**数据库开发-MySQL**

**前言**

在我们讲解SpringBootWeb基础知识(请求响应案例)的时候，我们讲到在web开发中，为了应用程序 职责单一，方便维护，我们一般将web应用程序分为三层，即：Controller、Service、Dao 。 之前我们的案例中，是这样子的请求流程：浏览器发起请求，先请求Controller；Controller接收 到请求之后，调用Service进行业务逻辑处理；Service再调用Dao，Dao再解析user.xml中所存储 的数据。



xml文件中可以存储数据，但是在企业项目开发中不会使用xml文件存储数据，因为不便管理维护，操作 难度大。在真实的企业开发中呢，都会采用数据库来存储和管理数据，那此时，web开发调用流程图如 下所示：



首先来了解一下什么是数据库。

数据库：英文为DataBase，简称DB，它是存储和管理数据的仓库。

像我们日常访问的电商网站京东，企业内部的管理系统OA、ERP、CRM这类的系统，以及大家每天都会 刷的头条、抖音类的app，那这些大家所看到的数据，其实都是存储在数据库中的。最终这些数据，只 是在浏览器或app中展示出来而已，最终数据的存储和管理都是数据库负责的。



数据是存储在数据库中的，那我们要如何来操作数据库以及数据库中所存放的数据呢？

那这里呢，会涉及到一个软件：数据库管理系统（**D**ata**B**ase **M**anagement **S**ystem，简称DBMS）

DBMS是操作和管理数据库的大型软件。将来我们只需要操作这个软件，就可以通过这个软件来操 纵和管理数据库了。

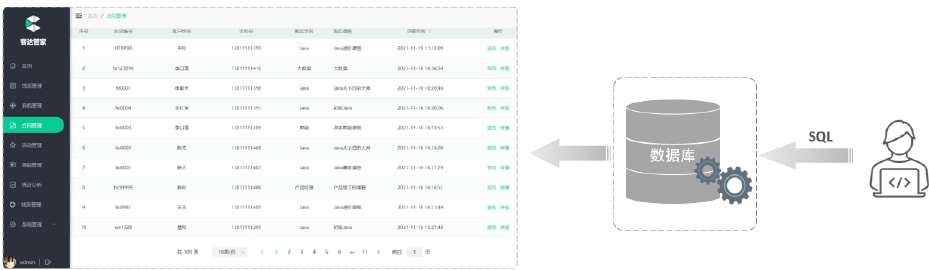
此时又出现一个问题：DBMS这个软件怎么知道要操作的是哪个数据库、哪个数据呢？是对数据做修改还 是查询呢？

需要给DBMS软件发送一条指令，告诉这个软件我们要执行的是什么样的操作，要对哪个数据进行 操作。而这个指令就是SQL语句

SQL（**S**tructured **Q**uery **L**anguage，简称SQL）：结构化查询语言，它是操作关系型数据库的编 程语言，定义了一套操作关系型数据库的统一标准。我们学习数据库开发，最为重要的就是学习SQL语 句。

关系型数据库：我们后面会详细讲解，现在大家只需要知道我们学习的数据库属于关系型数据库即 可。





结论：程序员给数据库管理系统(DBMS)发送SQL语句，再由数据库管理系统操作数据库当中的数据。

了解了数据库的一些简单概念之后，接下来我们再来介绍下目前主流的数据库，这里截取了排名前十的 数据库：



Oracle：大型的收费数据库，Oracle公司产品，价格昂贵。（通常是不差钱的公司会选择使用这 个数据库）

MySQL：开源免费的中小型数据库，后来Sun公司收购了MySQL，而Oracle又收购了Sun公司。目 前Oracle推出两个版本的Mysql：社区版(开源免费)、商业版(收费)。

SQL Server：Microsoft 公司推出的收费的中型数据库，C#、.net等语言常用。 PostgreSQL：开源免费的中小型数据库。

DB2：IBM公司的大型收费数据库产品。

SQLLite：嵌入式的微型数据库。Android内置的数据库采用的就是该数据库。 MariaDB：开源免费的中小型数据库。是MySQL数据库的另外一个分支、另外一个衍生产品，与 MySQL数据库有很好的兼容性。

那这么多数据库，我们全部都需要学习吗，其实并不用，我们只需要学习其中的一个就可以了，我们此 次课程中学习的数据库是现在互联网公司开发使用最为流行的MySQL数据库。

