



蜗牛学院

重构IT职业教育新生态

MySQL

- 数据库基础知识

目录

C O N T E N T S

1

数据库基本概念

2

数据库工具的使用

3

数据库操作

4

数据库表操作

- 一．数据库基本概念
- 二．创建及管理数据库，数据库表操作
- 三．数据库约束与数据库设计
- 四．DML语句(增加，删除，修改操作)
- 五．简单查询
- 六．复杂查询(统计、子查询、多表连接查询)
- 七．高级特性（存储过程 函数 游标 触发器 事务 视图 索引）

本章目标

- 一．理解关系型数据库的基本概念
- 二．掌握关系模型
- 三．掌握MySQL安装及连接
- 四．熟悉结构化查询语言定义及其分类
- 五．掌握数据库管理操作
- 六．掌握创建数据表，掌握MySQL的数据类型
- 七．掌握数据库表操作

为何需要数据库

存储数据的方法

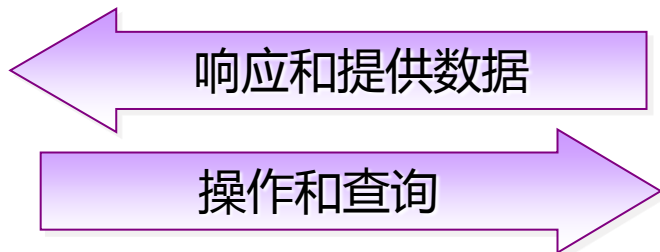
- 第一种方法：用大脑来记住数据
- 第二种方法：写在纸上
- 第三种方法：写在计算机的内存中
- 第四种方法：写成磁盘文件
-

