

第6章 MySQL



课程名称：Web程序设计



MySQL简介

- ❖ MySQL是MYSQL AB公司开发的一种开放源代码的关系
- ❖ MySQL关系型数据库于1998年1月发行第一个版本
- ❖ 2005年， Oracle公司收购其第三方的存储引擎InnoDB。
- ❖ 2008年1月， MySQL AB公司被Sun公司以收购。
- ❖ 2009年4月， Oracle公司收购Sun公司， 自此MySQL数据库进入Oracle时代。



MySQL特点

- ❖ 体积小
- ❖ 速度快
- ❖ 开放源码
- ❖ 结构化查询语言（Structured Query Language, SQL）
- ❖ 许多中小型网站的选择



表 (Table)

- ❖ 关系型数据库是采用关系模型组织数据的数据库
- ❖ 简单说，关系模型就是二维表格模型，而一个关系型数据库就是由二维表及其之间的联系组成的一个数据组织。

普通词汇	关系模型术语	数据库词汇
二维表	关系 (Relation)	表 (Table)
行 (Row)	元组 (Tuple)	记录 (Record)
列 (Column)	属性 (Attribute)	字段 (Column)
	域 (Field)	数据类型 (Data Type)



SQL

❖ SQL的分类

- 查询（Data Query Language, DQL）
 - SELECT
- 操纵（Data Manipulate Language, DML）
 - INSERT、DELETE、UPDATE
- 定义（Data Definition Language, DDL）
 - CREATE、DROP、ALTER
- 控制（Data Control Language, DCL）
 - GRANT、REVOKE、COMMIT、ROLLBACK



XP. 小皮
CN

首页

网站

数据库

FTP

软件管理

设置

【腾讯云】首单特惠秒杀, 2核2G云服务器45元/年 交流QQ群: 2805531

三 - x

全部 系统环境 安全 网站程序 工具

显示全部

全部 Web Servers 数据库 文件服务 php redis 数据库工具(客户端) composer 数据库工

	Apache2.4.39	web服务, 发布包内置	卸载	设置
	Nginx1.15.11	web服务, 发布包内置	卸载	设置
	Apache2.4.43	web服务, 发布包内置	安装	
	Nginx1.16.1	web服务, 发布包内置	安装	
	MySQL8.0.12	数据库服务	安装	
	MySQL5.7.26	数据库服务	卸载	设置
	MySQL5.0.96	数据库服务	安装	
	MySQL5.1.60	数据库服务	安装	
	MySQL5.5.29	数据库服务	安装	
	FTP0.9.60	文件服务	卸载	设置
	php5.2.17nts	php运行支持程序, 执行php程序需要为Apache指定	安装	
		php运行支持程序, 执行php程序需要为		

▶ Apache2.4.39 ▶ MySQL5.7.26

① 版本: 8.1.1.2



XP. 小皮
CN

首页

网站

数据库

FTP

软件管理

设置

【腾讯云】首单特惠秒杀, 2核2G云服务器45元/年 交流QQ群: 2805531

三 - x

全部 系统环境 安全 网站程序 工具

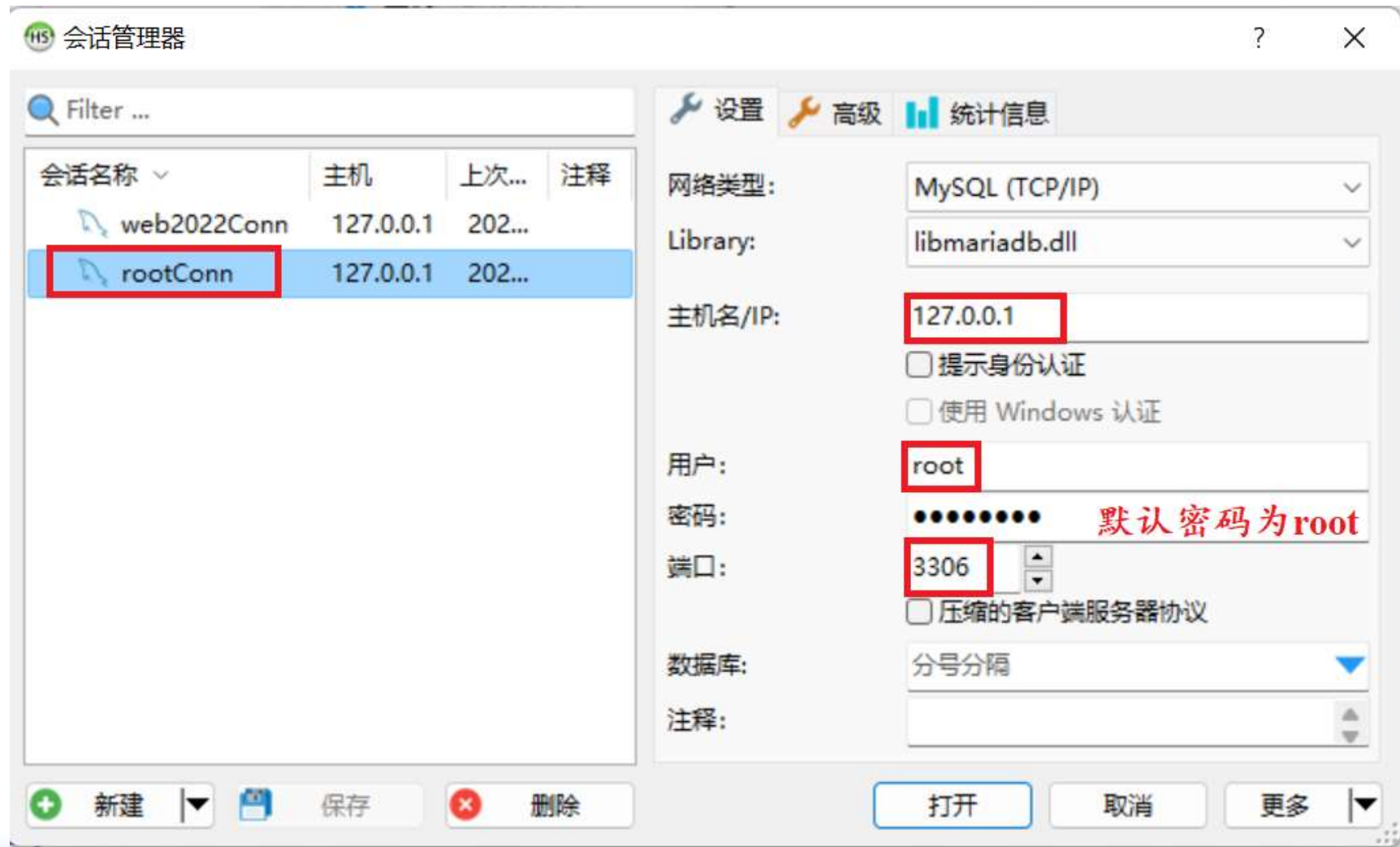
显示全部

全部 Web Servers 数据库 文件服务 php redis 数据库工具(客户端) composer 数据库工

	php7.3.4nts	php运行支持程序, 执行php程序需要为Apache指定	卸载	设置
	php7.4.3nts	php运行支持程序, 执行php程序需要为Apache指定	安装	
	php7.3.9nts	php运行支持程序, 执行php程序需要为Apache指定	安装	
	redis3.0.504	数据库缓存工具	安装	
	redisClient2.0.0	数据库缓存工具(客户端)	安装	
	SQL_Front5.3	轻巧的mysql图形管理工具	安装	
	HeidiSQL11.0	MariaDB和MySQL全球最流行的管理工具	卸载	管理
	DB4S3.12.0	用于创建、设计和编辑与SQLite兼容的数据库文件的高质量、可视化工具。	安装	
	composer1.8.5	Composer是PHP世界里用于管理项目依赖的工具	安装	
	phpMyAdmin4.8.5	mysql管理工具	安装	
	empirecms7.5	帝国cms	一键部署	安装信息

