



下载APP



01 | 存储：一个完整的数据存储过程是怎样的？

2021-03-08 朱晓峰

MySQL 必知必会

[进入课程 >](#)**讲述：朱晓峰**

时长 13:54 大小 12.73M



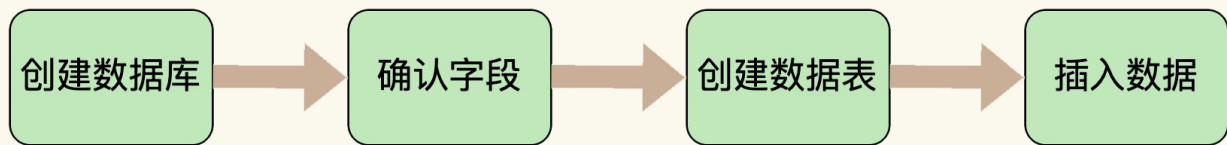
你好，我是朱晓峰。今天，我想跟你聊一聊 MySQL 是怎么存储数据的。

存储数据是处理数据的第一步。在咱们的超市项目中，每天都要处理大量的商品，比如说进货、卖货、盘点库存，商品的种类很多，而且数量也比较大。只有正确地把数据存储起来，我们才能进行有效的处理和分析，进而对经营情况进行科学的评估，超市负责人在做决策时，就能够拿到数据支持。否则，只能是一团乱麻，没有头绪，也无从着手。

那么，怎样才能把用户各种经营相关的、纷繁复杂的数据，有序和高效地存储起来呢？



在 MySQL 中，一个完整的数据存储过程总共有 4 步，分别是创建数据库、确认字段、创建数据表、插入数据。



接下来，我就给你详细讲解一下这个过程的每一步，帮你掌握 MySQL 的数据存储机制。

先提醒你一句，这节课最后有一个视频，我在视频里演示了今天讲到的所有操作。我建议你学完文字以后，跟着视频实操一下。

好了，话不多说，我们现在开始。

创建 MySQL 数据库

数据存储的第一步，就是创建数据库。

你可能会问，为啥我们要先创建一个数据库，而不是直接创建数据表呢？

这是个很好的问题。其实啊，这是因为，从系统架构的层次上看，MySQL 数据库系统从大到小依次是数据库服务器、数据库、数据表、数据表的行与列。

安装程序已经帮我们安装了 MySQL 数据库服务器，所以，我们必须从创建数据库开始。

数据库是 MySQL 里面最大的存储单元。数据表、数据表里的数据，以及我们以后会学到的表与表之间的关系，还有在它们的基础上衍生出来的各种工具，都存储在数据库里面。
没有数据库，数据表就没有载体，也就无法存储数据。

下面我就来给你具体介绍下，怎么在我们安装的 MySQL 服务器里面创建、删除和查看数据库。

1. 如何创建数据库？

创建数据库，我们已经在上节课介绍过了，你可以在 Workbench 的工作区，通过下面的 SQL 语句创建数据库 “demo”：

```
1 CREATE DATABASE demo;
```

2. 如何查看数据库？

下面我们来看一下，如何查看数据库。

在 Workbench 的导航栏，我们可以看到数据库服务器里的所有数据库，如下图所示：

你也可以在 Workbench 右边的工作区，通过查询语句，查看所有的数据库：

```
1 mysql> SHOW DATABASES;
2 +-----+
3 | Database |
4 +-----+
5 | demo     |
6 | information_schema |
7 | mysql    |
8 | performance_schema |
9 | sys      |
10 +-----+
11 5 rows in set (0.00 sec)
```

看到这儿，你是不是觉得很奇怪，为什么 Workbench 导航栏里面的数据库只有两个（我们创建的数据库“demo”和安装完 MySQL 就有的数据库“sys”）呢？

换句话说，为什么有的数据库我们可以在 Workbench 里面看到，有的数据库却必须通过查询语句才可以看到呢？要弄明白这个问题，你必须要知道这些数据库都是干什么的。