数据库能干什么

- 存储大量数据，方便检索和访问
- 保持数据信息的一致、完整
- 共享和安全
- 通过组合分析，产生新的有用信息



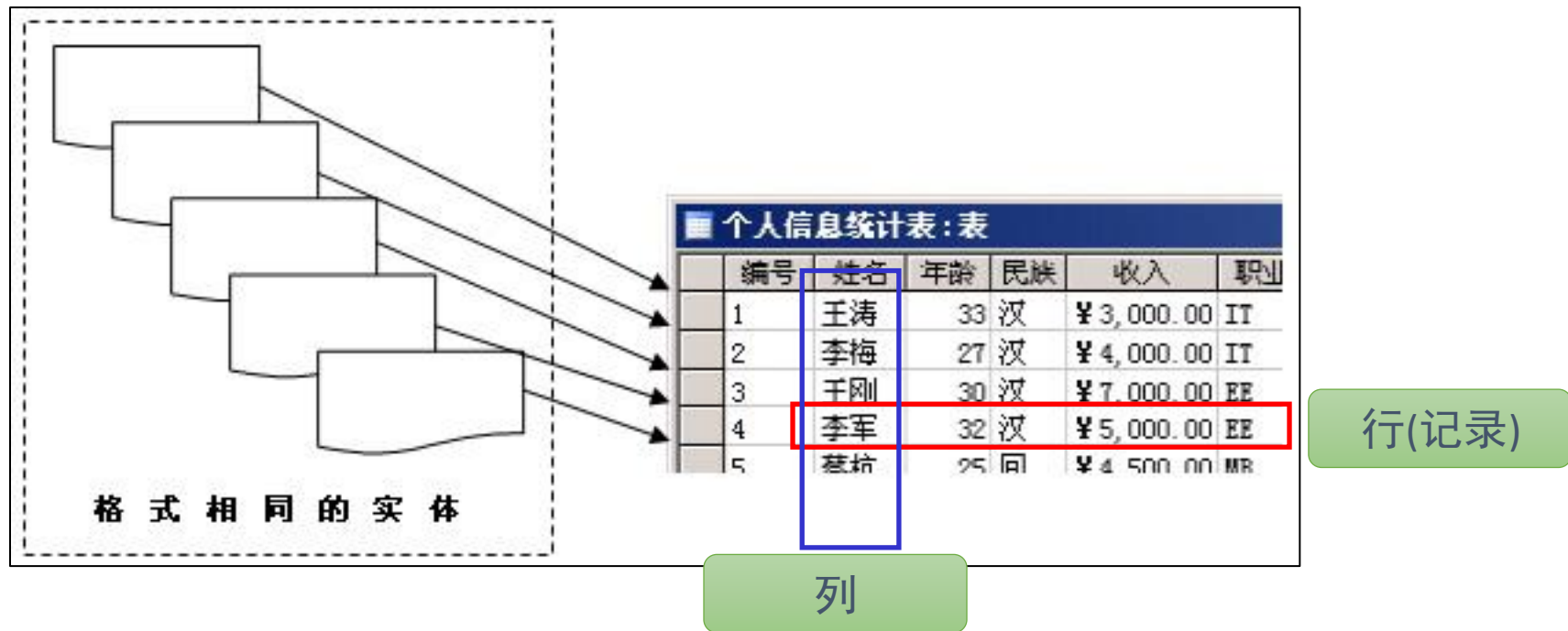
应用程序



数据库服务器

- 应用程序
 - 作用：响应操作并显示结果、向数据库请求数据
 - 要求：美观、操作简单方便
- 数据库
 - 作用：存储数据、检索数据、生成新的数据
 - 要求：统一、安全、性能等

- 萌芽阶段——文件系统
 - 使用磁盘文件来存储数据
- 初级阶段——第一代数据库
 - 出现了网状模型、层次模型的数据库
- 中级阶段——第二代数据库
 - 关系型数据库和结构化查询语言
- 高级阶段——新一代数据库
 - “关系-对象” 型数据库



➤ 一对一关系

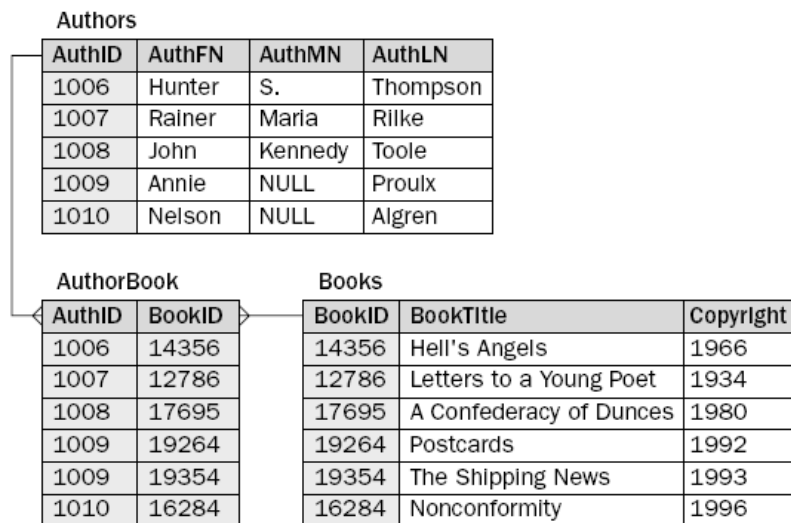
当两个表中第一个表中的一行只与第二个表中的一行相关，同时第二个表中的一行也只与第一个表中的一行相关时，我们称两个表之间是一对一关系。

Authors				AuthorsBlos		
AuthID	AuthFN	AuthMN	AuthLN	AuthID	Born	Died
1001	Edith	NULL	Wharton	1001	1862	1937
1002	Kate	NULL	Chopin	1002	1850	1904
1003	T.	S.	Eliot	1003	1888	1965
1004	Joyce	Carol	Oates	1004	1938	NULL
1005	Samuel	L.	Clemens	1005	1835	1910