▶ Apache2.4.39 ▶ MySQL5.7.26

① 版本: 8.1.1.2





rootConn\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助

用户管理 - web2022

选择用户账号(S):

+ 添加 克隆 - 删除

用户名 ^	主机
first	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
web2022	%

证书 限制 SSL 选项

用户名(N): web2022

来自主机(H): %

密码(P):

重输密码:

允许访问:

+ 添加

- ☒ 全局权限
 - ☒ EXECUTE
 - ☒ PROCESS
 - ☒ SELECT
 - ☒ SHOW DATABASES
 - ☒ SHOW VIEW
 - ☒ ALTER
 - ☒ ALTER ROUTINE
 - ☒ CREATE
 - ☒ CREATE ROUTINE

仅从服务器位置访问(Y)
本地网络: 192.168.%(1)
本地网络: 10.122.%(0)
☒ 从任意地点访问(Z)

保存 废弃 关闭

% 已连接: 00:02 MySQL 5.7.26 运行时间: 1 天, 00:39 h 服务器时间: 9: 空闲。



rootConn\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助

数据库过滤器 表过滤器 主机: 127.0.0.1 查询*

rootConn

- information
- first
- mysql
- performance
- sys
- web2022

会话管理器

Filter ...

会话名称	主机	上次...	注释
web2022Conn	127.0.0.1	202...	
rootConn	127.0.0.1	202...	

设置 高级 统计信息

网络类型: MySQL (TCP/IP)

Library: libmariadb.dll

主机名/IP: 127.0.0.1

☐ 提示身份认证

☐ 使用 Windows 认证

用户: web2022

密码:

端口: 3306

☐ 压缩的客户端服务器协议

数据库: 分号分隔

注释:

+ 新建 保存 删除

打开 取消 更多

```
322 SHOW GRANTS FOR 'web2022'@'%';
323 SHOW GRANTS FOR 'web2022'@'%';
324 SET PASSWORD FOR 'web2022'@'% ' = 'web2022';
325 FLUSH PRIVILEGES;
326 SHOW GRANTS FOR 'web2022'@'%';
```

已连接: 00:05 MySQL 5.7.26 运行时间: 1 天, 00:42 h 服务器时间: 9: 空闲。



web2022Conn\ - HeidiSQL Portable 11.0.0.5919

英

文件 编辑 搜索 工具 转到 帮助



Donate

数据库过滤器 表过滤器

主机: 127.0.0.1 查询*

- rootConn
 - information_schema
 - first
 - mysql
 - performance_schema
 - sys
 - web2022

```
1 SELECT * FROM user;
```

- Filter ...
- 字段
 - SQL 函数
 - SQL 关键字
 - SQL代码片段
 - 查询历史
 - 查询分析
 - 绑定参数

- web2022Conn
 - info
 - first
 - mys
 - per
 - sys
 - web

- 编辑(K) Alt+Enter
- 删除(L)...
- 清空表(M)... Shift+Del
- 运行过程(N)...
- 创建新的(O) →
- 清除数据选项卡过滤器(P)
- 导出数据库为 SQL 脚本(S)
- 维护(Q)
- 在服务器上查找文本(R) Shift+Ctrl+F
- 批量表编辑器(T)
- 全部展开(U)
- 全部折叠(V)
- 对象树类型选项(W)
- 打印(X)... Ctrl+P
- 刷新(Y) F5
- 断开连接(Z)

- 数据库(T)
- 表(U)
- 表复制(V)
- 视图(W)
- 存储过程(X)
- 触发器(Y)
- 事件(Z)

```
372 /* 进入会话 "rootConn" */
373 /* 进入会话 "rootConn" */
374 /* 到 */
375 /* 进入会话 "rootConn" */
376 /* 进入会话 "web2022Conn" */
```

创建一个新的空白的数据库

已连接: 00:02 MySQL 5.7.26

运行时间: 1 天, 00:45 h

服务器时间: 9: 空闲。



数据库相关SQL命令

- ❖ 创建数据库

CREATE DATABASE [IF NOT EXISTS] 数据库名;

- ❖ 使用数据库

USE 数据库名;

- ❖ 删除数据库

DROP DATABASE [IF EXISTS] 数据库名;

- ❖ 查看数据库信息

SHOW DATABASES;



web2022Conn\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助

主机: 127.0.0.1 查询*

数据库过滤器 表过滤器

rootConn

- information_schema
- first
- mysql
- performance_schema
- sys
- web2022

web2022Conn

- information_schema 160.0 KiB
- first
- mysql
- performance_schema
- sys
- web2022 9.2 KiB

1 SELECT * FROM user;

Filter ...

- 字段
- SQL 函数
- SQL 关键字
- SQL 代码片段
- 查询历史
- 查询分析
- 绑定参数

创建数据库...

名称(N): web2022

字符校对(O): utf8_unicode_ci

服务器默认: utf8_unicode_ci

确定 取消

CREATE 代码:

```
CREATE DATABASE `web2022` /*!40100 COLLATE
```

过滤: 正则表达式

```
373 /* 进入会话 "webApp" */
374 /* 到 127.0.0.1 的连接于 2022-05-27 09:52:39 关闭 */
375 /* 进入会话 "rootConn" */
376 /* 进入会话 "web2022Conn" */
377 SHOW COLLATION;
```

已连接: 00:03 MySQL 5.7.26 运行时间: 1 天, 00:46 h 服务器时间: 9: 空闲。



概念模型

- ❖ 基于客户的想法和观点所形成的认识和抽象
 - 实体（Entity）：客观存在的、可以被描述的事务
 - 如，用户、学生、教师、分组、成绩、课节等。
 - 属性（Attribute）：用于描述实体所具有的特性。
 - 如，账号、密码、学号、姓名、性别、课节号等。
 - 关系（Relationship）：实体之间的联系
 - 如，分组和学生之间的一对多的关系，系统注册账号与学生之间的一对一的关系。



数据模型

- ❖ 关系模型：实体、属性和关系在数据库中的具体体现。
 - 关系数据库：存储各种关系型数据，二维表的集合。
 - 表：实体的映射
 - 行：也叫记录，一个具体的实体数据。
 - 如，一名学生、一个账号。
 - 列：也叫字段，实体的属性。
 - 如，学生的姓名、学生的性别。



