**01**

database:数据库 date:日期

数据库：是按照一定的形式来组织，存储数据，目的是便于增删改查。

层次型数据库；

RDBMS关系型数据库；

数据库发展史：网状数据库->层次型数据库->关系型数据库（当前主流）->非关系型数据库

关系型数据库的逻辑结构

Server->Database->Table->Row（行）->Column（列）

MySQL开源中小型数据库，可用于各种操作系统

Linux+Apache+MySQL+PHP（LAMP组合）

2.使用MySQL数据库

Orcle分支http://www.mysql.com

Martin分支http://www.mysql.com

mysql的部署结构

服务器端的软件:负责存储/维护数据——工行

客户端软件：负责向客户端服务器发起增删改查——atm机

安装并使用mysql服务器软件（xampp套装）

（1）服务器端：C:\xampp\mysql\bin\mysqld.exe启动文件。mysqld.exe（服务器端）

确保MySQL的端口号3306不被占用

（mysqld->deamon精灵，守护者）

（2）客户端：C:\xampp\mysql\bin\mysql.exe（不推荐）.mysqld.exe（客户端端）

进入数据库的方式mysql.exe　-h127.0.0.1　-P3306　-uroot　-p(简化版mysql -uroot)注意：结尾的时候不能加分号。

-h　host　主机

-P　port　端口

-u　user　用户名

-p　password　密码

（3）mysql 常用的管理符号

quit 退出服务器连接

show databases ;显示服务器上当前所有数据库。

use 数据库名称；进入指定数据库

show Tables ;显示数据库中所有的表格

desc +表名 ;描述表中有哪些列（表头）

练习

查询出phpmyadmin数据库中有哪些列

4.SQL语句

SQL:Structured Query Language结构化查询语言，用于操作关系型数据库服务器中的数据——增删改查

ANSI

SQL命令的两种执行方式

两者不区分大小写

（1）交互模式：客户端输入一行，点击回车，服务器执行一行。**只适用于临时性查看数据**。

（2）脚本模式：客户端把要执行的多行命令编写在一个文本文件中，一次性提交给服务期执行，适用于批量的操作数据。

mysql -uroot < 拖拽进来的文件.sql 回车

练习：使用脚本模式向数据库服务器提交多条sql语句。

SQL语句规范

（1）每条语句必须以英文符号的分号作为结尾，一条语句可以跨越多行，见到分号认为语句结束。use命令除外。

（2）若某一条语句出现了语法错误，则此条语句一级后边所有语句不会再执行。

（3）SQL命令不区分大小写，习惯上数据库关键字都使用大写，非关键字都使用小写。

（4）SQL命令中可以使用单行注释（#...）和多行注释（/\*../\*），注释的内容不会被服务器所执行。

常用的sql命令:

(1)删除指定的数据库，如果存在的话

DROP DATABASE IF EXISTS tedu;

(2)创建新的数据库

CREATE DATABASE tedu;

(3)进入刚刚创建的数据库tedu

USE tedu;

（4）创建数据表

CREATE TABLE laptop(

id INT,

title VARCHAR(128),

price INT,

spec VARCHAR(32)

);

（5）查询数据表中所有的数据

SELECT \* FROM +表名

（6）从数据表中删除数据

DELETE FROM 表名 WHERE id=' '

练习:放弃数据库tedu，创建数据库tedu，进入该数据库，创建学生表student，包含编号（sid）

姓名（name），

课后练习:编写一个脚本文件lianxi.sql，若果存在先放弃数据xuezi，创建数据库xuezi，进入数据库，创建一个用户表user，包含uid，uname，upwd，email，phone，

gender，userName,registerTime,isOnline,插入5条记录你然后删除2条记录。

02

复习：

MySQL服务器端和客户端

server->database->table->row->column

交互模式 输入一行 返回结果

show tables

MySQL增删改查

INSERT INTO student VALUES(.....);#增加数据

DELETE FROM student WHERE id='...';#删除数据

UPDATE student SET uname='...' WHERE id='...';#更改数据

SELECT \* FROM student;#查询数据。

练习：

创建一个1.sql文件，放弃数据库dangdang，如果存在；创建数据库dangdang，进入数据库；创建一个保存图书的数据表book，包含bid，bname，author，price，pubDate，isAddsale;插入4本书信息。