- ◆ 通常，一对一关系是比较少见的关系类型。很多数据库也很少包含一对一关系。我们碰到的最常用的关系类型是一对多关系。

➤ 一对多关系

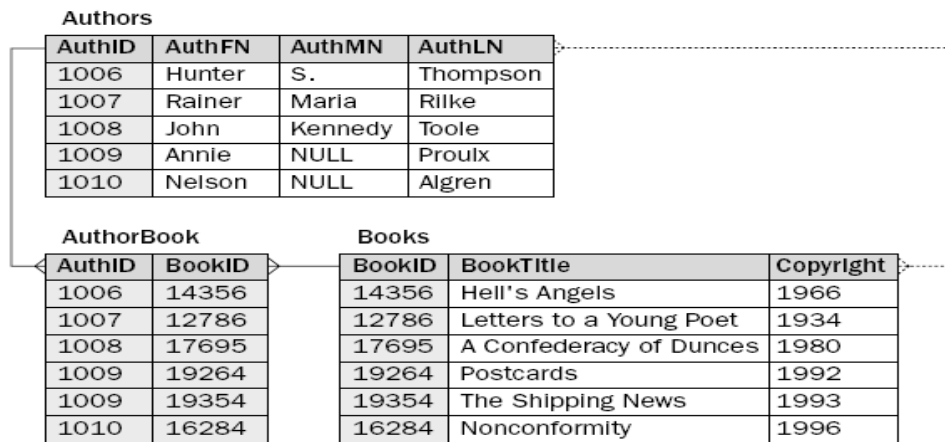
一对多关系与一对一关系的不同之处在于，第一个表中的行可以与第二个表、中的一到多个行相关联，但是第二表中的一行只能与第一个表中的一行相关联。



- ◆ 我们通常把一对多关系中，“多边”的表称为从表，把“一边”的表称为主。一对多关系是我们在数据库中最常看到的一种关系类型。

➤ 多对多关系

多对多关系是指两个表中，第一个表的一行可以与第二个表中的一到多个行相关联，同时，第二个表中的一行可以与第一个表中的一到多个行相关联。



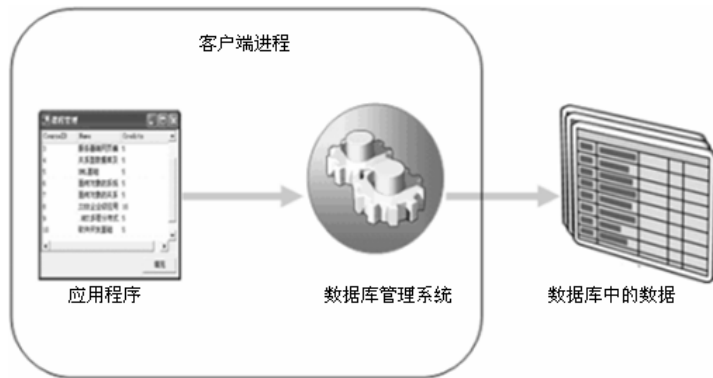
- 在大多数关系型数据库管理系统中，多对多关系通过在关系的两个表间增加第三个表来实现。我们把第三个表称为连接表（Junction Table），连接表充当支持多对多关系的两个表之间的桥梁，多对多关系是逻辑而不是物理实现的。

- 什么是数据库管理系统

关系型数据库只是一个保存数据的容器，大多数数据库依靠一个称为数据库管理系统（Database Management System，简称DBMS）的软件来管理数据库中数据。管理关系型数据库的软件称为关系型数据库管理系统（Relational Database Management System，简称RDBMS）。数据库应用程序通过RDBMS与关系型数据库进行交互。