“demo”是我们通过 SQL 语句创建的数据库，是我们用来存储用户数据的，也是我们使用的主要数据库。

“information_schema”是 MySQL 系统自带的数据库，主要保存 MySQL 数据库服务器的系统信息，比如数据库的名称、数据表的名称、字段名称、存取权限、数据文件所在的文件夹和系统使用的文件夹，等等。

“performance_schema” 是 MySQL 系统自带的数据库，可以用来监控 MySQL 的各类性能指标。

“sys” 数据库是 MySQL 系统自带的数据库，主要作用是，以一种更容易被理解的方式展示 MySQL 数据库服务器的各类性能指标，帮助系统管理员和开发人员监控 MySQL 的技术性能。

“mysql” 数据库保存了 MySQL 数据库服务器运行时的系统信息，比如数据文件夹、当前使用的字符集、约束检查信息，等等。

如果你是 DBA，或者是 MySQL 数据库程序员，想深入了解 MySQL 数据库系统的相关信息，可以看下 [🔗 官方文档](#)。

话说回来，为什么 Workbench 里面我们只能看到 “demo” 和 “sys” 这 2 个数据库呢？其实啊，这是因为，Workbench 是图形化的管理工具，主要面向开发人员，“demo” 和 “sys” 这 2 个数据库已经够用了。如果有特殊需求，比如，需要监控 MySQL 数据库各项性能指标、直接操作 MySQL 数据库系统文件等，可以由 DBA 通过 SQL 语句，查看其它的系统数据库。

确认字段

数据存储流程的第二步是确认表的字段。

创建好数据库之后，我们选择要导入的 Excel 数据文件，MySQL 会让我们确认新表中有哪些列，以及它们的数据类型。这些列就是 MySQL 数据表的字段。

MySQL 数据表由行与列组成，一行就是一条数据记录，每一条数据记录都被分成许多列，一列就叫一个字段。每个字段都需要定义数据类型，这个数据类型叫做字段类型。

这样一来，每一条数据记录的每一个片段，就按照字段的定义被严格地管理起来了，从而使数据有序而且可靠。MySQL 支持多种字段类型，字段的定义会影响数据的取值范围、精度，以及系统的可靠性，下节课我会重点给你讲一讲字段的定义。这里你只要选择系统默认的字类型，就可以了。

创建数据表

数据存储流程的第三步，是创建数据表。

当我们确认好了表的字段，点击下一步，Workbench 就帮助我们创建了一张表。

MySQL 中的数据表是什么呢？**你可以把它看成用来存储数据的最主要工具**。数据表对存储在里面的数据进行组织和管理，使数据变得有序，并且能够实现高效查询和处理。

虽然 Workbench 帮助我们创建了一个表，但大多数情况下，我们是不会先准备一个 Excel 文件，再通过 Workbench 的数据导入来创建表的，这样太麻烦了。**我们可以通过 SQL 语句，自己来创建表。**

具体咋做呢？我来介绍一下。

首先，在 Workbench 的工作区，输入以下 SQL 语句：

```
1 CREATE TABLE demo.test
2 (
3     barcode text,
4     goodsname text,
5     price int
6 );
```

[复制代码](#)

执行这个 SQL 语句之后，就能创建出一个与导入的 Excel 表一样的 MySQL 数据表了。

这里有 2 点需要你格外注意一下。

创建表的时候，最好指明数据库。否则，如果你没有选中数据库，Workbench 会提示错误；要是你当前选中的数据库不对，还可能把表创建到错误的数据库中。

不要在最后一个字段的后面加逗号 “,”，这也是初学者容易犯的错误。

下面我们就来聊一聊，查看数据表的结构、查看数据库中的表的方法。

1. 如何查看表的结构？

创建好了表，再来看看如何查看表的结构。

我们运行下面的 SQL 语句：

```
1 DESCRIBE demo.test;
```

[复制代码](#)