约束

❖ 非空（NOT NULL）

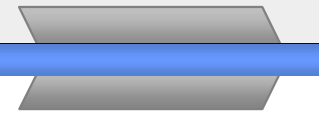
- 要求表中所有记录在该字段上都必须有数据

❖ 唯一键（UNIQUE）

- 要求表中任意两条记录在该字段上的值都不相同，但可以为空。

❖ 主键（PRIMARY KEY）

- 能够唯一地标识表中一行数据的字段或者最小字段组合
- 非空的唯一键



❖ 外键（FOREIGN KEY）

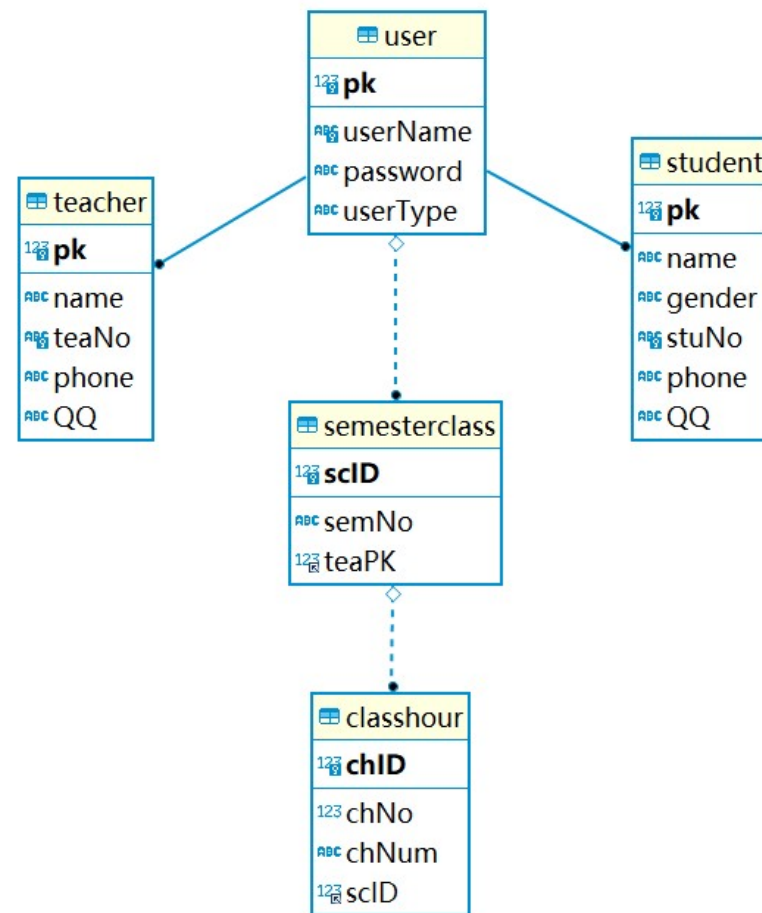
- 同时作用在两个表（父表和子表）上
 - 父键：父表上的主键或唯一键
 - 子键：子表外键的值必须是父表字段值或者空

❖ 检查（CHECK）

- 检查一行记录是否满足条件
- 在一条记录的字段上，不能在多条记录上进行。
 - 正确：工资 + 奖金 > 5000
 - 错误：甲的工资 + 乙的工资 > 2000



ER图



状态status
创建时间ctime
操作日志log

.....



web2022Conn\web2022\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助

英



— □ ×

Donate

数据库过滤器 表过滤器

主机: 127.0.0.1 数据库: web2022 查询*

- rootConn
 - information_schema
 - first
 - mysql
 - performance_schema
 - sys
 - web2022
- web2022Conn
 - information_schema 160.0 KiB
 - first
 - mysql
 - performance_s
 - sys
 - web2022

1 SELECT * FROM user;

Filter ...

- 编辑(K) Alt+Enter
- 删除(L)...
- 清空表(M)... Shift+Del
- 运行过程(N)...
- 创建新的(O)
 - 数据库(T)
 - 表(U)
 - 表复制(V)
 - 视图(W)
 - 存储过程(X)
 - 触发器(Y)
 - 事件(Z)
- 清除数据选项卡过滤器(P)
- 导出数据库为 SQL 脚本(S)
- 维护(Q)
- 在服务器上查找文本(R) Shift+Ctrl+F
- 批量表编辑器(T)
- 全部展开(U)
- 全部折叠(V)
- 对象树类型选项(W)
- 打印(X)... Ctrl+P
- 刷新(Y) F5
- 断开连接(Z)

374 /* 到 127.0.0.1
375 /* 进入会话 "rd
在选择的数据库中创建新

已连接: 00:11 h MySQL 5.7.26 运行时间: 1 天, 00:54 h 服务器时间: 10:02 空闲。



web2022Conn\web2022\user\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助

数据库过滤器 表过滤器

主机: 127.0.0.1 数据库: web2022 表: user 数据 查询*

Donate

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

名称: user

注释: 为能够登录系统的内每位用户保存一条账号记录

字段: + 添加 - 删除 ▲ 向上 ▼ 向下

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释	校对规则	表达式
1	pk	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT			
2	userName	VARCHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值		utf8_unicode_ci	
3	password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值		utf8_unicode_ci	
4	userType	CHAR	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'S'	S表示学生T表示教师	utf8_unicode_ci	

帮助 废弃 保存

过滤: 正则表达式

366 SELECT * FROM `web2022`.`user` LIMIT 100;

已连接: 03:18 h MySQL 5.7.26 运行时间: 04:16 h 服务器时间: 10:59 空闲。



web2022Conn\web2022\student\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助



数据库过滤器 表过滤器

- web2022Conn
 - information_schema 160.0 KiB
 - first
 - mysql
 - performance_schema
 - sys
 - web2022 160.0 KiB
 - classhour 32.0 KiB
 - semesterclass 32.0 KiB
 - student 32.0 KiB
 - teacher 32.0 KiB
 - user 32.0 KiB
 - webapp

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

名称: student

注释: 保存使用系统的学生用户的特有信息

字段: + 添加 - 删除 ▲ 向上 ▼ 向下

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释	校对规则	表达式
1	pk	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	与users表的主键一致		
2	name	VARCHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值		utf8_unicode_ci	
3	gender	CHAR	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	F表示女M表示男	utf8_unicode_ci	
4	stuNo	CHAR	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	学号	utf8_unicode_ci	
5	phone	CHAR	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	电话号码	utf8_unicode_ci	
6	QQ	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	QQ号码	utf8_unicode_ci	

