




Mysql数据库

- 数据库相关概念
- Mysql数据库
- Sql语言
- Php+mysql

目录



Part
01

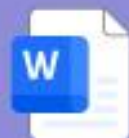
数据库相关概念

想一想?

假如让你来记录一个班级的学生信息
或是让你来记录一个公司的员工信息
你会选择哪种文件来做记录更加清晰明了?



新建 文本文
档.txt



新建 DOCX
文档.docx



新建 XLSX 工
作表.xlsx

什么是数据库？

是统一管理的、长期储存在计算机内的、有组织的相关数据的集合。数据库的基本特征包括：数据按一定的数据模型组织、描述和储存；数据间联系密切、冗余度较小；数据独立性较高；易扩展；可为各种用户共享。



数据库实际应用：

- 数据存储：数据库能够存储和管理大量的数据，将数据以一定的结构存储在计算机中，使得数据的存储、管理、检索更加高效和便捷。
- 数据管理：数据库可以对数据进行增加、删除、修改、查询等操作，并且可以对数据进行权限控制，保证数据的安全性和完整性。
- 数据分析：数据库可以连接各种数据源，对数据进行提取、转换、加载等操作，从而将数据转换成有用的信息，为决策提供支持



为什么要学习数据库？

应用程序产生的数据是在内存中的，如果程序退出或者是断电了，则数据就会消失。使用数据库是为了能够永久保存数据。当然这里指的是非内存数据库。

用普通文件存储行不行？



- 1、关系型数据库
- 2、非关系型数据库

关系型数据库系统

MySQL、Oracle、Microsoft SQL Server 和IBM DB2都是关系型数据库系统(database system)。除了管理数据，一个这样的系统还包括用来管理各种关系数据库的程序。通过SQL结构化查询语言来存取、管理关系型数据库的数据。



数据表

table

记录

字段、列


column

stuid	stuname	stusex	age
1	张三	男	20
2	李四	女	9
3	张柳	女	20
4	张四	男	38
5	刘海	男	30

记录

字段

学生信息表



Part
02

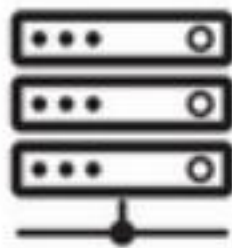
Mysql数据库



是一个开源的关系型数据库管理系统（RDBMS），由瑞典的MySQL AB公司开发，现属于Oracle旗下产品。它是目前最流行的关系型数据库之一，特别在WEB应用方面表现出色
开源性、高性能、简单易用、安全性、可扩展性、支持多种编程语言



- 1、数据库管理系统（最外层）：**DBMS**，专门管理服务器端的所有内容
- 2、数据库（第二层）：**Database**，专门用于存储数据的仓库（可以有很多个）
- 3、二维数据表（第三层）：**Table**，专门用于存储具体实体的数据
- 4、字段（第四层）：**Field**，具体存储某种类型的数据（实际存储单元）