运行结果如下：

```
1 mysql> DESCRIBE demo.test;
2 +-----+-----+-----+-----+-----+-----+
3 | Field      | Type | Null | Key | Default | Extra |
4 +-----+-----+-----+-----+-----+-----+
5 | barcode    | text | YES  |     | NULL    |       |
6 | goodsname  | text | YES  |     | NULL    |       |
7 | price      | int  | YES  |     | NULL    |       |
8 +-----+-----+-----+-----+-----+-----+
9 3 rows in set (0.00 sec)
```

[复制代码](#)

我来解释下这些信息都代表什么意思。

Field：表示字段名称。

Type：表示字段类型，这里 barcode、goodsname 是文本型的，price 是整数类型的。

Null：表示这个字段是否允许是空值（NULL）。这里你一定要注意，在 MySQL 里面，空值不等于空字符串。一个空字符串的长度是 0，而一个空值的长度是空。而且，在 MySQL 里面，空值是占用空间的。

Key：我们暂时把它叫做键。

Default：表示默认值。我们导入的表的所有的字段都允许是空，默认值都是 NULL。

Extra：表示附加信息。

关于字段类型和 Key，后面我会具体讲解，这里你只需要知道它们的含义就可以了。

2. 如何查看数据库中的表？

创建完成后，怎么查看表有没有真的被创建出来呢？

我们可以通过 Workbench 的导航栏，点击数据库下面的“Tables”，找到这个数据库中的所有数据表。另外，我们也可以在工作区，通过 SQL 语句查询某个数据库中的数据表。

先用 USE 语句选择数据库：

```
1 USE demo;
```

[复制代码](#)

运行后，进入 demo 数据库，然后用 SHOW 语句查询这个数据库中所有的表：

```
1 SHOW TABLES;
```

[复制代码](#)

运行后结果如下：

```
1 mysql> SHOW TABLES;
2 +-----+
3 | Tables_in_demo |
4 +-----+
5 | test           |
6 +-----+
7 1 row in set (0.00 sec)
```

[复制代码](#)

这样我们就可以看到数据库“demo”里面，只有一个数据表“test”。

3. 如何设置主键？

讲到了数据表，我就一定要给你讲一讲主键。因为主键可以确保数据的唯一性，而且能够减少数据错误。

举个简单的小例子。主键就像是咱们的身份证号码，它是唯一的。每个身份证号码只对应唯一的一个人。同样，每一个人只有唯一的身份证号码。

MySQL 中数据表的主键，是表中的一个字段或者几个字段的组合。它主要有 3 个特征：

必须唯一，不能重复；

不能是空；

必须可以唯一标识数据表中的记录。

一个 MySQL 数据表中只能有一个主键。虽然 MySQL 也允许创建没有主键的表，但是，**我建议你一定要给表定义主键，并且养成习惯。因为主键可以帮助你减少错误数据，并且提高查询的速度**（后面我会专门用一节课的时间介绍“怎么用好主键”）。

我来举个例子，假设我们有这样一张表：

我们给它起个名字，叫“goodsmaster”，意思是商品表。说到这儿，你可能注意到了，我的表名、字段名都用的是英文。其实，MySQL 也允许数据表名称、字段名称用中文，但我还是**建议你用英文，原因有 2 个：一是书写方便；二是不容易出错**。如果用汉字，涉及到编码问题，就会容易出现错误。

那么，在这个表里，哪个字段是主键呢？

商品名称“goodsname”行不行呢？不行，原因是重名的商品太多了，比如“笔”，大家都可以生产一种叫“笔”的商品，各种各样的笔，不同规格的，比如长的、短的；不同材料的，比如墨水的、铅芯的……由于可能重复，商品名称和数据记录之间不能形成一一对应的关系，所以“goodsname”不能作为主键。同样，价格“price”重复的可能性很大，也不能做主键。