此时大家可能会有一个疑问，我们现在学习的是Mysql数据库，我们以后去公司做开发，如果用到的是 Oracle数据库或SQL Server数据库该怎么办？其实大家完全不用担心这个问题，因为这些数据库都 是属于关系型数据库，要操作关系型数据库都是通过SQL语句来实现的，而SQL语句又是操作关系型数 据库的统一标准。

结论：只要我们学会了SQL语句，就可以通过SQL语句来操作Mysql，也可以通过SQL语句来操作 Oracle或SQL Server



课程内容安排：



以上课程内容拆解为3部分知识点：



接下来我们就先进入到第一天的Mysql概述学习。

**1. MySQL概述**



官网：**https://dev.mysql.com/**

**1.1 安装**

**1.1.1 版本**

MySQL官方提供了两个版本：

商业版本（MySQL Enterprise Edition）

该版本是收费的，我们可以使用30天。 官方会提供对应的技术支持。 社区版本（MySQL Community Server）

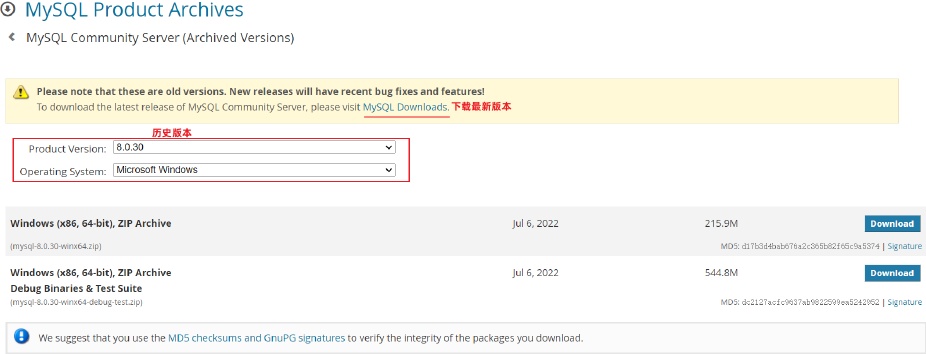
该版本是免费的，但是MySQL不会提供任何的技术支持。

本课程，采用的是MySQL的社区版本（8.0.31）



**1.1.2 安装**

官网下载地址：**https://downloads.mysql.com/archives/community/**



这个安装包呢，大家可以不用下载，因为资料中已提供现成的安装包。

IMAGE



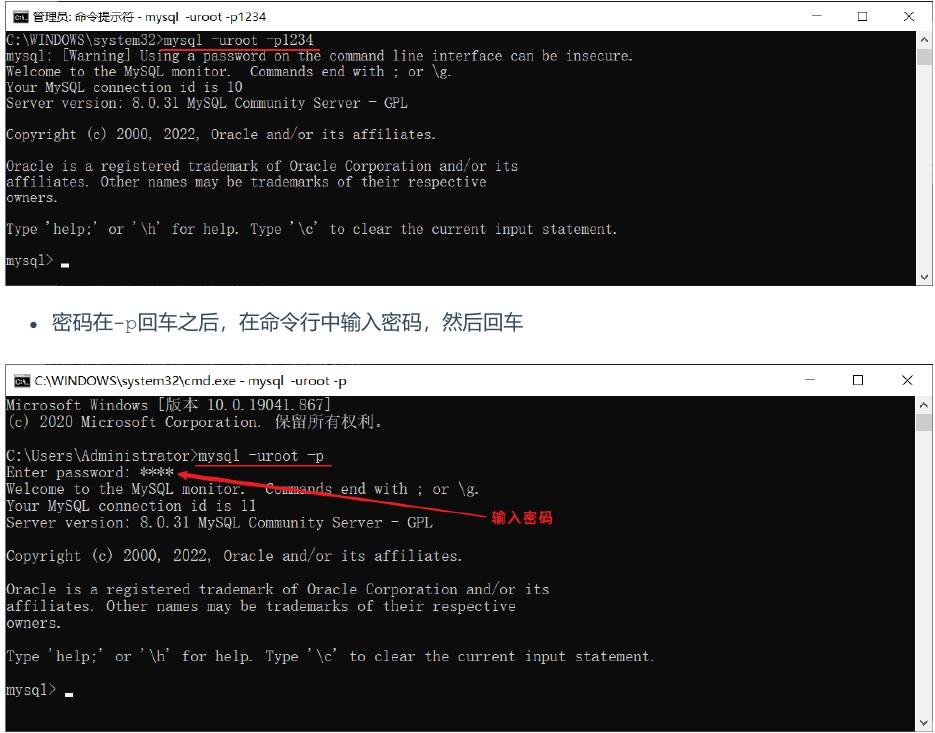
安装MySQL具体的步骤，参考资料中提供的 《MySQL安装文档》 ，进行软件的安装即可。