帮助

废弃

保存

过滤: 正则表达式

366 SELECT * FROM `web2022`.`user` LIMIT 1000;

已连接: 03:19 h

MySQL 5.7.26

运行时间: 04:16 h

服务器时间: 11:00

空闲。





主机: 127.0.0.1 数据库: web2022 表: user 数据 查询*

web2022.user: 8 总记录数 (大约)

pk	userName	password	userType
7	601232	\$2y\$10\$MPH40LR4I5FgRVhSZTA82egisUD8Vgs....	T
17	user_1	\$2y\$10\$TqEKbh99eWfYqBbgGkeeT.q2pwsaCU...	S
18	user_2	\$2y\$10\$ZsK1eQjZVBWNGh.Mxhxkd.hVWIZxV.T...	S
22	user_3	\$2y\$10\$CuPbO6HuoBewwBHfyNNhU.tcYJ5zYP8...	S
24	user_4	\$2y\$10\$OJB0/wts.l3taldaXS0g2OCnvB7EKu4rk0...	S
27	user_5	\$2y\$10\$.H/bkW5Z/bAznaJVwMp0mu8Y4tAb7Lu...	S
28	user_6	\$2y\$10\$tOTP8pJTUBD7x/IUIUUFpu23rGOViqqz...	S
29	user_7	\$2y\$10\$dktgShYoHnjtggJ49IDfUuXTwMzKS8YS...	S

主机: 127.0.0.1 数据库: web2022 表: student 数据 查询*

web2022.student: 7 总记录数 (大约) >> 下一个 < 显示所有 < 排序 < 字段

pk	name	gender	stuNo	phone	QQ
17	姓名_1	0	21200101	1300000001	QQ1
18	姓名_2	1	21200102	1300000002	QQ2
22	姓名_3	1	21200103	1300000003	QQ3
24	姓名_4	1	21200104	1300000004	QQ4
27	姓名_5	1	21200105	1300000005	QQ5
28	姓名_6	1	21200106	1300000006	QQ6
29	姓名_7	0	21200107	1300000007	QQ7



数据类型

❖ 数值

- INT、DECIMAL、TINYINT、DOUBLE等

❖ 字符串

- 文本字符串：CHAR、VARCHAR
- 可变类型：TEXT、BLOB
- 特殊类型：SET、ENUM

❖ 日期和时间

- DATE: YYYY-MM-DD格式
- TIME: HH:MM:SS格式
- DATETIME: YYYY-MM-DD HH:MM:SS格式
- TIMESTAMP: 1970-01-01 00:00:00基准
- YEAR: 1901-2155之间两位或四位数字



创建表

CREATE TABLE [IF NOT EXISTS] 表名(
字段1 属性, 字段2 属性,);

属性可包含:

- col_type 数据类型
- NOT NULL| NULL 是否允许控制
- DEFAULT default_value 缺省值
- AUTO_INCREMENT 自动编号, 表中只有1个, 且被索引
- PRIMARY KEY 主键
- reference_definition 注释



```
CREATE TABLE `teacher`(  
  `pk` INT(11) NOT NULL COMMENT '与users表的主键一致',  
  `name` VARCHAR(32) NOT NULL,  
  `teaNo` CHAR(6) NOT NULL COMMENT '工作证号',  
  `phone` CHAR(13) NOT NULL COMMENT '电话号',  
  `QQ` VARCHAR(20) NOT NULL COMMENT 'QQ号',  
  PRIMARY KEY (`pk`) USING BTREE,  
  UNIQUE INDEX `teaNo_UNIQUE` (`teaNo`) USING BTREE,  
  CONSTRAINT `FK_teacher_user` FOREIGN KEY (`pk`)  
    REFERENCES `web2022`.`user` (`pk`) ON UPDATE CASCADE  
);
```



表

❖ 删除表

- DROP TABLE [IF EXISTS] 表名;

❖ 修改表定义

- ALTER TABLE 表名 修改内容;

❖ 重命名表

- RENAME TABLE 旧表名 TO 新表名;

❖ 查看表结构

- DESCRIBE 表名;

❖ 复制表

- CREATE TABLE [IF NOT EXISTS] 表名 LIKE 源表名;
- CREATE TABLE [IF NOT EXISTS] 表名
AS SELECT * FROM 源表名;



web2022Conn\web2022\teacher\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助



数据库过滤器 表过滤器

- web2022Conn
 - information_schema 160.0 KiB
 - first
 - mysql
 - performance_schema
 - sys
 - web2022 160.0 KiB
 - classhour 32.0 KiB
 - semesterclass 32.0 KiB
 - student 32.0 KiB
 - teacher 32.0 KiB**
 - user 32.0 KiB
 - webapp

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

名称: teacher

注释: 保存使用系统的教师用户的特有信息

字段: + 添加 - 删除 ▲ 向上 ▼ 向下

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释	校对规则	表达式
1	pk	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	与users表的主键一致		
2	name	VARCHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值		utf8_unicode_ci	
3	teaNo	CHAR	6	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	工作证号	utf8_unicode_ci	
4	phone	CHAR	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	电话号	utf8_unicode_ci	
5	QQ	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	QQ号	utf8_unicode_ci	

帮助 废弃 保存

过滤: 正则表达式

366 SELECT * FROM `web2022`.`user` LIMIT 1000;

已连接: 03:19 h

MySQL 5.7.26

运行时间: 04:17 h

服务器时间: 11:01

空闲。

中



web2022Conn\web2022\semesterclass\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助

数据库过滤器 表过滤器 主机: 127.0.0.1 数据库: web2022 表: semesterclass 数据 查询*

Donate

- web2022Conn
 - information_schema 160.0 KiB
 - first
 - mysql
 - performance_schema
 - sys
 - web2022 160.0 KiB
 - classhour 32.0 KiB
 - semesterclass 32.0 KiB**
 - student 32.0 KiB
 - teacher 32.0 KiB
 - user 32.0 KiB
 - webapp

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

名称: semesterclass

注释: 每个semNo学期可同时开设多个教学班。一个教学班有一名负责的老师teaPK(user.pk)

字段: + 添加 - 删除 ▲ 向上 ▼ 向下

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释	校对规则	表达式
1	scID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT			
2	semNo	CHAR	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	例如"2021-2022-2"	utf8_unicode_ci	
3	teaPK	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL			

帮助

废弃

保存

过滤: 正则表达式

366 SELECT * FROM `web2022`.`user` LIMIT 1000;

已连接: 03:20 h

MySQL 5.7.26

运行时间: 04:18 h

服务器时间: 11:02 空闲。

英



web2022Conn\web2022\classhour\ - HeidiSQL Portable 11.0.0.5919

文件 编辑 搜索 工具 转到 帮助

数据库过滤器 表过滤器

主机: 127.0.0.1 数据库: web2022 表: classhour 数据 查询*

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

名称: classhour
注释: 每次授课的信息

字段: + 添加 - 删除 ▲ 向上 ▼ 向下

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释	校对规则	表达式	虚拟
1	chID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_I...				
2	chNo	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	课序号, 如1表示...			
3	chNum	CHAR	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	课时的学时信息, 如...	utf8_unicode_ci		
4	sclID	INT	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	所属教学班			

帮助 废弃 保存

X 过滤: 正则表达式

```
366 SELECT * FROM `web2022`.`user` LIMIT 1000;
```

已连接: 03:17 h MySQL 5.7.26 运行时间: 04:14 h 服务器时间: 10:58 空闲。



查看表

❖ SELECT 字段列表|* FROM 表名
[WHERE 子句]
[GROUP BY 子句]
[HAVING 子句]]
[ORDER BY 子句];