商品条码“barcode”能不能成为这个数据表的主键呢？

好像可以。商品的条码都是由中国物品编码中心统一编制的，一种商品对应一个条码，一个条码对应一种商品。这不就是一一对应的关系吗？

在实际操作中，有例外的情况。比较典型的就是用户的门店里面有很多自己生产或者加工的商品。比如，馒头、面条等自产食品，散装的糕点、糖果等称重商品，等等。为了管理方便，门店往往会自己给它们设置条码。这样，很容易产生重复、重用的现象。

这么说，商品条码“barcode”也有重复的可能，也不能用做主键。


那么，如果数据表中所有的字段都有重复的可能，我们怎么设置主键呢？答案是**我们可以自己添加一个不会重复的字段来做主键。**

比如在上面的例子中，我们就可以添加一个字段，字段类型是整数，我们给它取个名字叫商品编号“itemnumber”。而且，我们可以每次增加一条新数据的时候，让这个字段的值自动加 1，这样就永远不会重复了，如下表所示：

我们添加字段商品编号“itemnumber”为主键，这样，我们的商品表“goodsmaster”就有了主键。

在 Workbench 中，我们可以通过修改表结构，来增加一个主键字段：

你也可以通过一条 SQL 语句，修改表的结构，来增加一个主键字段：

 复制代码

```
1 ALTER TABLE demo.test
2 ADD COLUMN itemnumber int PRIMARY KEY AUTO_INCREMENT;
```

我简单解释下这些关键字的含义。

ALTER TABLE，表示修改表；

ADD COLUMN，表示增加一列；


PRIMARY KEY，表示这一列是主键；

AUTO_INCREMENT，表示每增加一条记录，这个值自动增加。一会儿讲到添加数据的时候，我还会详细介绍一下它。

插入数据

数据存储流程的第四步，也是最后一步，是把数据插入到表当中去。

Workbench 的数据导入功能，可以帮助我们吧 Excel 的数据导入到表里面，那么，我们自己怎么向数据表中插入一条数据呢？我们可以借助 SQL 语句。

 复制代码

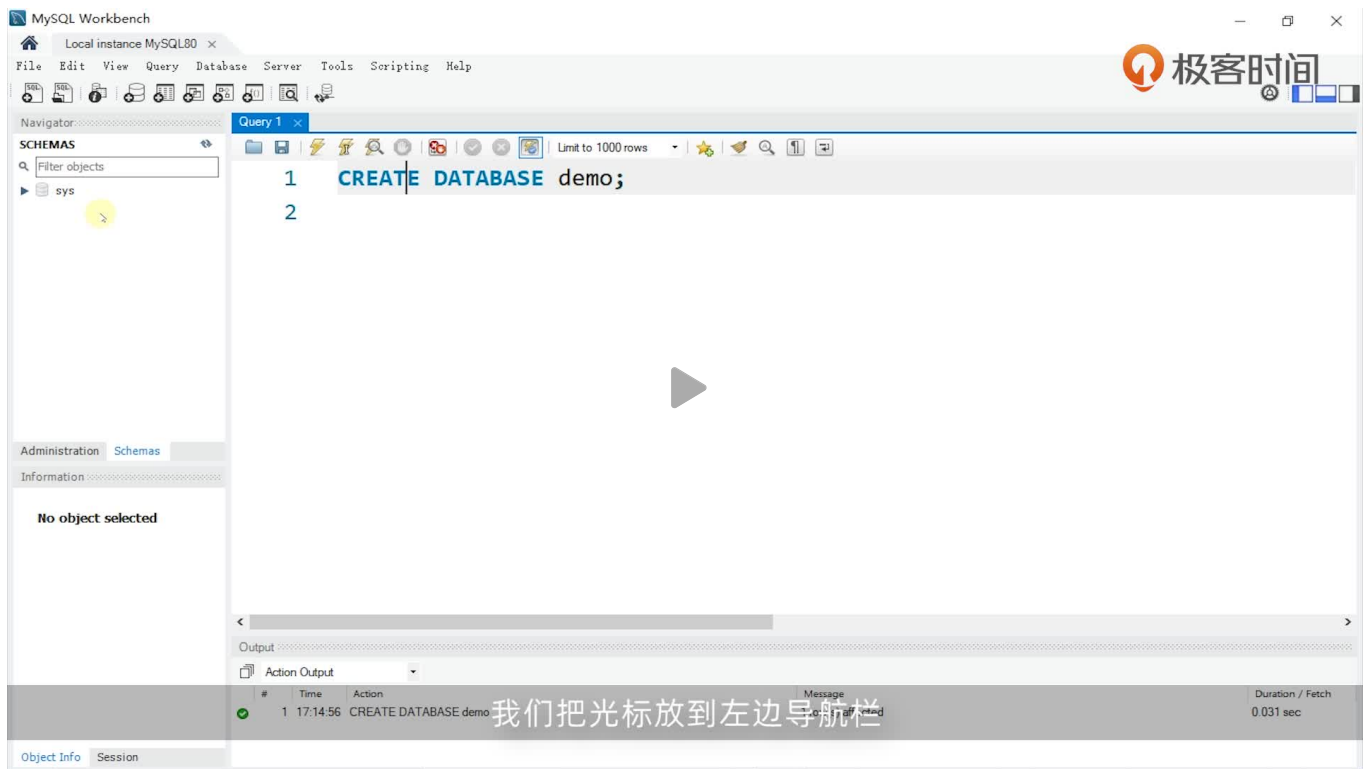
```
1 INSERT INTO demo.test  
2 (barcode,goodsname,price)  
3 VALUES ('0001','本',3);
```