**1.1.3 连接**

MySQL服务器启动完毕后，然后再使用如下指令，来连接MySQL服务器：

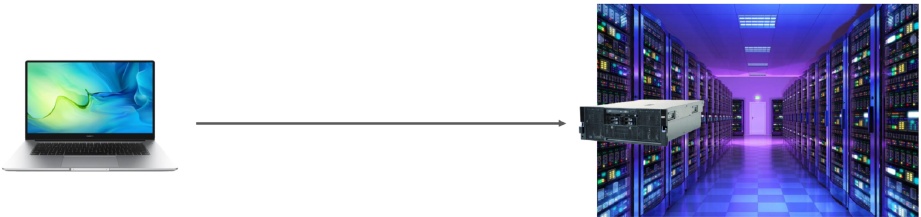
**上述指令，可以有两种形式：**

密码直接在-p参数之后直接指定 （这种方式不安全，密码直接以明文形式出现在命令行）



**1.1.4 企业使用方式(了解)**

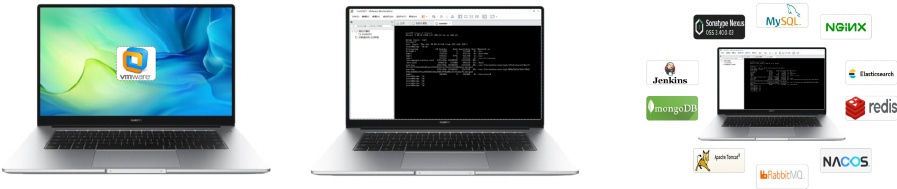
上述的MySQL服务器我们是安装在本地的，这个仅仅是在我们学习阶段，在真实的企业开发中，MySQL 数据库服务器是不会在我们本地安装的，是在公司的服务器上安装的，而服务器还需要放置在专门的 IDC机房中的，IDC机房呢，就需要保证恒温、恒湿、恒压，而且还要保证网络、电源的可靠性(备用电 源及网络)。



那我们要想使用服务器上的这台MySQL服务器，就需要在我们的电脑上去远程连接这台MySQL。 而服 务器上安装的MySQL数据库呢，并不是你一个人在访问，我们项目组的其他开发人员也是需要访问这台 MySQL的。



我们在学习阶段，如果想提前体验一下企业中数据库的使用方式，可以借助于VMware虚拟机来实现。我 们可以在我们的电脑上安装一个VMware，然后在Vmware虚拟机上在安装一个服务器操作系统Linux， 然后再在Linux服务器上安装各种企业级软件。



接下来，就来演示一下，通过MySQL的客户端命令行，如何来连接服务器上部署的MySQL ：

**1.2 数据模型**

介绍完了Mysql数据库的安装配置之后，接下来我们再来聊一聊Mysql当中的数据模型。学完了这一小 节之后，我们就能够知道在Mysql数据库当中到底是如何来存储和管理数据的。

在介绍Mysql的数据模型之前，需要先了解一个概念：关系型数据库。

**（RDBMS）**

**关系型数据库**

概念：建立在关系模型基础上，由多张相互连接的**二维表**组成的数据库。 而所谓二维表，指的是由行和列组成的表，如下图：



二维表的优点：

使用表存储数据，格式统一，便于维护

使用SQL语言操作，标准统一，使用方便，可用于复杂查询

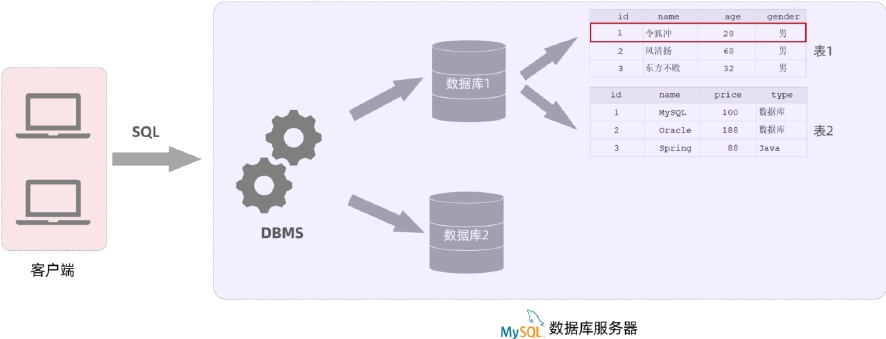
我们之前提到的MySQL、Oracle、DB2、SQLServer这些都是属于关系型数据库，里面都是基于 二维表存储数据的。