操作表

❖ 插入数据

- INSERT [INTO] 表名 [(字段列表)] VALUES (值列表), (值列表), ...;
- INSERT [INTO] 表名 [(字段列表)] SELECT语句;

❖ 删除数据

- DELETE FROM 表名 [WHERE 子句];

❖ 修改数据

- UPDATE 表名 SET 字段名=值, ... [WHERE 子句];

❖ 清空表

- TRUNCATE [TABLE] 表名;



PHP操作MySQL数据库

- ❖ PHP操作MySQL数据库极为简单和高效，使得PHP+MySQL成为当今最为流行的Web开发语言与数据库搭配之一。
- ❖ PHP可以连接MySQL数据库服务器并对数据进行操作。有三种主要的API可供选择：
 - PHP的MySQL扩展
 - PHP的MySQLi扩展
 - PHP数据对象（PHP Database Object, PDO）



PHP的MySQL扩展

- ❖ PHP应用与MySQL数据库交互的早期扩展
 - 面向过程的接口
 - 针对MySQL4.1.3或更早版本
 - 虽然可以与MySQL4.1.3或更新的数据库服务端进行交互，但无法支持后期MySQL服务端提供的一些特性。
 - MySQL4.1.3或更新的服务端版本，建议使用mysqli 扩展。



PHP的MySQLi扩展

❖ MySQL增强扩展

- 包含在PHP 5及以后版本中
- 可使用 MySQL4.1.3或更新版本中新的高级特性
- 相对于MySQL扩展的主要优势有
 - 面向对象接口（也支持面向过程）
 - prepared语句支持
 - 对语句执行支持
 - 事务支持
 - 增强的调试能力
 - 嵌入式服务支持



PHP Database Object, PDO

❖ PHP应用中的一个数据库抽象层规范

■ 一个统一的API接口

- 主要优点“可移植性”：PHP应用无需关心数据库服务器系统类型。即使用PDO的API可以在需要时无缝切换数据库服务器，比如从Firebird 到MySQL，仅仅需要修改很少的PHP代码。
- 缺点：无法使用后期MySQL服务端提供的数据库高级特性。
- PDO的API使用PDO的MySQL驱动（众多PDO驱动中的一个）完成与MySQL服务器端的交互。



比较PHP中三种主要的MySQL连接方式

	PHP的mysqli扩展	PDO (使用PDO MySQL驱动和MySQL Native 驱动)	PHP的mysql扩展
引入的PHP版本	5.0	5.0	3.0之前
PHP5.x是否包含	是	是	是
MySQL开发状态	活跃	在PHP5.3中活跃	仅维护
在MySQL新项目中的建议使用程度	建议 - 首选	建议	不建议
API的字符集支持	是	是	否
服务端prepare语句的支持情况	是	是	否
客户端prepare语句的支持情况	否	是	否
存储过程支持情况	是	是	否
多语句执行支持情况	是	大多数	否
是否支持所有MySQL4.1以上功能	是	大多数	否



php.ini中MySQLi的配置参数

- ❖ `mysqli.max_persistent = -1`
 - 每个进程中允许的最大持久连接数。-1 代表无限制。
- ❖ `mysqli.allow_persistent = On`
 - 是否允许持久连接
- ❖ `mysqli.max_links = -1`
 - 每个进程中允许的最大连接数(持久和非持久)。-1 代表无限制。
- ❖ `mysqli.default_port = 3306`
 - `mysqli_connect()`连接到MySQL数据库时使用的默认TCP端口
- ❖ `mysqli.reconnect = Off`
 - 是否允许重新连接



MySQLi扩展功能概述

- ❖ mysqli类：PHP和MySQL数据库之间的连接
- ❖ mysqli_stmt类：预编译的SQL语句
- ❖ mysqli_result类：从数据库查询中获取的结果集

- ❖ mysqli_driver类：MySQLi驱动
- ❖ mysqli_warning类：MySQLi警告
- ❖ mysqli_sql_exception类：MySQLi异常



mysqli类

- ❖ mysqli类：PHP和MySQL数据库之间的连接
 - connect_errno：最后一次连接的错误状态码
 - 0表示连接成功，非0表示连接失败。
 - connect_error：最后一次连接错误的原因
 - affected_rows：前一条语句影响的记录数
 - set_charset(charset)：设置字符集charset，如'utf8'。
 - query(sql)：执行sql语句
 - DQL返回mysqli_result对象，DML返回布尔值。
 - prepare(sql)：准备执行sql语句
 - 返回mysqli_stmt类型的预处理对象。sql语句中的参数通过占位符?表示。
 - close()：关闭已打开的数据库连接



mysqli_stmt类

- ❖ mysqli_stmt类：预编译的SQL语句
 - insert_id：前一条INSERT语句的主键
 - bind_param(type, var_1[, var_n])：参数绑定
 - type字符串的每一位分别表示参数var_1至var_n的类型，i表示int类型、d表示double类型、s表示字符串类型。
 - execute()：执行SQL语句，返回布尔值。
 - bind_result(var_1[, var_n])：结果集中的字段分别绑定到变量var_1到var_n
 - fetch()：从结果集中取一条数据到绑定变量中
 - get_result()：返回mysqli_result类型对象
 - affected_rows()：返回DML语句影响的记录数
 - free_result()：释放结果集
 - close()：释放语句句柄和结果集



mysqli_result类

- ❖ mysqli_result类：从数据库查询中获取的结果集
 - field_count：结果集的列（字段）数
 - num_rows：结果集的行（记录）数
 - fetch_assoc()：返回结果集中一条数据，形式为关联数组，键是字段名。（DQL使用较easy）
 - fetch_row()：返回结果集中一条数据，形式为索引数组。
 - fetch_array()：返回结果集中一条数据，形式为关联数组和索引数组组合的数组。
 - fetch_object()：返回结果集中一条数据，形式为对象，对象的属性是字段名。
 - fetch_all(mode)：返回结果集，mode是数组形式（MYSQLI_ASSOC、MYSQLI_NUM或MYSQLI_BOTH）。
 - free()：释放结果集