命令行窗口

```
管理员: C:\Windows\system32\cmd.exe - mysql -u root -p
Microsoft Windows [版本 10.0.22631.3007]
(c) Microsoft Corporation。保留所有权利。

C:\Users\EDY>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.53 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



Navicat可以连接到MySQL服务器，进行数据库的创建、修改、删除等操作。

它支持数据表的创建、修改、删除以及数据的增删改查等操作。

Navicat还提供了可视化的查询构建工具，使得查询操作更加直观和方便。

Mysql默认的三个库

information_schema

performance_schema

mysql

对象 SCHEMATA @information_s...

开始事务 文本 筛选 排序 导入 导出 数据库名

CATALOG_NAME	SCHEMA_NAME	DEFAULT_CHARACTER_SET_NAME	DEFAULT_COLLATION_NAME	SQL_PATH
def	information_schema	utf8	utf8_general_ci	(Null)
def	challenges	gbk	gbk_chinese_ci	(Null)
def	chan	utf8	utf8_general_ci	(Null)
def	mysql	utf8	utf8_general_ci	(Null)
def	performance_schema	utf8	utf8_general_ci	(Null)
def	pikachu	utf8	utf8_general_ci	(Null)
def	qq_mail_db	utf8	utf8_general_ci	(Null)
def	security	gbk	gbk_chinese_ci	(Null)
def	test	latin1	latin1_swedish_ci	(Null)

对象 SCHEMATA @information_s... TABLES @information_sche...

开始事务 文本 筛选 排序 导入 导出 某个数据库对应的表名

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE	ENGINE
def	information_schema	CHARACTER_SETS	SYSTEM VIEW	MEMORY
def	information_schema	COLLATIONS	SYSTEM VIEW	MEMORY
def	information_schema	COLLATION_CHARACTER_SET_APPLICABILITY	SYSTEM VIEW	MEMORY
def	information_schema	COLUMNS	SYSTEM VIEW	MyISAM
def	information_schema	COLUMN_PRIVILEGES	SYSTEM VIEW	MEMORY
def	information_schema	ENGINES	SYSTEM VIEW	MEMORY
def	information schema	EVENTS	SYSTEM VIEW	MyISAM

对象 SCHEMATA @information_s... TABLES @information_sche... COLUMNS @information_sc...

开始事务 文本 筛选 排序 导入 导出 字段名

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_TYPE
def	information_schema	CHARACTER_SETS	CHARACTER_SET_NAME	1	CHAR(32)
def	information_schema	CHARACTER_SETS	DEFAULT_COLLATE_NAME	2	CHAR(32)
def	information_schema	CHARACTER_SETS	DESCRIPTION	3	TEXT
def	information_schema	CHARACTER_SETS	MAXLEN	4	INT(10)
def	information_schema	COLLATIONS	COLLATION_NAME	1	CHAR(32)
def	information schema	COLLATIONS	CHARACTER SET NAME	2	CHAR(32)

是指MySQL数据库中用于定义表的列时可以指定的数据类型，它们决定了列中可以存储什么类型的数据。MySQL支持多种数据类型，每种数据类型都有其特定的用途和限制。

数值类型

类型	存储（字节）	有符号数值取值范围	无符号数值取值范围
TINYINT	1	-128 ~ 127	0 ~ 255
SMALLINT	2	-32768 ~ 32767	0 ~ 65535
MEDIUMINT	3	-8388608 ~ 8388607	0 ~ 16777215
INT	4	-2147483648 ~ 2147483647	0 ~ 4294967295
BIGINT（整数值超过int时使用）	8	$-2^{63} \sim 2^{63}-1$	$0 \sim 2^{64}-1$

浮点数类型

FLOAT: 占用4个字节，用于表示单精度浮点数值。

DOUBLE: 占用8个字节，用于表示双精度浮点数值。（小数点后的数字更多，内存空间也会更多）

DECIMAL: 用于表示高精度的小数，其精度和计数方法可以指定，以适应特定的需求。（小数点后位更多，一般银行使用的较多）

日期和时间类型

类型	大小(bytes)	范围	格式	用途
DATE	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
TIME	3	-838:59:59/838:59:59	HH:MM:SS	时间值或持续时间
YEAR	1	1901/2155	YYYY	年份值
DATETIME	8	1000-01-01 00:00:00' 到 '9999-12-31 23:59:59	YYYY-MM-DD hh:mm:ss	混合日期和时间值
TIMESTAMP	4	1970-01-01 00:00:01' UTC 到 '2038-01-19 03:14:07	YYYY-MM-DD hh:mm:ss	混合日期和时间值，时间戳

字符串类型

类型	大小	用途
CHAR	0-255 bytes	定长字符串
VARCHAR	0-65535 bytes	变长字符串
TINYBLOB	0-255 bytes	不超过 255 个字符的二进制字符串
TINYTEXT	0-255 bytes	短文本字符串
BLOB	0-65 535 bytes	二进制形式的长文本数据
TEXT	0-65 535 bytes	长文本数据
MEDIUMBLOB	0-16 777 215 bytes	二进制形式的中等长度文本数据
MEDIUMTEXT	0-16 777 215 bytes	中等长度文本数据
LOBLOB	0-4 294 967 295 bytes	二进制形式的极大文本数据
LONGTEXT	0-4 294 967 295 bytes	极大文本数据

主键primary key:

用于确保表中的每一行都有可以唯一识别自己的一列或多列。

主键约束确保:

- 主键列的每个值都唯一
- 主键列不能NULL

外键:

外键约束（Foreign Key Constraint）是用来维护数据库表之间数据一致性和完整性的一种机制。它指定了一个表中的列（或列的组合）必须在另一个表的主键或唯一键列中有对应的值。通过外键约束，可以确保在一个表（称为子表或从表）中的值在另一个表（称为父表或主表）中已经存在，从而防止孤立数据的产生

AUTO_INCREMENT（自增）

用于为表中的某一列（通常是主键列）自动生成唯一的递增值。

NULL（为空）

NOT NULL（不为空）


非空约束，用于约束该字段的值不能为空。

默认:

☒ 自动递增

☐ 无符号

☐ 填充零



Part
03

SQL语言



SQL-结构化查询语言

- 1.数据定义语言（DDL）**：用于创建和修改数据库对象（如表和索引）的结构，如CREATE、ALTER、DROP等语句。
- 2.数据操纵语言（DML）**：用于对数据库中的数据进行新增、修改、删除等操作，如INSERT、UPDATE、DELETE等语句。
- 3.数据查询语言（DQL）**：用于对数据进行查询操作，如SELECT语句。
- 4.数据控制语言（DCL）**：用于控制用户对数据的访问权限，如GRANT和REVOKE语句

`show databases` : 查看所有数据库

`charset gbk` : 设置编码为gbk

`use 数据库名` : 切换到某个数据库

`select database()` : 查看当前所选数据库

`show tables` : 查看当前数据库所有的表

`desc 表名` : 查看某个表的描述

`select * from 表名` : 查看表的数据



新建数据库：

`create database （数据库名）；` 创建数据库

`create database if not exists （数据库名）；` 判断，如果不存在则创建

字符集：

字符（Character）是各种文字和符号的总称，包括各国家文字、标点符号、图形符号、数字等。

字符集（Character Set）在MySQL中是指一套用于存储、处理和检索文本数据的字符和编码规则。

它定义了数据库中可以使用的字符集合，以及这些字符如何映射到数字编码上。

创建数据库，并指定字符集：

`create database （数据库名） character set （字符集）；`

修改数据库的字符集：

`alter database （数据库名） character set （字符集）；`

查看数据库的字符集：