结论：基于二维表存储数据的数据库就成为关系型数据库，不是基于二维表存储数据的数据库，就 是非关系型数据库（比如大家后面要学习的Redis，就属于非关系型数据库）。

**2). 数据模型**

介绍完了关系型数据库之后，接下来我们再来看一看在Mysql数据库当中到底是如何来存储数据的，也 就是Mysql 的数据模型。

MySQL是关系型数据库，是基于二维表进行数据存储的，具体的结构图下:



通过MySQL客户端连接数据库管理系统DBMS，然后通过DBMS操作数据库

使用MySQL客户端，向数据库管理系统发送一条SQL语句，由数据库管理系统根据SQL语句指令去 操作数据库中的表结构及数据

一个数据库服务器中可以创建多个数据库，一个数据库中也可以包含多张表，而一张表中又可以包 含多行记录。

在Mysql数据库服务器当中存储数据，你需要：

1. 先去创建数据库（可以创建多个数据库，之间是相互独立的）

2. 在数据库下再去创建数据表（一个数据库下可以创建多张表）

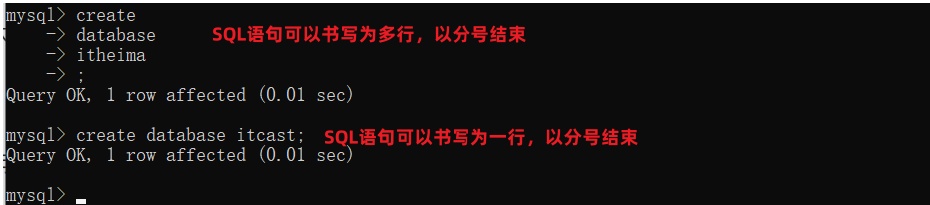
3. 再将数据存放在数据表中（一张表可以存储多行数据）

**1.3 SQL简介**

SQL：结构化查询语言。一门操作关系型数据库的编程语言，定义操作所有关系型数据库的统一标准。 在学习具体的SQL语句之前，先来了解一下SQL语言的语法。

**1.3.1 SQL通用语法**

1、SQL语句可以单行或多行书写，以分号结尾。



2、SQL语句可以使用空格/缩进来增强语句的可读性。



3、MySQL数据库的SQL语句不区分大小写。

4、注释：

单行注释：-- 注释内容 或 # 注释内容(MySQL特有) 多行注释：/\* 注释内容\*/

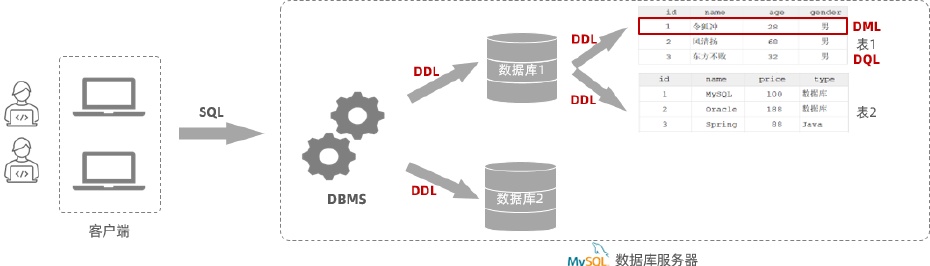
以上就是SQL语句的通用语法，这些通用语法大家目前先有一个直观的认识，我们后面在讲解每一 类SQL语句的时候，还会再来强调通用语法。