计算机如何存储英文字符

ASCII：总共有128个，对所有的英文字符以及标点符号进行编码。

latin-1：总共有256个，兼容ASCII，同时对欧洲符号进行编码。nysql默认使用的就是这个。

计算机如何存储中文字符

GB2312：对常用的6千多个汉字进行编码，兼容ASCII码。

GBK：对2万多个汉字进行编码，兼容GB2312。

BIG5：台湾繁体字编码标准，兼容ASCII码。

Unicode:对世界上主流的语言中常用的字符进行编码，兼容ASCII码，不兼容GB2312，GBK,BIG5等系列。具体分为UTF-8 ,UTF-16 ,UTF-32等存储方案。

乱码产生的原因

mysql中默认使用Latin-1编码，而Latin-1不存在中文编码

解决MySQL数据库中的乱码问题，三个统一

（1）.sql脚本文件另存为的编码

（2）客户端连接服务器使用的编码

（3）服务器端，创建数据库存储所使用的编码

今日目标

列类型——稍显杂乱

列约束

1.MySQL中的列类型

创建数据表的时候，指定的列可以存储的数据类型

(1)数值类型——数值类型的值使用的时候可以不加引号

TINYINT 微整型，占用一个字节，-128~127

SMALLINT 小整型 占用2个字节，-32768~32767

INT 整型 站用4个字节 -2147483648~2147483647

BIGINT 大整型占用8个字节

DECIMAL 定点小数

FLOAT ：单精度浮点小数，占用4个字节，最大是3.4E38范围远远大于INT，可能产生四舍五入。

DOUBLE：双精度浮点小数，占用8个字节范围远远大于BIGINT,可能产生四舍五入。

DECIMAL(M.D)（8.2）定点小数，不会产生计算误差，M代表总的有效位数（不包含小数点），D代表小数点后的有效位数

999999.99（价格、工资）

BOOL布尔类型，只能是TRUE(1)和FALSE(0)

（2）日期时间类型

DATE 日期类型 ' 2018-7-3' 加引号

TIME 时间类型 14:35:30'

DATATIME 日期时间类型 '2018-7-3 14:35:30'

（3）字符串类型——必须添加引号

VARCHAR(M)变长字符串,不会产生空间的浪费，操作速度相对较慢，M最大值是65535

CHAR(M)定长字符串，可能会产生空间的浪费，常用语固定的长度，例如手机号，身份证号，操作速度远比varchar快

TEXT 大型变长字符串，最多能存2G

|  |  |  |
| --- | --- | --- |
|  | CHAR(4) |  |
| a | a\0\0\0 |  |
| ab | ab\0\0 |  |
| abc | abc\0 |  |
| abcd | abcd |  |

练习1：使用合适的列类型。

创建2.sql文件，编写sql语句，创建数据库xuezi，保存的编码（字符集）UTF8，进入该数据库，创建保存笔记本的信息表laptop，包含lid，image，title，price，stockCount（库存量），sheifTime（上架时间），

练习2.使用合适的列类型

创建3.sql 编写sql、语句；重新创建数据库tedu保存的编码格式为utf8进入该数据库

创建保存公司部门信息的表dept，包含（did，ndname，empCount）

10 研发部 3 20 人事部 2 30 行政部 2

创建保存员工信息表格emp，包含（eid，ename， gender，birthday，salary，deptld隶属部门）

2.列约束

Constraint：约束；MySQL可以对插入的数据进行特定的验证，只有满足条件才允许插入到数据表中，否者被认为是非法的数据，禁止插入。

例如：

在MySQL中一共提供了6种列约束

（1）主键约束——PRIMARY KEY

声明了主键列上的值不能出现重复,表中查询的记录会按照主键上的值由小到大排序——加快查找的速度，不能插入NULL。

注意；一个表中只能有一个主键。

（2）唯一约束-UNIQUE

声明了唯一约束的列不能插入重复的值，允许插入多个NULL值，

**课后作业**

创建lianxi.sql文件，创建数据库xuezi，使用utf8编码

创建笔记本分类表xz\_laptop\_family(fid,fname,laptopCount)

10 联想 3 20 戴尔 2 30 小米 2

创建笔记本信息表xz\_laptop(lid,title,price,spec规格，details详细介绍，shelfTime（上架时间） ，isOnsale是否在售，familyid隶属的分类)

添加 7款笔记本信息记录。

03

复习

中文乱码——utf8编码

脚本.sql文件，.....

