

# 狂神说MySQL07：权限及如何设计数据库

秦疆 狂神说 2020-05-04

狂神说MySQL系列连载课程，通俗易懂，基于MySQL5.7.19版本，欢迎各位狂粉转发关注学习。禁止随意转载，转载记住贴出B站视频链接及公众号链接！



上课视频同步文档



演讲·公开课 【狂神说Java】MySQL最新教程通俗易懂

20-01-14 17:47:32

3.4万 1129 136 927 961 824 99

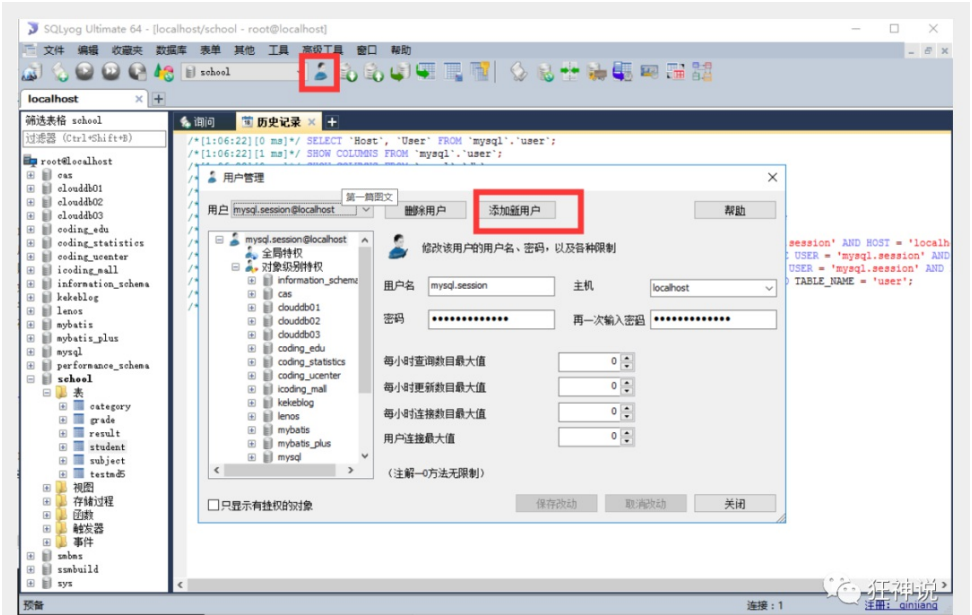
编辑 数据 ...

狂神说

## 权限及设计数据库

### 用户管理

使用SQLyog 创建用户，并授予权限演示



```

/* 用户和权限管理 */ -----
用户信息表: mysql.user

-- 刷新权限
FLUSH PRIVILEGES

-- 增加用户 CREATE USER kuangshen IDENTIFIED BY '123456'
CREATE USER 用户名 IDENTIFIED BY [PASSWORD] 密码 (字符串)
    - 必须拥有mysql数据库的全局CREATE USER权限, 或拥有INSERT权限。
    - 只能创建用户, 不能赋予权限。
    - 用户名, 注意引号: 如 'user_name'@'192.168.1.1'
    - 密码也需引号, 纯数字密码也要加引号
    - 要在纯文本中指定密码, 需忽略PASSWORD关键词。要把密码指定为由PASSWORD() 函数返回的混编值, 需包含关键字PASSWORD

-- 重命名用户 RENAME USER kuangshen TO kuangshen2
RENAME USER old_user TO new_user

-- 设置密码
SET PASSWORD = PASSWORD('密码')    -- 为当前用户设置密码
SET PASSWORD FOR 用户名 = PASSWORD('密码')    -- 为指定用户设置密码

-- 删除用户 DROP USER kuangshen2
DROP USER 用户名

-- 分配权限/添加用户
GRANT 权限列表 ON 表名 TO 用户名 [IDENTIFIED BY [PASSWORD] 'password']
    - all privileges 表示所有权限
    - *.* 表示所有库的所有表
    - 库名.表名 表示某库下面的某表

-- 查看权限 SHOW GRANTS FOR root@localhost;
SHOW GRANTS FOR 用户名

-- 查看当前用户权限
SHOW GRANTS; 或 SHOW GRANTS FOR CURRENT_USER; 或 SHOW GRANTS FOR CURRENT_USER();

-- 撤销权限
REVOKE 权限列表 ON 表名 FROM 用户名
REVOKE ALL PRIVILEGES, GRANT OPTION FROM 用户名    -- 撤销所有权限

```

## 权限解释

```

-- 权限列表
ALL [PRIVILEGES]    -- 设置除GRANT OPTION之外的所有简单权限
ALTER    -- 允许使用ALTER TABLE
ALTER ROUTINE    -- 更改或取消已存储的子程序
CREATE    -- 允许使用CREATE TABLE
CREATE ROUTINE    -- 创建已存储的子程序
CREATE TEMPORARY TABLES    -- 允许使用CREATE TEMPORARY TABLE
CREATE USER    -- 允许使用CREATE USER, DROP USER, RENAME USER和REVOKE ALL PRIVILEGES。