- 本地数据库管理系统

本地数据库管理系统又称桌面型数据库管理系统。在这种系统模式下，RDBMS与数据库应用程序运行在同一客户端的进程中，其模型如下图所示。



345 systems in ranking, September 2018

Sep 2018	Rank		DBMS	Database Model	Score		
	Aug 2018	Sep 2017			Sep 2018	Aug 2018	Sep 2017
1.	1.	1.	Oracle +	Relational DBMS	1309.12	-2.91	-49.97
2.	2.	2.	MySQL +	Relational DBMS	1180.48	-26.33	-132.13
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1051.28	-21.37	-161.26
4.	4.	4.	PostgreSQL +	Relational DBMS	406.43	-11.07	+34.07
5.	5.	5.	MongoDB +	Document store	358.79	+7.81	+26.06
6.	6.	6.	DB2 +	Relational DBMS	181.06	-0.78	-17.28
7.	↑ 8.	↑ 10.	Elasticsearch +	Search engine	142.61	+4.49	+22.61
8.	↓ 7.	↑ 9.	Redis +	Key-value store	140.94	+2.37	+20.54
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	133.39	+4.30	+4.58
10.	10.	↓ 8.	Cassandra +	Wide column store	119.55	-0.02	-6.65
11.	11.	11.	SQLite +	Relational DBMS	115.46	+1.73	+3.42
12.	12.	12.	Teradata +	Relational DBMS	77.38	-0.02	-3.52
13.	13.	↑ 16.	Splunk	Search engine	74.03	+3.53	+11.45
14.	14.	↑ 18.	MariaDB +	Relational DBMS	70.64	+2.34	+15.17
15.	15.	↓ 13.	Solr	Search engine	60.20	-1.69	-9.71
16.	↑ 18.	↑ 19.	Hive +	Relational DBMS	59.63	+1.69	+11.02
17.	17.	↓ 15.	HBase +	Wide column store	58.47	-0.33	-5.87
18.	↓ 16.	↓ 14.	SAP Adaptive Server +	Relational DBMS	58.04	-2.39	-8.71
19.	19.	↓ 17.	FileMaker	Relational DBMS	55.00	-5.69	-5.69
20.	↑ 21.	↑ 22.	Amazon DynamoDB +	Multi-model i	53.34	+1.69	+15.52

常见数据库

Oracle

Oracle公司的产品；产品免费、服务收费；
金融、通信、生产制造、快速消费品零售、
物流运输、医疗、互联网、政府；



SQL Server

针对不同用户群体的多个版本；易用性好；
政府、生产制造、快速消费品零售、医疗、
互联网为主；



DB2

IBM公司的产品；支持多操作系统、多种类型
的硬件和设备；金融为主；



MySQL

开放源代码；
互联网为主，近2年开始扩展到金融、通信、
生产制造、快速消费品零售、物流运输、
医疗、政府等行业。



常见数据库

MongoDB

非关系型数据库，{编号:1,姓名:张三,年龄:20,性别:男}



Redis

内存数据库，非关系型数据库，可进行持久化到磁盘，缓存数据库



Solr

帮助进行大数据搜索，集群化



Hbase、Hive

大数据系统常使用的数据库



MySQL 是最流行的关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。在WEB应用方面 MySQL 是最好的RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

- ✓MySQL 是开源的，不需要支付额外的费用。
- ✓MySQL 支持大型的数据库。支持5000万条记录的数据仓库，32位系统表文件最大可支持4GB，64位系统支持最大的表文件为8TB。
- ✓MySQL 使用标准的SQL数据语言形式。
- ✓MySQL 可以运行于多个系统上，并且支持多种语言。这些编程语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby和Tcl等。
- ✓MySQL 对PHP有很好的支持，PHP是目前最流行的Web开发语言。

MySQL数据库的下载：

所有平台的 MySQL 下载地址为：<https://dev.mysql.com/downloads/mysql/>. 挑选你需要的 *MySQL Community Server* 版本及对应的平台。

MySQL数据库的安装：

典型安装 超级用户（管理员）：root 默认端口：3306

自定义安装 更改安装路径 修改默认字符集为UTF8

MySQL数据库登录方式：

1.命令行方式：

打开cmd，输入以下命令：`mysql -u root -p`,之后输入密码；

基本命令：

Show databases;

2.MySQL方式：

打开MySQL自带连接工具，输入root密码即可登录；

Use 数据库名；

Show tables；

3.第三方工具：

Navicat、SQLYOG

名词 解释

DB	DataBase 数据库
DBA	DataBase Administrator 数据库管理员
DBMS	DataBase Management System 数据库管理系统
RDBMS	Relational DataBase Management System 关系型数据库管理系统
SQL	Structured Query Language 结构化查询语言
DDL	Data Definition language 数据定义语言
DML	Data manipulation language 数据操纵语言
DQL	Data query language 数据查询语言
DCL	Data Control Language 数据控制语言