这里的 INSERT INTO 表示向 demo.test 中插入数据，后面是要插入数据的字段名，VALUES 表示对应的值。

在添加数据的时候，有 2 点需要你格外注意一下。

1. 要插入数据的字段名也可以不写，但是我建议你不要怕麻烦，**一定要每次都写**。这样做的好处是可读性好，不易出错，而且容易修改。否则，如果你记不住表的字段，就只能去查表的结构，才能知道值所对应的字段了。
2. 由于字段 itemnumber 定义了 AUTO_INCREMENT，所以我们插入一条记录的时候，不给它赋值，系统也会自动给它赋值。而且，每次赋值，都会在上次的赋值基础上，自动增加 1。你也可以在插入一条记录的时候给 itemnumber 赋值，由于它是主键，新的值必须与已有记录的 itemnumber 值不同，否则系统会提示错误。

最后，我还专门录制了一段视频，把今天讲到的实际操作演示了一遍，你可以跟着视频再演练下。



总结

今天，我们学习了数据存储的完整过程，包括创建数据库、创建数据表、确认字段和插入数据。建议你跟着文字和视频实际操作一下，熟练掌握存储数据的方法。

在进行具体操作的时候，我们会用到 8 种 SQL 语句，我再给你汇总下。

复制代码

```
1  -- 创建数据库
2  CREATE DATABASE demo;
3  -- 删除数据库
4  DROP DATABASE demo;
5  -- 查看数据库
6  SHOW DATABASES;
7  -- 创建数据表：
8  CREATE TABLE demo.test
9  (
10     barcode text,
11     goodsname text,
12     price int
13 );
14 -- 查看表结构
15 DESCRIBE demo.test;
16 -- 查看所有表
17 SHOW TABLES;
18 -- 添加主键
19 ALTER TABLE demo.test
20 ADD COLUMN itemnumber int PRIMARY KEY AUTO_INCREMENT;
```

```
21 -- 向表中添加数据
22 INSERT INTO demo.test
23 (barcode,goodsname,price)
24 VALUES ('0001','本',3);
```

最后，我还想再给你讲一讲 MySQL 中 SQL 语句的书写规范。

MySQL 以分号来识别一条 SQL 语句结束，所以，**你写的每一条 SQL 语句的最后，都必须有一个分号，否则，MySQL 会认为这条语句没有完成，提示语法错误。**