```

```

CREATE VIEW          -- 允许使用CREATE VIEW
DELETE              -- 允许使用DELETE
DROP                -- 允许使用DROP TABLE
EXECUTE             -- 允许用户运行已存储的子程序
FILE               -- 允许使用SELECT...INTO OUTFILE和LOAD DATA INFILE
INDEX              -- 允许使用CREATE INDEX和DROP INDEX
INSERT             -- 允许使用INSERT
LOCK TABLES       -- 允许对您拥有SELECT权限的表使用LOCK TABLES
PROCESS            -- 允许使用SHOW FULL PROCESSLIST
REFERENCES         -- 未被实施
RELOAD             -- 允许使用FLUSH
REPLICATION CLIENT  -- 允许用户询问从属服务器或主服务器的地址
REPLICATION SLAVE   -- 用于复制型从属服务器（从主服务器中读取二进制日志事件）
SELECT            -- 允许使用SELECT
SHOW DATABASES     -- 显示所有数据库
SHOW VIEW          -- 允许使用SHOW CREATE VIEW
SHUTDOWN           -- 允许使用mysqladmin shutdown
SUPER             -- 允许使用CHANGE MASTER, KILL, PURGE MASTER LOGS和SET GLOBAL语句, mysqladmin debug命令; 允许您连接（一次），即使已达到max_connections。
UPDATE            -- 允许使用UPDATE
USAGE              -- “无权限”的同义词
GRANT OPTION        -- 允许授予权限

/* 表维护 */

-- 分析和存储表的关键字分布
ANALYZE [LOCAL | NO_WRITE_TO_BINLOG] TABLE 表名 ...
-- 检查一个或多个表是否有错误
CHECK TABLE tbl_name [, tbl_name] ... [option] ...
option = {QUICK | FAST | MEDIUM | EXTENDED | CHANGED}
-- 整理数据文件的碎片
OPTIMIZE [LOCAL | NO_WRITE_TO_BINLOG] TABLE tbl_name [, tbl_name] ...

```

## MySQL备份

### 数据库备份必要性

- 保证重要数据不丢失
- 数据转移

### MySQL数据库备份方法

- mysqldump备份工具
- 数据库管理工具,如SQLyog
- 直接拷贝数据库文件和相关配置文件

### mysqldump客户端

作用：

- 转储数据库
- 搜集数据库进行备份
- 将数据转移到另一个SQL服务器,不一定是MySQL服务器

```
mysqldump -h 主机名 -u 用户名 -p [options] 数据库名  
[ table1 table2 table3 ] > path/filename.sql
```

## 示例

# 备份myschool数据库如：

```
> mysqldump -u root -p myschool > d:/myschool.sql
```

EnterPassword: \*\*\*\*\*

预存文件目录，须有该  
目录读写权限



### -- 导出

1. 导出一张表 -- `mysqldump -uroot -p123456 school student >D:/a.sql`  
`mysqldump -u用户名 -p密码 库名 表名 > 文件名 (D:/a.sql)`
2. 导出多张表 -- `mysqldump -uroot -p123456 school student result >D:/a.sql`  
`mysqldump -u用户名 -p密码 库名 表1 表2 表3 > 文件名 (D:/a.sql)`
3. 导出所有表 -- `mysqldump -uroot -p123456 school >D:/a.sql`  
`mysqldump -u用户名 -p密码 库名 > 文件名 (D:/a.sql)`
4. 导出一个库 -- `mysqldump -uroot -p123456 -B school >D:/a.sql`  
`mysqldump -u用户名 -p密码 -B 库名 > 文件名 (D:/a.sql)`

可以-w携带备份条件

### -- 导入

1. 在登录mysql的情况下: -- `source D:/a.sql`  
`source 备份文件`
2. 在不登录的情况下  
`mysql -u用户名 -p密码 库名 < 备份文件`

## 规范化数据库设计

### 为什么需要数据库设计

当数据库比较复杂时我们需要设计数据库

糟糕的数据库设计：

- 数据冗余,存储空间浪费

- 数据更新和插入的异常
- 程序性能差

良好的数据库设计：

- 节省数据的存储空间
- 能够保证数据的完整性
- 方便进行数据库应用系统的开发

软件项目开发周期中数据库设计：

- 需求分析阶段: 分析客户的业务和数据处理需求
- 概要设计阶段:设计数据库的E-R模型图，确认需求信息的正确和完整.

设计数据库步骤

- 收集信息
  - 与该系统有关人员进行交流，座谈，充分了解用户需求，理解数据库需要完成的任务.
- 标识实体[Entity]
- - 标识数据库要管理的关键对象或实体,实体一般是名词
- 标识每个实体需要存储的详细信息[Attribute]
- 标识实体之间的关系[Relationship]

## 三大范式

问题：为什么需要数据规范化？

不合规范的表设计会导致的问题：

- 信息重复
- 更新异常
- 插入异常
  - 无法正确表示信息
- 删除异常
  - 丢失有效信息

### 三大范式

#### 第一范式 (1st NF)

第一范式的目标是确保每列的原子性,如果每列都是不可再分的最小数据单元,则满足第一范式

#### 第二范式(2nd NF)

第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。

第二范式要求每个表只描述一件事情

### 第三范式(3rd NF)

如果一个关系满足第二范式,并且除了主键以外的其他列都不传递依赖于主键列,则满足第三范式.

第三范式需要确保数据表中的每一列数据都和主键直接相关，而不能间接相关。

### 规范化和性能的关系

为满足某种商业目标，数据库性能比规范化数据库更重要

在数据规范化的同时，要综合考虑数据库的性能

通过在给定的表中添加额外的字段,以大量减少需要从中搜索信息所需的时间

通过在给定的表中插入计算列,以方便查询



视频同步更新

如果觉得帮助到了您，不妨赞赏支持一下吧！



“赠人玫瑰，手有余香”

狂神说的赞赏码

 狂神说