基本操作流程

(1) 建立与MySQL数据库服务器的连接

- `$conn = new mysqli($host, $username, $password, $dbname, $port);`

(2) 设置字符集

- `$conn->set_charset('utf8');`

(3) 编写并执行SQL语句（DML或者DQL）

- `$result = $conn->query($sql);`

(4) 循环地取回SQL语句执行结果进行处理

- `$result ->fetch_assoc();`

(5) 释放结果集

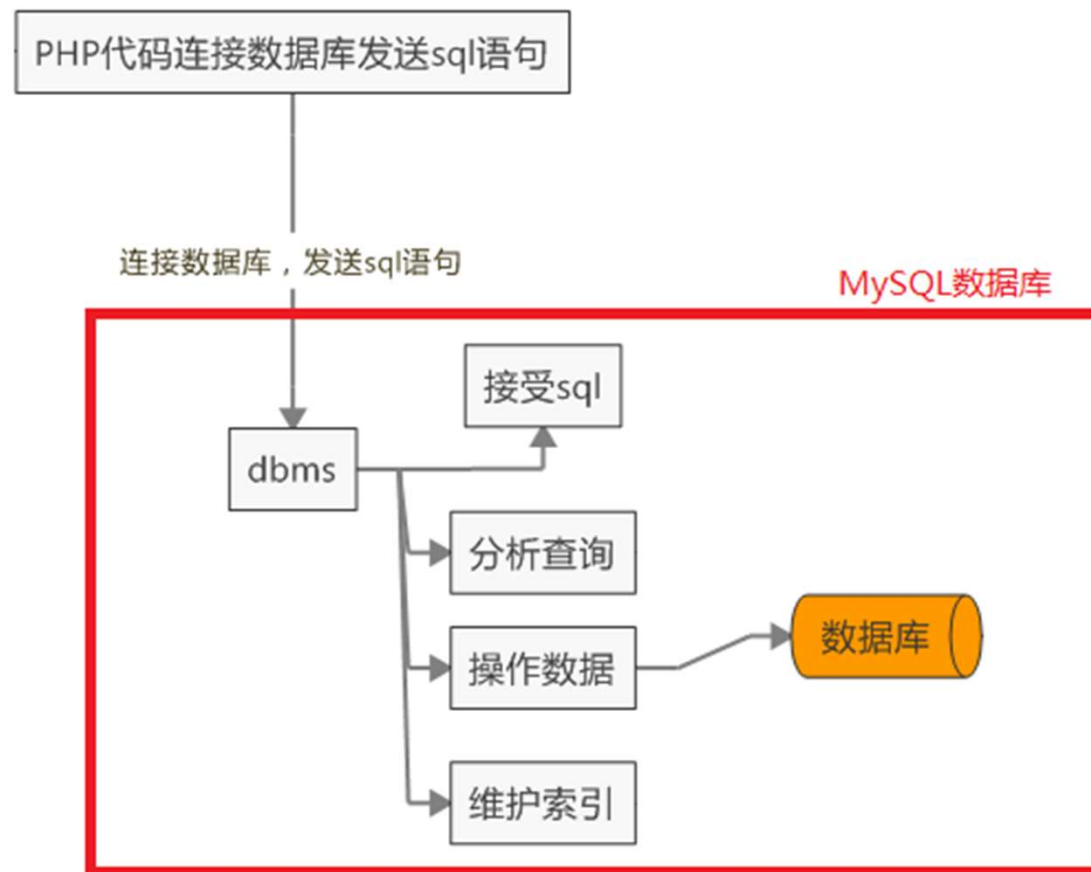
- `$result->free();`

(6) 关闭连接

- `$conn->close();`



MySQLi预处理





使用**mysqli_stmt**类进行语句预处理

(1) MySQL成功建立连接后得到\$conn对象

(2) 获得预处理语句对象\$stmt

```
$stmt = $conn->prepare($sql);
```

(3) 绑定参数

```
$stmt->bind_param($type, $var);
```

(4) 执行语句

```
$stmt->execute();
```

(5) 获取结果记录集\$result

```
$result = $stmt->get_result();
```

(6) 循环地取回SQL语句执行结果进行处理

(7) 释放语句对象

```
$stmt->close();
```



事务处理

❖ 事务（Transaction）

- 是包括多个数据库操作，构成一个完整业务操作的逻辑工作单元。
- 是数据库区别于文件系统的主要特性之一

❖ 事务支持是DBMS对程序员的承诺

- 目的：把数据库从一种一致状态转变为另一种一致状态
- 执行结果：全部成功 or 全部失败



事务的ACID特性

❖ 原子性（Atomicity）

- 事务的一组操作是不可分的
- 要么完全成功，要么完全失败（即便系统崩溃）。

❖ 一致性（Consistency）

- 事务将数据库从一种一致状态转变为下一种一致状态
- 保证事务符合预先设定的业务规则

❖ 隔离性（Isolation）

- 一个事务对数据的影响，在该事务提交前对其他事务都不可见。
- 两个事务并行执行的结果和（某一种）串行执行的结果是一样的，也称可串行性（Serializability）。

❖ 持久性（Durability）

- 事务一旦提交后，其对数据的影响是永久性的（即便系统崩溃）。



使用mysql类进行事务处理

❖ 开启事务

- `query('START TRANSACTION')`
- `query('SET AUTOCOMMIT = FALSE')`
- `begin_transaction(flag)`

❖ 结束事务

- `savepoint(sp)`: 在当前事务中增加名为sp的保存点
- `commit()`: 提交事务
- `rollback()`: 回滚事务



注册用户举例

- ❖ 注册页面（registerStu.html）：用户填写表单，提交至 registerStu.php。
- ❖ 操作数据库（registerStu.php）：使用manageDB.php，向MySQL的user表和student表INSERT两条关联的记录。
 - 成功：跳转至欢迎页面（welcomeStu.php）
 - 失败：跳转回注册页面（registerStu.html）
- ❖ manageDB.php：包含对数据库操作的函数
- ❖ dbInfo.php：包含数据库连接信息



← → ↻ 🏠 ⓘ localhost:3000/6/registerStu.html

用户名:

密码:

姓名:

性别: ☐ 男 ☐ 女

学号:

电话号码:

QQ号码:

← → ↻ 🏠 ⓘ localhost:3000/6/welcomeStu.php

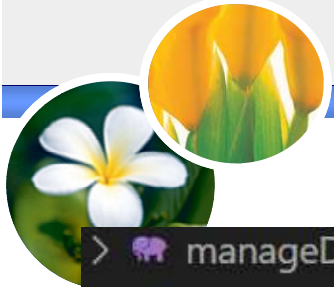
欢迎, user_2023!
[查看所有学生用户信息](#)
[注销用户](#)



```
> <> registerStu.html > ...
1  <!--registerStu.html-->
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5  |   <meta charset="UTF-8">
6  |   <title>学生注册页面</title>
7  </head>
8  <body>
9  |   <form action="registerStu.php" method="post">
10 |       用户名:   <input name="userName" type="text" value="user_"><br>
11 |       密码:     <input name="pwd" type="password" value="pass_"><br>
12 |       姓名:     <input name="name" type="text" value="姓名_"><br>
13 |       性别:     <input name="gender" type="radio" value="0">男
14 |               <input name="gender" type="radio" value="1">女<br>
15 |       学号:     <input name="stuNo" type="text" maxlength="8" value="2120010"><br>
16 |       电话号码: <input name="phone" type="text" maxlength="13" value="1300000000"><br>
17 |       QQ号码:   <input name="QQ" type="text" maxlength="20" value="QQ"><br>
18 |               <input type="submit" value="注册"><br>
19 |   </form>
20 </body>
21 </html>
```




```
> 🐘 registerStu.php
1  <?php    //registerStu.php
2      require_once("manageDB.php");
3
4      if(isset($_POST['userName']) && isset($_POST['pwd']) && isset($_POST['name']) && isset($_POST['gender'])
5          && isset($_POST['stuNo']) && isset($_POST['phone']) && isset($_POST['QQ'])) {
6          if (regStuUser($_POST['userName'], $_POST['pwd'], $_POST['name'], $_POST['gender'], $_POST['stuNo'],
7              $_POST['phone'], $_POST['QQ'])) {
8              header("location:welcomeStu.php");
9              setcookie('userName', $_POST['userName'], time() + 14400);
10         }else{
11             header("location:registerStu.html");
12         }
13     }
14     ?>
```