```
SHOW CREATE DATABASE （数据库名）；
```

删除数据库：

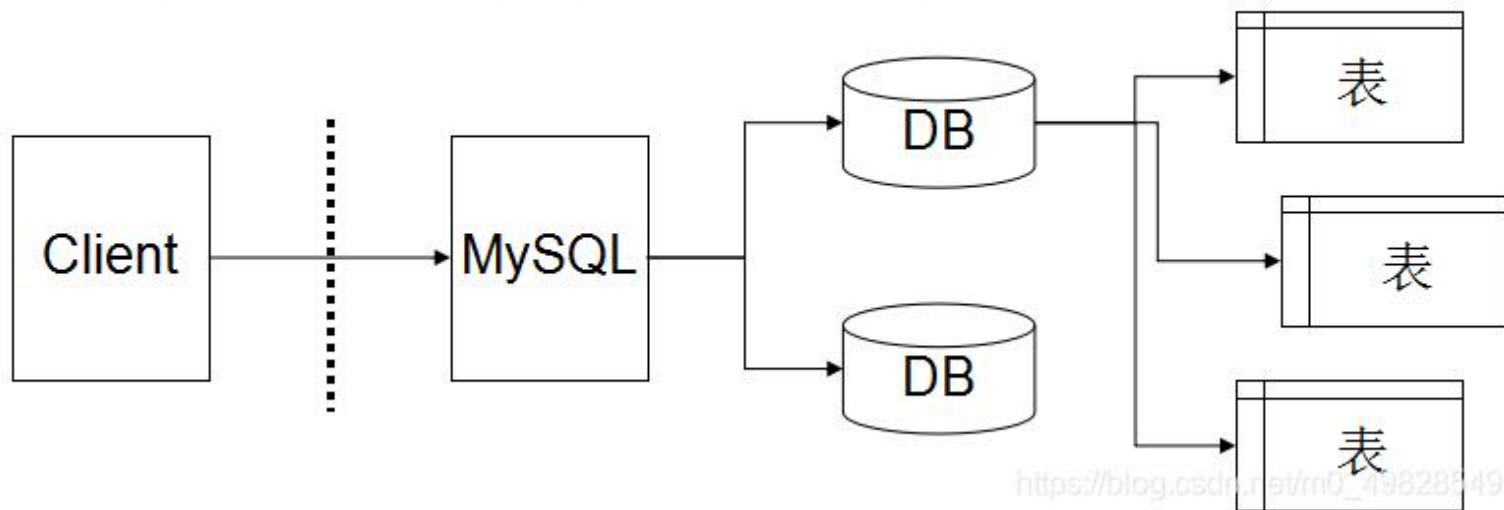
```
drop database （数据库名）；
```

使用数据库：