所以，我建议你写在 SQL 语句时遵循统一的样式，以增加可读性，减少错误。如果你不是很清楚具体的规范，可以点击这个 [🔗 链接](#) 学习下。

思考题

我想请你思考一下，假设用户现在要销售商品，你能不能帮它设计一个销售表，把销售信息（商品名称、价格、数量、金额等）都保存起来？

欢迎在留言区写下你的思考和答案，我们一起交流讨论。如果你觉得今天的内容对你有所帮助，也欢迎你分享你的朋友或同事，我们下节课见。

提建议

12.12 大促

每日一课 VIP 年卡

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 环境准备 | 带你安装MySQL和图形化管理工具Workbench

下一篇 02 | 字段：这么多字段类型，该怎么定义？

精选留言 (15)

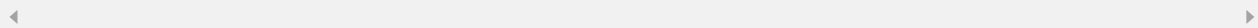
[写留言](#)**Gavin**

2021-03-09

老师，程序员的第一性原理是什么？

展开 ▾

作者回复: 我以为，程序员的第一性原理应该是实际业务需求，技术服务于业务，一切从实际需求出发。



9

**青生先森**

2021-03-08

老师，不要太优秀啊，很接地气的专栏。

展开 ∨

作者回复: 感谢鼓励



8



封志强

2021-03-08

老师说的很细，👏👏👏

展开 ∨

作者回复: 谢谢鼓励



6



星空下

2021-03-09

老师，面试遇到的一个问题，`select count(*) from t;`
`t`中有`id`(主键), `name`, `age`, `sex`4个字段。假设数据10条，对`sex`添加索引。用`explain` 查看执行计划发现用了`sex`索引，为什么不是主键索引呢？主键索引应该更快的

作者回复: 问题很好，下面是我的理解，供你参考：

1. MySQL Innodb的主键索引是一个B+树，数据存储在叶子节点上，10条数据，就有10个叶子节点。
2. `sex`索引是辅助索引，也是一个B+树，不同之处在于，叶子节点存储的是主键值，由于`sex`只有2个可能的值：男和女，因此，这个B+树只有2个叶子节点，比主键索引的B+树小的多
3. 这个表有主键，因此不存在所有字段都为空的记录，所以`COUNT(*)`只要统计所有主键的值就可以了，不需要回表读取数据
4. `SELECT COUNT(*) FROM t`，使用`sex`索引，只需要访问辅助索引的小B+树，而使用主键索引，要访问主键索引那个大B+树，明细工作量大，这就是为什么，优化器使用辅助索引的原因



5



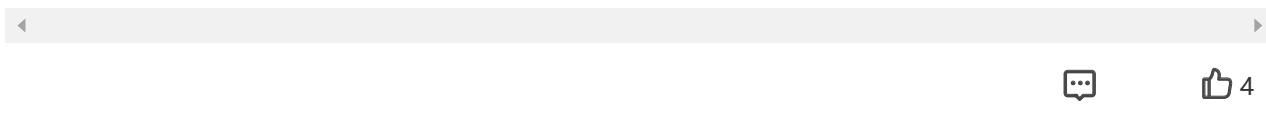
zhongsun

2021-03-09

navicat可以吗？

展开 ∨

作者回复: navicat也可以连接MySQL。新建连接->选择MySQL->编辑连接参数，端口3306，本机localhost，密码是你自己安装MySQL时候设置的root的密码。参数填写完成，就可以连接了



