect integer value- - for column -id- at row

解决办法是更改 mysql 中的配置 my.ini，而 Linux 系统则是 my.cfg。

打开my.ini中查找 sql-mode，

sql-mode的默认值为

sql-mode="STRICT\_TRANS\_TABLES,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION"，

将其修改为

sql-mode="NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION"，

重启mysql后即可。

## mysqld\_safe A mysqld process already exists

|  |
| --- |
| [root@localhost mysql]# gedit /etc/my.cnf  [root@localhost mysql]# service mysqld restart  Shutting down MySQL.................................................................................................................................................................................................................^C  [root@localhost mysql]# service mysqld start  Starting MySQL SUCCESS!  [root@localhost mysql]# 2019-04-23T02:10:46.328601Z mysqld\_safe A mysqld process already exists  ^C  [root@localhost mysql]# ps aux |grep mysqld  root 2141 0.0 0.0 115436 0 ? S Feb19 0:00 /bin/sh /zls/server/mysql//bin/mysqld\_safe --datadir=/zls/server/mysql/data/ --pid-file=/zls/server/mysql/data/mysqld.pid  mysql 2420 10.1 6.5 4761528 524820 ? Sl Feb19 9122:34 /zls/server/mysql/bin/mysqld --basedir=/zls/server/mysql/ --datadir=/zls/server/mysql/data --plugin-dir=/zls/server/mysql//lib/plugin --user=mysql --log-error=/zls/server/mysql/log/mysqld.log --pid-file=/zls/server/mysql/data/mysqld.pid --socket=/zls/server/mysql/data/mysql.sock --port=3306  root 14298 0.0 0.0 112708 968 pts/0 S+ 22:13 0:00 grep --color=auto mysqld  [root@localhost mysql]# kill -9 2141  [root@localhost mysql]# kill -9 2420  [root@localhost mysql]# ps aux |grep mysqld  root 14319 0.0 0.0 112704 968 pts/0 S+ 22:14 0:00 grep --color=auto mysqld  [root@localhost mysql]# service mysqld start  Starting MySQL.. SUCCESS!  [root@localhost mysql]# |

参考：

<https://blog.csdn.net/xiaoqiu_cr/article/details/80906955>

<https://blog.csdn.net/zhouyingge1104/article/details/85254895>