```
use （数据库名）；
```



- 数据库服务器、数据库和表的关系如图所示：



CREATE TABLE 创建表

ALTER 修改表

DROP 删除表

INSERT INTO

插入数据

UPDATE

修改数据

DELETE

删除数据



语法:

-- 查询表数据

SELECT (列名) FROM (表名);

--条件查询

SELECT (列名) FROM (表名) WHERE (条件);

--模糊查询

SELECT (列名) FROM (表名) WHERE (列名) LIKE ' ';
(%: 匹配零个或多个字符; _: 匹配单个字符。)

	id	NAME	price	brand	stock	insert_time
▶	1	华为手机	3999	华为	23	2088-03-10
	2	小米手机	2999	小米	30	2088-05-15
	3	苹果手机	5999	苹果	18	2088-08-20
	4	华为电脑	6999	华为	14	2088-06-16
	5	小米电脑	4999	小米	26	2088-07-08
	6	苹果电脑	8999	苹果	15	2088-10-25
	7	联想电脑	7999	联想	(Null)	2088-11-11

聚合查询：

COUNT(): 计算行数。

MAX(): 返回数值列中的最大值。

MIN(): 返回数值列中的最小值。

SUM(): 计算数值列的总和。

AVG(): 计算数值列的平均值。

排序查询：

SELECT (列)

FROM (表)

ORDER BY (列) DESC\ASC;

asc为升序，desc为降序；

group by

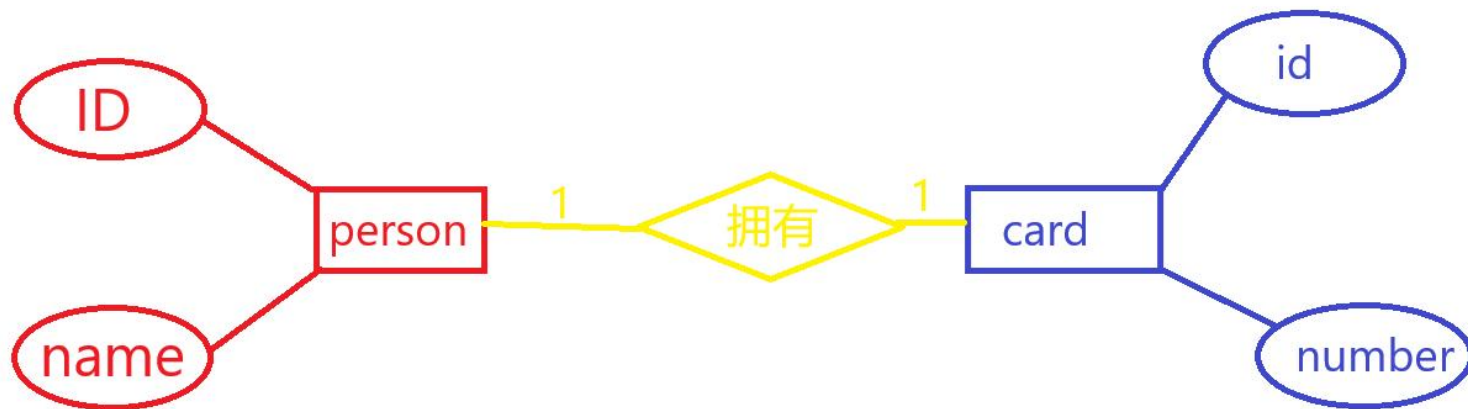
分页查询：

SELECT * FROM 表名 LIMIT (页码-1)*每页数量, 每页数量;

mysql表关系：

一对一表关系：

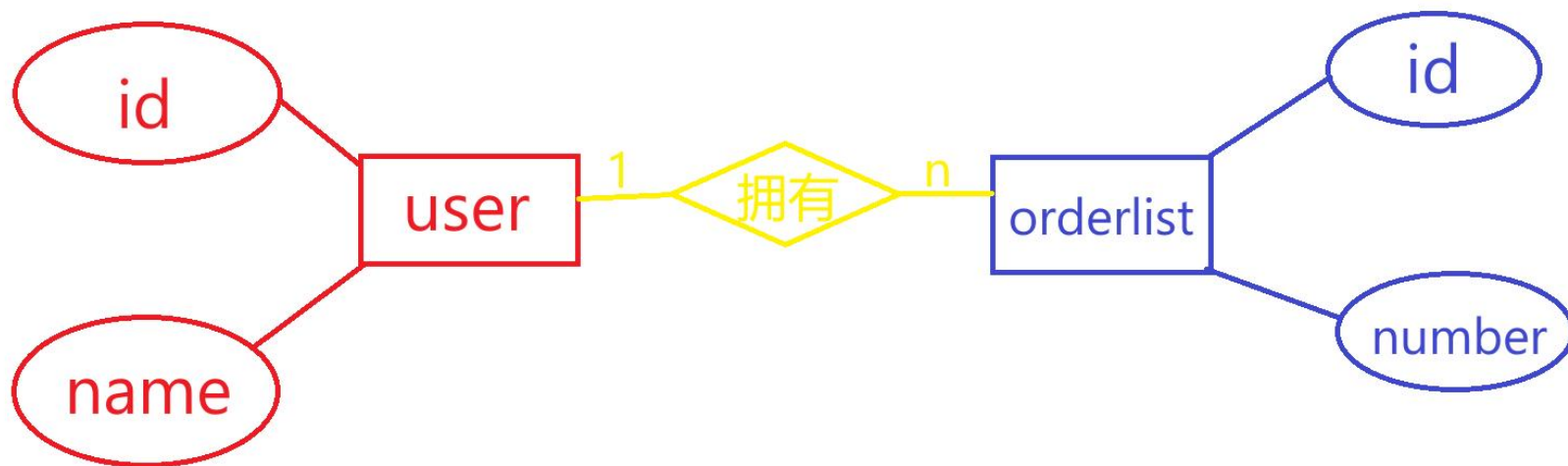