- SQL是结构化查询语言（**Structured Query Language**）的英文缩写，是一种用于管理关系型数据库，并与数据库中的数据进行通讯的计算机语言。SQL是最普遍实现和应用的数据库语言，并且已经成为数据库管理的**标准语言**。
- 我们可以使用SQL管理关系型数据库，例如创建和删除数据库存储结构或者访问和修改数据库安全设置。对于程序员来说，不管使用哪种语言开发程序，不管使用哪种数据库存储数据，SQL的知识都是**必不可少的**。
- 关系型数据库理论和SQL均起源于IBM的一个研究实验室。1979年Relational Software, Inc.（即现在的Oracle公司）发布了第一个支持SQL的商业关系型数据库产品**Oracle**。1986年ANSI（美国国家标准化协会）发布了一个正式的标准，命名为SQL-86。1989年，ANSI将标准更新为SQL-89，1992年更新为**SQL-92**，1999年继续更新为SQL-99。到目前为止，最新版本的SQL标准为SQL：2016。
- SQL提供所有基本的数据操作，但是，不同的RDBMS之间存在一些差别。例如，虽然当前SQL的标准是SQL：2016，但是还是有很多RDBMS只支持早期的SQL-92版本。这意味着SQL：2016的某些部分也许不被我们所使用的RDBMS支持。同时，每个RDBMS实现标准的方法**可能有细微的区别**。

语言分类	名称	详细描述
1、数据查询语句(DQL)	Select	数据选择查询
2、数据操纵语句(DML)	Insert; Update; Delete;	插入 更新 删除
3、数据定义语句(DDL)	Create; Alter; Drop; Rename; Truncate;	新建 修改结构 删除 重命名 删除
4、数据控制语言(DCL)	Grant; Revoke;	授权 解除授权

- 什么是数据库？数据库可以做什么？
- DBA、RDBMS、SQL是什么意思？
- 目前有哪些常见的数据库？各自的主要特点是什么？
- MySQL数据库的特点有哪些？
- SQL的分类及语句名称是什么？

● 创建数据库：CREATE DATABASE [IF NOT EXISTS] 数据库名;

- ◆ 其中，数据库名在服务器中必须是唯一的，并且符合标识符规则。所谓标识符规则指：
- ◆ 第一个字符必须是下列字符之一：
 - ◆ Unicode 标准 3.0 所定义的字母（Unicode 中定义的字母包括拉丁字母 a-z 和 A-Z，以及来自其它语言的字母字符）。
 - ◆ 下划线（_）、at 符号（@）或者数字符号（#）。
- ◆ 后续字符可以是：
 - ◆ Unicode 标准 3.0 所定义的字母。
 - ◆ 来自基本拉丁字母或其它国家/地区脚本的十进制数字。
 - ◆ at符号（@）、美元符号（\$）、数字符号或下划线。
- ◆ 标识符不能是所用RDBMS的保留字。
- ◆ 不允许嵌入空格或其它特殊字符。

- 查看数据库

```
SHOW DATABASES;
```

- 删除数据库

```
DROP DATABASE 数据库名;
```

- 连接数据库

```
USE 数据库名;
```

- 修改数据库编码集

语法：ALTER DATABASE db_name CHARACTER SET = charset_name

例：ALTER DATABASE test CHARACTER SET = gb2312;

创建表

创建表的基础SQL语法如下：

```
CREATE TABLE 表名(  
    列名 数据类型 约束,  
    列名 数据类型 约束  
) [ENGINE = engine_name引擎 | [DEFAULT] CHARACTER SET [=] charset_name];
```

其中，表名在一个数据库中必须是唯一的，并且符合标识符规则。列名在一个表内必须是唯一的，并且符合标识符规则。列的数据类型决定了什么样的数据可以存储在列中，而列的约束定义了创建列时的限制条件。Mysql引擎类型有InnoDB和MyISAM两种。

举例：

```
CREATE TABLE Friend (Name VARCHAR(50), PhoneNo VARCHAR(15));  
CREATE TABLE demo(id INT, name VARCHAR(10)) ENGINE=INNODB DEFAULT CHARACTER  
SET utf8;
```

- 整数数据类型

类型	大小	范围(有符号)	范围(无符号)	用途
TINYINT	1B	(-128 , 127)	(0 , 255)	极小整数值
SMALLINT	2B	(-32 768 , 32 767)	(0 , 65 535)	小整数值
MEDIUMINT	3B	(-8 388 608 , 8 388 607)	(0 , 16 777 215)	小整数值
INT	4B	(-2 147 483 648 , 2 147 483 647)	(0 , 4 294 967 295)	整数值
BIGINT	8B	(-9 233 372 036 854 775 808 , 9 223 372 036 854 775 807)	(0 , 18 446 744 073 709 551 615)	大整数值