数值类型

tinyint/smallint/int/bigint

float/double/decimal/bool（true或1/false或0）

日期类型

date/time/datetime

字符串类型

varchar/char/text

列约束

主键约束 primary key

唯一约束 unique

常用的SQL语句

drop database if exists xuezi;

create database xuezi;

use xuezi;

**标准SQL语句分类**

（1）DDL：Data Define Language定义数据列

CREATE/DROP/ALERT(修改)/TRUNCHATE(截断)

（2）DML:Data Manipulate Language操作数据

INSERT/DELETE/UPDATE

(3)DQL:Data Query Language查询数据

SELECT/

(4)DCL:Data Control Language控制用户权限

GRANT(授权)/REVOKE(收权)

1.列约束

（1）默认值约束——DEFAULT

可以使用默认值DEFAULT来为列指定默认值；使用方法；

1.INSERT INTO xz\_laptop\_family VALUES(- - -);

2. INSERT INTO xz\_laptop\_family(fid,fname) VALUES(- -);

(2).非空约束——NOT NULL

插所要入的列中的值不允许为空的

（4）检查约束——CHECK

检查约束可以对输入的数据范围进行检查

CREATE TABLE student(age TINYINT CHECK(age>=0 AND age<=100));

**注意：MySQL中不支持检查约束**

（5）外键约束——FOREIGN KEY

声明了外键约束的列，取值必须在另一个表

（允许使用NULL，一次或多次）

2.项目中如何存储日期

保存指定日期记录计算机元年(1970-1-1 0:0:0)的毫秒数

通常存储的日期类型使用 BIGINT

目的:为了解决国际化，不同国家显示不同的时间格式。性别也是同样的原理，存储的时候（1男）或(0女)

练习：列约束的使用，实现如下功能；

创建数据库tedu，使用utf8编码，进入该数据库

创建部门表dept（did，dname，）

10 研发部 20市场部 30运用部 40测试部

创建员工表emp（eid，ename，gender，birthday，salary，deptld，隶属部门id）

插入14条记录，分别隶属于部门10.20.30其中编号为14的员工不在任何一个部门（null）

3.MySQL中的自增列

AUTO\_INCREMENT;自动增长，加入一个列声明为自增列，无需手动赋值，直接赋值为null，会获取当前的最大值，新纪录就会执行+1插入。

注意：只适用于整数型的主键列，允许手动赋值。

**简单查询**

1,简单查询——只查询特定的列

示例；

SELECT ename,birthday,salary FROM emp;

SELECT deptld,ename,salary,eid FROM emp;

2.简单查询——所有的列(\*代表所有的列)

SELECT \* FROM emp;

3.简单查询——给列取名

SELECT ename AS 姓名,salary AS 工资 FROM emp;

4.简单查询——只显示不同的记录并且合并相同的记录

SELECT DISTINCT deptld FROM emp;

SELECT DISTINCT sex FROM emp;

5. 简单查询——在查询的时候执行计算

示例；计算2\*3/4

SELECT 2\*3/4 AS 结果;

示例；查询所有员工的姓名及年薪

SELECT ename AS 姓名,salsry\*12 AS 年薪 FROM emp;

SELECT ename AS 名字,(salary+500)\*12+5000 年薪 FROM emp;

6.简单查询——查询结果的排序(ASC升序，DESC降序)

示例；查询所有员工信息，结果按照工资由低到高

SELECT \* FROM emp ORDER BY salary ASC;

降序排列

SELECT \* FROM emp ORDER BY salary DESC;

练习；查询所有员工信息，结果按照生日由小到大

SELECT \* FROM emp ORDER BY birthday DESC;

练习；查询所有员工信息，结果按照姓名排序

SELECT \* FROM emp ORDER BY ename ASC;

练习；查询所有员工信息，结果按照生日由小到大，如果生日相同的员工，再按姓名排序，

SELECT \* FROM emp ORDER BY birthday DESC,ename ASC;

练习；查询所有员工信息, 结果按照工资由小到大，要求所得女员工必须在男员工的前面

SELECT \* FROM emp ORDER BY sex ASC,salary ASC;

7.简单查询——条件查询

示例：查出编号为5的员工信息

SELECT \* FROM emp WHERE eid=5;

示例；查出名字叫KING的员工的eid,salary,birthday

SELECT eid,salary,birthday FROM emp WHERE ename='KING';