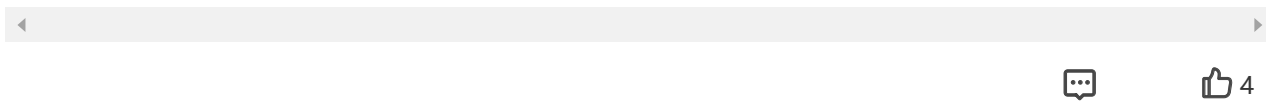
蓝色梦幻

2021-03-08

后面做项目的时候，老师选择哪个语言？

展开 ∨

作者回复: 这个要看是什么场景，比如，web端的应用，我一般是用Java来处理与MySQL的数据交互。主要是Java功能比较强大，我也比较熟。如果是PC端的应用，我比较喜欢用VC++。



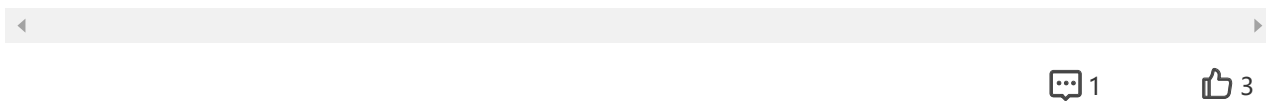
行与修

2021-03-11

```
CREATE TABLE demo.goodSales
(  
id int PRIMARY KEY AUTO_INCREMENT,  
goodId int, ...
```

展开 ∨

作者回复: 如果不需要合并数据，可以用自增字段做主键。加上商品名称，我是赞成的，这样的冗余有好处



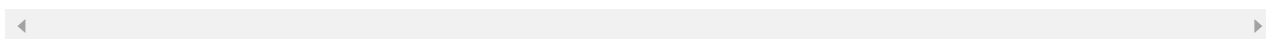
石头

2021-03-10

这是入门课吗，后面会讲得深入一些吗

展开 ∨

作者回复: 是从零开始的入门课，随着课程的展开，我会遵循由浅入深，循序渐进的原则进行讲解。后面肯定会不断深入。



**Kansei**

2021-03-10

老师您好，我工作上使用的是SQL Server 在高峰期大量读写的时候经常会很慢，希望老师在讲优化的时候，尽量多讲一些日常我们遇到类似这样的情况的时候，我们应该如何处理

作者回复: 我会从SQL语句设计，表的设计，以及服务器调优等方面进行阐述，SQLserver也是关系型数据库，很多方面的思路是一样的。相信会对你的工作，有所帮助。



2

Paul _Chan

2021-03-19

太详细了

展开 ∨



1

kingtiwns

2021-03-11

看了sql规范，说表格名不要用tbl开头，那要怎么写，我们都是用t表示table，v表示视图

作者回复: 规范的目的是为了提高可读性。SQL规范只是可读性好的书写方式的一种示例，并不是说一定要按照规范来书写，如果你们单位有自己统一的书写标准，大家都遵循，而且有连续性，我觉得完全可以按照你们自己的规范来写。



1

右耳朵猫咪

2021-03-09

老师，面试的时候被问到mysql的原理是什么？我该从哪几个方面回答呢？

作者回复: 这个问题有点大，我理解是不是问的“MySQL的工作原理”。那么可以从客户端，服务器两个方面阐述。包括客户端为各种开发环境提供的数据库接口，服务器端的链接层、服务层和存储层。这部分内容，我在直播中有所阐述，你也可以通过极客时间的哔站找到直播相关的PPT，作为参考。



1

武明

2021-03-25

老师好，数据库名称是不是不能带小数点 “.” 啊，已经带了点的数据库怎么操作 DESCRIBE 命令，我用了括号、引号试试都不行

展开 ∨

**Initiative Thinker**

2021-03-15

为什么设置goodsname字段的数据类型为varchar会出错呢？

作者回复: 应该是你没有指定varchar类型的最大长度。MySQL要求指定varchar类型的时候，必须要指定最大长度。否则没法分配存储空间

**袋袋**

2021-03-12

```
create table goodsname; -- 新建一个表goodsname
insert into goodsname
(number,barcode,goodsname,price)
values('0001','00000101','本子',3),('0002','00000103','铅笔',1.0),('0003','00000105','钢笔',10),...
```

展开 ∨

作者回复: 对的