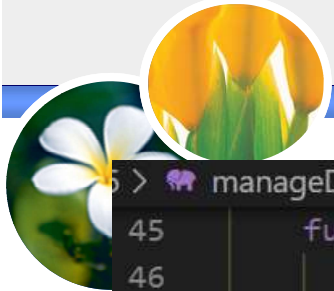
```
> manageDB.php > ...
1  <?php    //manageDB.php
2      require_once('dbInfo.php');
   2 references
3  > function regStuUser($user, $pwd, $stuName, $stuGender, $stuNo, $phone, $QQ){ ...
31      }
32
   2 references
33 > function selStuUser(){ ...
43      }
44
   2 references
45 function delUser($user, $pwd)
46 > { ...
74      }
75  ?>
```

```
> dbInfo.php > ...
1  <?php    //dbInfo.php
2      define('HOST','localhost');
3      define('USERNAME','web2022');
4      define('PASSWORD','web2022');
5      define('DBNAME','web2022');
6      define('PORT', 3306);
7      define('CHARSET', 'utf8');
8      ?>
```



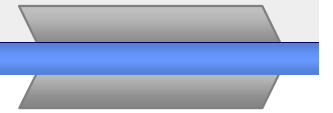
> manageDB.php > regStuUser

```
3 function regStuUser($user, $pwd, $stuName, $stuGender, $stuNo, $phone, $QQ){
4     mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
5     $conn = new mysqli(HOST, USERNAME, PASSWORD, DBNAME, PORT);
6     if($conn->connect_error)
7         die("发生致命错误!");
8     $conn->begin_transaction();
9     $pwdHash = password_hash($pwd, PASSWORD_DEFAULT);
10    $stmt_user = $conn->prepare("INSERT INTO user(userName, password, userType) VALUES( ?, ?, 'S')");
11    $stmt_stu = $conn->prepare("INSERT INTO student( pk, name, gender, stuNo, phone, QQ)
12        VALUES( ?, ?, ?, ?, ?, ?)");
13    try{
14        $stmt_user->bind_param('ss', $user, $pwdHash);
15        $stmt_user->execute();
16        $userPk = $stmt_user->insert_id;
17        $stmt_stu->bind_param('isssss', $userPk, $stuName, $stuGender, $stuNo, $phone, $QQ);
18        $stmt_stu->execute();
19        $stmt_user->close();
20        $stmt_stu->close();
21        $conn->commit();
22        $conn->close();
23        return true;
24    }catch (mysqli_sql_exception $exception){
25        $stmt_user->close();
26        $stmt_stu->close();
27        $conn->rollback();
28        $conn->close();
29        return false;
30    }
31 }
```



5 > manageDB.php > ...

```
45 function delUser($user, $pwd){
46     mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
47     $conn = new mysqli(HOST, USERNAME, PASSWORD, DBNAME, PORT);
48     if ($conn->connect_error)
49         die("发生致命错误!");
50     $stmt_sel = $conn->prepare("SELECT password FROM user WHERE userName = ?");
51     $stmt_sel->bind_param('s', $user);
52     $stmt_sel->execute();
53     $stmt_sel->bind_result($hash_pwd);
54     $result = false;
55     $stmt_upd = null;
56     $stmt_sel->fetch();
57     $stmt_sel->close();
58     if ($hash_pwd){
59         if (password_verify($pwd, $hash_pwd)) {
60             $stmt_upd = $conn->prepare("UPDATE user SET status = 'U' WHERE userName = ?");
61             $stmt_upd->bind_param('s', $user);
62             $stmt_upd->execute();
63             if ($stmt_upd->affected_rows == 1)
64                 $result = true;
65             else
66                 $result = false;
67         } else {
68             $result = false;
69         }
70     }
71     $conn->close();
72     return $result;
73 }
```

```
> manageDB.php > ...
33     function selStuUser(){
34         mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
35         $conn = new mysqli(HOST, USERNAME, PASSWORD, DBNAME, PORT);
36         if($conn->connect_error)
37             die("发生致命错误! ");
38         $sql = "SELECT userName, userType, status, name, stuNo, gender, phone, QQ
39                FROM user u, student s
40                WHERE u.pk = s.pk";
41         $resultSet = $conn->query($sql);
42         return $resultSet->fetch_all(MYSQLI_BOTH);
43     }
```



```
> welcomeStu.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>注册成功, 欢迎! </title>
6  </head>
7  <body>
8      欢迎,
9      <?php
10     echo $_COOKIE['userName'];
11     ?>!
12     <br>
13     <a href="allStudents.php">查看所有学生用户信息</a>
14     <br>
15     <a href="delUser.php">注销用户</a>
16 </body>
17 </html>
```



localhost:3000/6/welcomeStu.php

欢迎, user_2023!

[查看所有学生用户信息](#)

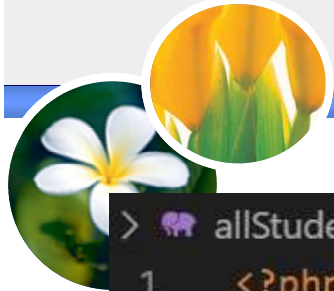
[注销用户](#)



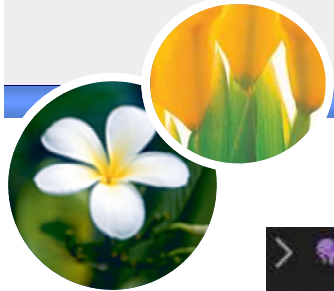
localhost:3000/6/allStudents.php

目前有注册学生用户3人

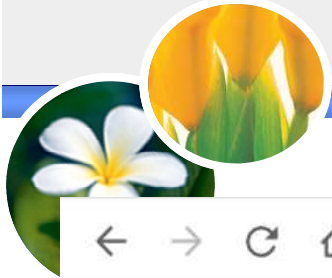
用户名	用户类型	是否有效	姓名	学号	性别	电话	QQ
user_1	S	U	姓名_1	21200101	1	1300000001	QQ1
user_2	S	U	姓名_2	21200102	1	1300000002	QQ2
user_2023	S	V	姓名_2023	2120010	1	1300000000	QQ



```
> allStudents.php
1  <?php    //allStudents.php
2      require_once("manageDB.php");
3      $allStu = selStuUser();
4  ?>
5  <table border="1 solid">
6      <caption>目前有注册学生用户<?=sizeof($allStu)?>人</caption>
7      <tr><th>用户名</th><th>用户类型</th><th>是否有效</th><th>姓名</th><th>学号</th>
8      |   <th>性别</th><th>电话</th><th>QQ</th></tr>
9      <?php
10     foreach ($allStu as $stu){
11         echo "<tr>";
12         echo "<td>{$stu["userName"]}</td>";
13         echo "<td>{$stu["userType"]}</td>";
14         echo "<td>{$stu["status"]}</td>";
15         echo "<td>{$stu["name"]}</td>";
16         echo "<td>{$stu["stuNo"]}</td>";
17         echo "<td>{$stu["gender"]}</td>";
18         echo "<td>{$stu["phone"]}</td>";
19         echo "<td>{$stu["QQ"]}</td>";
20         echo "</tr>";
21     }
22     ?>
23 </table>
```



```
> delUser.php
1  <?php    //delUser.php
2      require_once("manageDB.php");
3
4      if ($_POST['delUser'] == "确认注销"){
5          if (delUser($_COOKIE['userName'], $_POST['password'])){
6              echo <<<_END
7              成功注销用户{$_COOKIE['userName']}<br>
8              <a href="allStudents.php">查看所有学生用户信息</a>
9          _END;
10         }
11         else {
12             echo "注销用户{$_COOKIE['userName']}失败";
13         }
14     }else{
15         echo <<<_END
16         <form action="" method="post">
17             请输入密码: <input type="password" name="password">
18             <input type="submit" name="delUser" value="确认注销">
19         </form>
20     _END;
21     }
22  ?>
```