SELECT \* FROM emp WHERE deptId=20;

示例；查出所有男员工的信息

SELECT \* FROM emp WHERE sex=1;

示例；查出工资大于等于5000的员工信息

SELECT \* FROM emp WHERE salary >=5000;

MySQL的比较运算符(> < <= >= != = )

练习；查出1991-1-1之后出生的员工信息

SELECT \* FROM emp WHERE birthday>='1991-1-1';

练习；查出不在10号部门的所有员工信息

SELECT \* FROM emp WHERE deptId!=10;

练习；查出没有部门的员工信息

SELECT \* FROM emp WHERE deptId IS NULL;（不能使用deptId=NULL）

练习；查出有明确部门的员工信息

SELECT \* FROM emp WHERE deptId IS NOT NULL;

练习；查出工资大于6000的女员工所有信息

SELECT \* FROM emp WHERE salary>6000 AND sex=0;

练习；查出工资大于6000的1991-1-1以后的女员工所有信息

SELECT \* FROM emp WHERE salary>6000 AND sex=0 AND birthday >'1991-1-1';

练习；查出工资大于等于5000小于等于5999员工所有信息

SELECT \* FROM emp WHERE salary>=5000 AND salary<=5999;

SELECT \* FROM emp WHERE salary BETWEEN 5000 AND 5999;

练习；查出工资小于4000和大于8000员工所有信息

SELECT \* FROM emp WHERE salary<4000 OR salary>8000;

练习；查出在1991年出生的员工信息

SELECT \* FROM emp WHERE birthday BETWEEN '1991-1-1' AND '1991-12-31';

SELECT \* FROM emp WHERE birthday >='1991-1-1' AND birthday <='1991-12-31';

练习；查询出在1990年之前和1993年之后出生的员工信息

SELECT \* FROM emp WHERE birthday<'1990-1-1' OR birthday>'1993-12-31';

练习:查询出在10和30号部门的信息

SELECT \* FROM emp WHERE deptId=10 OR deptId=30;

SELECT \* FROM emp WHERE deptId IN(10,30);

练习：查询出在10 30 50 80 号部门的员工信息

SELECT \* FROM emp WHERE deptId IN(10,30,50,80);

练习：查询出不在10和30号部门的员工信息

SELECT \* FROM emp WHERE deptId NOT IN(10,30);

8.简单查询——模糊条件查询(%表示任意多个字符>=0；\_任意一个字符1)

示例：查询出姓名中包含字母E的员工所有信息

SELECT \* FROM emp WHERE ename LIKE '%e%';

练习：查询出姓名以E结尾的员工所有信息

SELECT \* FROM emp WHERE ename LIKE '%e';

练习：查询出姓名以E开头的员工所有信息

SELECT \* FROM emp WHERE ename LIKE 'e%';

练习：查询出姓名倒数第二个字母是E的员工信息

SELECT \* FROM emp WHERE ename LIKE '%e\_';

9.简单查询——分页查询

|  |  |  |  |
| --- | --- | --- | --- |
| 页码 | 开始 | 数量 | 结束 |
| 第1页 | 0 | 5 | 4 |
| 第2页 | 5 | 5 | 9 |
| 第3页 | 10 | 5 | 14 |
| 第4页 | 15 | 5 | 19 |
| 第5页 | 20 | 5 | 24 |

求出具体从那一页开始：（页码-1）\*数量

第1页SELECT \* FROM emp LIMIT 0,5;

第2页SELECT \* FROM emp LIMIT 5,5;

第3页SELECT \* FROM emp LIMIT 10,5;

LIMIT start,count;注意事项start和count不能加引号

start 开始的值

count 查询的数量

第n页SELECT \* FROM emp LIMIT (n-1)\*5,5;

练习：

查询所有员工的分页信息，假设每页显示7条记录；分别查询前四页；写出查询每一页的sql语句。

SELECT \* FROM emp LIMIT 0,7;

SELECT \* FROM emp LIMIT 7,7;

SELECT \* FROM emp LIMIT 14,7;

SELECT \* FROM emp LIMIT 21,7;

**简单查询综合**

排序 ORDER BY

条件 WHERE

分页 LIMIT start,count

要求：查询员工中，条件为男员工，按工资降序排列，在结果中显示前3条记录

SELECT \* FROM WHERE sex=1 ORDER BY salary DESC LIMIT 0,3;