- 浮点数据类型

类型	大小	范围	精度	用途
FLOAT	4B	$(-3.40E-38, 3.40E+38)$	7位小数	单精度浮点数
DOUBLE	8B	$(-8\ 388\ 608, 8\ 388\ 607)$	15位小数	双精度浮点数

- 定点数据类型

类型	大小	范围	精度	用途
DECIMAL(M,D)	17B	$(-10\text{的}38\text{次方}-1, 10\text{的}38\text{次方}-1)$	30位小数	大浮点数

注意：

1. M为总数，D为小数，M必须大于D

- 字符串类型

类型	大小	范围	用途
CHAR(M)	M	0-255	字符型
VARCHAR(M)	M+1B	0-65535	字符型
TINYTEXT	LENGTH+2B	0-255	文本型
TEXT	LENGTH+2B	0-65535	文本型（无默认值）
BINARY(M)	M	0-M	0-M变长字符串
VARBINARY(M)	M+1B	0-M	0-M变长字符串

- 二进制数据类型

类型	大小	范围	用途
TINYBLOB	255	0-255	二进制大对象
BLOB	65K	0-65KB	二进制大对象
MEDIUMBLOB	16M	0-16M	二进制大对象
LOB	4G	0-4G	二进制大对象

- 日期和时间数据类型

类型	大小	格式	范围
YEAR	1B	YYYY	1901 - 2155
DATE	3B	YYYY-MM-DD	1000-01-01 - 9999-12-31
TIME	3B	HH:MM:SS	-838 : 59 : 59 - 838:59 : 59
TIMESTAMP	4B	YYYY-MM-DD HH:MM:SS	1970-01-01 00:00:01 - 2038
DATETIME	8B	YYYY-MM-DD HH:MM:SS	1000-01-01 00:00:00 - 9999-12-31 23:59:59

- 日期和时间数据类型插入示例

```
CREATE TABLE demo(  
    id INT,  
    reg_datetime DATETIME,  
    reg_time TIME,  
    reg_date DATE,  
    reg_year YEAR  
)
```

```
INSERT INTO table1 VALUES(1, '2018-03-29 23:23:34', '09:45:34', '2018-03-29', '2018')
```

- 查看数据库中的所有表

```
SHOW TABLES;
```

- 查看表的定义

```
SHOW CREATE TABLE 表名;
```

- 删除表

➤ 删除表的基础SQL语法如下：

```
DROP TABLE 表名;
```

- 通过外键约束连接在一起的表不能被删除。在删除表之前，必须先删除约束。在删除表时，我们必须要是表的所有者或者对该表有管理员权限。

- 复制表

- 在DB中，下面的语句创建一个名为new_aimtable的新表，该表是复制了table表的数据和结构（列、数据类型）：

```
CREATE TABLE 新表名 AS ( SELECT * FROM 被复制表名);
```

- 如果是只复制表结构，不复制数据。

```
CREATE TABLE 新表名 LIKE 被复制表名 ;
```

- 修改表名：

```
alter table 表名 RENAME [TO | AS] new_tbl_name;
```

- 添加列：

```
alter table 表名 ADD [COLUMN] col_name column_definition;
```

- 更改列的定义：

```
alter table 表名 CHANGE [COLUMN] old_col_name new_col_name  
column_definition; -- 修改列名及列类型
```

```
alter table 表名 MODIFY [COLUMN] col_name column_definition ; -- 只修改列类型
```

- 删除列：

```
alter table 表名 DROP [COLUMN] col_name;
```

- 思考：在创建表时下列列一般定义为何种数据类型？
 - 姓名、地址、电话号码等：
 - 年龄：
 - 出生日期：
 - 照片：
 - 薪水：
- 练习：新建User表，添加以下字段：
 - 1. id int ；
 - 2.name char 长度16 ；
 - 3.age int ；
 - 4.gender char 长度10 ；
 - 5.birthday date ；
 - 5.city varchar 长度16。



谢谢聆听

蜗牛学院，只为成就更好的你！