一对一（1:1）表关系是一种较为特殊的表关系，表示一个表中的一条记录与另一个表中的一条记录有唯一的对应关系。这种关系通常用于将一个表的某些列分离到一个新的表中，以便保持数据的清晰性和模块化。



mysql表关系：

一对多表关系：

这意味着在一个表中，可以有多个记录与另一个表中的单个记录相关联。常用于表示两个表之间的连接，其中一个表中的多行可以与另一个表中的单行相关联。这种关系通常通过外键（Foreign Key）来实现。



mysql表关系：

多对多表关系：

多对多（Many-to-Many）表关系是一种常见的数据关系模式，它描述了两个实体集之间的一种复杂关系，即一个实体集中的多个元素可以与另一个实体集中的多个元素相关联。



多表查询：


Union查询：

用于合并两个或多个 **SELECT** 语句的结果集，同时去除重复的记录。每个 **SELECT** 语句必须拥有相同数量的列，列的数据类型也必须兼容。**UNION** 默认为 **UNION ALL**，但 **UNION ALL** 会包含所有的重复记录，而 **UNION** 会去除重复的记录



子查询：

子查询是一种在**SQL**语言中嵌套查询下层的程序模块。当一个查询是另一个查询的条件时，这个被嵌套的查询就被称为子查询。子查询也被称为内查询或内部选择，而包含子查询的语句则被称为主查询或外部查询。



Part
04

Php+mysql



编程接口:

在PHP中，与SQL数据库进行交互通常涉及使用数据库扩展库，如PDO（PHP Data Objects）或MySQLi。这些库提供了与数据库建立连接、执行查询和处理结果的方法。

数据库操作一般步骤:

- 1、与mysql建立连接
- 2、选择要操作的数据库
- 3、进行增删查改操作
- 4、关闭连接

mysql相关函数

- mysqli_connect(); 连接数据库
- mysqli_select_db(); 选择数据库
- mysqli_query(); 执行SQL语句
- mysqli_assoc(); 获取单条数据，返回关联数组
- mysqli_num_rows(); 获取查询的条数
- mysqli_close(); 关闭数据库



谢谢观看！