← → ↻ 🏠 ⓘ localhost:3000/6/welcomeStu.php

欢迎, user_2023!
[查看所有学生用户信息](#)
[注销用户](#)

→ ↻ 🏠 ⓘ localhost:3000/6/delUser.php

请输入密码:

← → ↻ 🏠 ⓘ localhost:3000/6/delUser.php

请输入密码:

← → ↻ 🏠 ⓘ localhost:3000/6/delUser.php

成功注销用户user_2023
[查看所有学生用户信息](#)

← → ↻ 🏠 ⓘ localhost:3000/6/allStudents.php

目前有注册学生用户3人

用户名	用户类型	是否有效	姓名	学号	性别	电话	QQ
user_1	S	U	姓名_1	21200101	1	1300000001	QQ1
user_2	S	U	姓名_2	21200102	1	1300000002	QQ2
user_2023	S	U	姓名_2023	2120010	1	130000000	QQ



数据库过滤器 表过滤器

主机: 127.0.0.1 数据库: web2022 表: user 数据 查询*

web2022Conn

- information_schema 160.0 KiB
- first
- mysql
- performance_schema
- sys
- web2022 160.0 KiB
 - classhour 32.0 KiB
 - semesterclass 32.0 KiB
 - student 32.0 KiB
 - teacher 32.0 KiB
 - user 32.0 KiB**
 - webapp

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

名称: user

注释: 为能够登录系统的内每位用户保存一条账号记录

字段: + 添加 - 删除 ▲ 向上 ▼ 向下

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释	校对规则	表达式	虚拟
1	pk	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	userName	VARCHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值		utf8_unicode_ci		
3	password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值		utf8_unicode_ci		
4	userType	CHAR	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'S'	S表示学生T表示教师	utf8_unicode_ci		

主机: 127.0.0.1 数据库: web2022 表: user 数据 查询*

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

自动增量: 26 默认字符校对: utf8_unicode_ci ☐ 转换数据

平均记录行长度: 引擎: InnoDB ☒

最大记录行数: 联合的表:

记录行校验值: ☐

记录行格式: DEFAULT ☐ INSERT 方法:



主机: 127.0.0.1 数据库: web2022 表: user 数据 查询*

基本 选项 **索引** 外键 分区 CREATE 代码 ALTER 代码

	名称	类型/长度	算法
+ 添加	> PRIMARY KEY	PRIMARY	BTREE
✕ 删除	> userName_UNIQUE	UNIQUE	BTREE
✕ 清除			
▲ 向上			
▼ 向下			

主机: 127.0.0.1 数据库: web2022 表: user 数据 查询*

web2022.user: 8 总记录数 (大约)

pk	userName	password	userType
7	601232	\$2y\$10\$MPH40LR4I5FgRVhSZTA82egisUD8Vgs....	T
17	user_1	\$2y\$10\$TqEKbh99eWfYqBbgGkeeT.q2pwsaCU...	S
18	user_2	\$2y\$10\$ZsK1eQjZVBWNGh.Mxhxkd.hVWIZxV.T...	S
22	user_3	\$2y\$10\$CuPbO6HuoBewwBHfyNNhU.tcYJ5zYP8...	S
24	user_4	\$2y\$10\$OJB0/wts.l3taldaXS0g2OCnvB7EKu4rk0...	S
27	user_5	\$2y\$10\$.H/bkW5Z/bAznajVwMp0mu8Y4tAb7Lu...	S
28	user_6	\$2y\$10\$tOTP8pJTUBD7x/IUIUUFpu23rGOViqqz...	S
29	user_7	\$2y\$10\$dktgShYoHnjtgJ49IDfUuXTwMzKS8YS...	S



数据库过滤器 表过滤器

主机: 127.0.0.1 数据库: web2022 表: student 数据 查询*

web2022Conn

- information_schema 160.0 KiB
- first
- mysql
- performance_schema
- sys
- web2022 160.0 KiB
 - classhour 32.0 KiB
 - semesterclass 32.0 KiB
 - student 32.0 KiB**
 - teacher 32.0 KiB
 - user 32.0 KiB
 - webapp

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

名称: student

注释: 保存使用系统的学生用户的特有信息

字段: + 添加 - 删除 ▲ 向上 ▼ 向下

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释	校对规则
1	pk	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	与users表的主键一致	
2	name	VARCHAR	32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值		utf8_unicode_ci
3	gender	CHAR	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL	F表示女M表示男	utf8_unicode_ci
4	stuNo	CHAR	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	学号	utf8_unicode_ci
5	phone	CHAR	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	电话号码	utf8_unicode_ci
6	QQ	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	无默认值	QQ号码	utf8_unicode_ci

主机: 127.0.0.1 数据库: web2022 表: student 数据 查询*

基本 选项 **索引** 外键 分区 CREATE 代码 ALTER 代码

+ 添加 - 删除 X 清除 ▲ 向上 ▼ 向下

名称	类型/长度	算法
> PRIMARY KEY	PRIMARY	BTREE
> stuNo_UNIQUE	UNIQUE	BTREE



主机: 127.0.0.1 数据库: web2022 表: student 数据 查询*

基本 选项 索引 外键 分区 CREATE 代码 ALTER 代码

	键名	字段	关联表	外联字段	UPDATE 时	DELETE 时
+ 添加	FK_student_user	pk	web2022.user	pk	CASCADE	CASCADE
× 删除						
× 清除						

主机: 127.0.0.1 数据库: web2022 表: student 数据 查询*

web2022.student: 7 总记录数 (大约) >> 下一个 显示所有 排序 字段

pk	name	gender	stuNo	phone	QQ
17	姓名_1	0	21200101	1300000001	QQ1
18	姓名_2	1	21200102	1300000002	QQ2
22	姓名_3	1	21200103	1300000003	QQ3
24	姓名_4	1	21200104	1300000004	QQ4
27	姓名_	1	21200105	1300000005	QQ5
28	姓名_6	1	21200106	1300000006	QQ6
29	姓名_7	0	21200107	1300000007	QQ7