**1.3.2 分类**

SQL语句根据其功能被分为四大类：DDL、DML、DQL、DCL

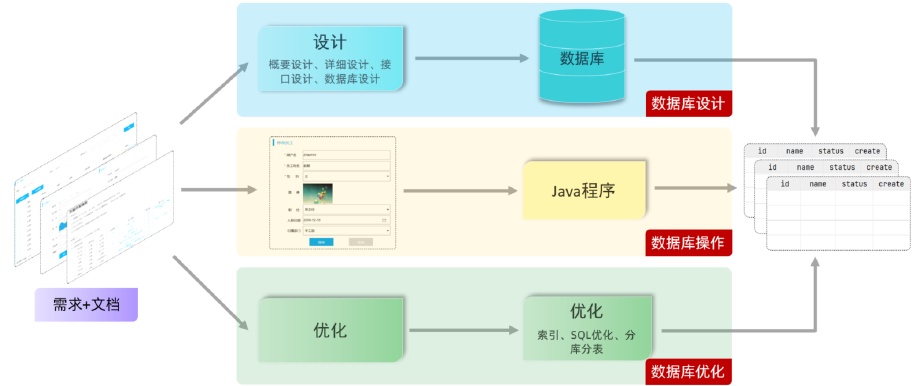




**2. 数据库设计-DDL**

下面我们就正式的进入到SQL语句的学习，在学习之前先给大家介绍一下我们要开发一个项目，整个开 发流程是什么样的，以及在流程当中哪些环节会涉及到数据库。

**2.1 项目开发流程**



需求文档：

在我们开发一个项目或者项目当中的某个模块之前，会先会拿到产品经理给我们提供的页面原型及 需求文档。



设计：

拿到产品原型和需求文档之后，我们首先要做的不是编码，而是要先进行项目的设计，其中就包括 概要设计、详细设计、接口设计、数据库设计等等。

数据库设计根据产品原型以及需求文档，要分析各个模块涉及到的表结构以及表结构之间的关系， 以及表结构的详细信息。最终我们需要将数据库以及数据库当中的表结构设计创建出来。

开发/测试：

参照页面原型和需求进行编码，实现业务功能。在这个过程当中，我们就需要来操作设计出来的数 据库表结构，来完成业务的增删改查操作等。

部署上线：

在项目的功能开发测试完成之后，项目就可以上线运行了，后期如果项目遇到性能瓶颈，还需要对 项目进行优化。优化很重要的一个部分就是数据库的优化，包括数据库当中索引的建立、SQL 的

优化、分库分表等操作。

在上述的流程当中，针对于数据库来说，主要包括三个阶段：

1. 数据库设计阶段

参照页面原型以及需求文档设计数据库表结构 数据库操作阶段

2.

根据业务功能的实现，编写SQL语句对数据表中的数据进行增删改查操作

3. 数据库优化阶段

通过数据库的优化来提高数据库的访问性能。优化手段：索引、SQL优化、分库分表等 接下来我们就先来学习第一部分数据库的设计，而数据库的设计就是来定义数据库，定义表结构以及表 中的字段。

**2.2 数据库操作**

我们在进行数据库设计，需要使用到刚才所介绍SQL分类中的DDL语句。

DDL英文全称是Data Definition Language(数据定义语言)，用来定义数据库对象(数据库、 表)。

DDL中数据库的常见操作：查询、创建、使用、删除。

**2.2.1 查询数据库**

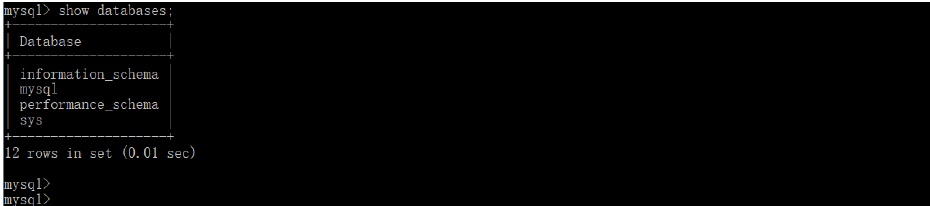
**查询所有数据库：**



1

show databases;

命令行中执行效果如下：



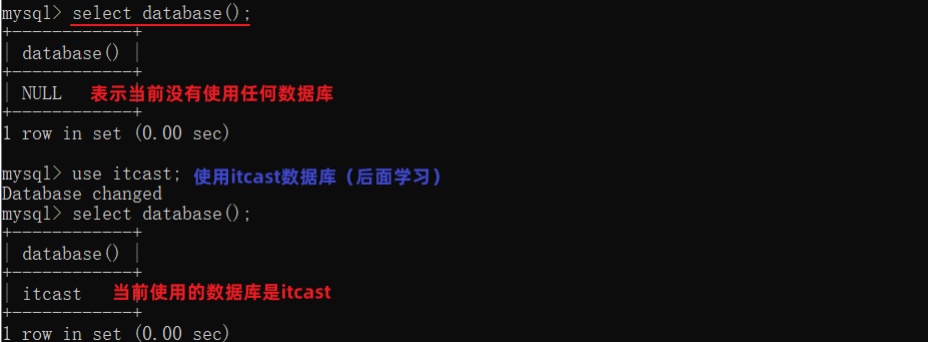
**查询当前数据库：**



1

select database();

命令行中执行效果如果：



我们要操作某一个数据库，必须要切换到对应的数据库中。

通过指令：select database() ，就可以查询到当前所处的数据库

**2.2.2 创建数据库**

**语法：**

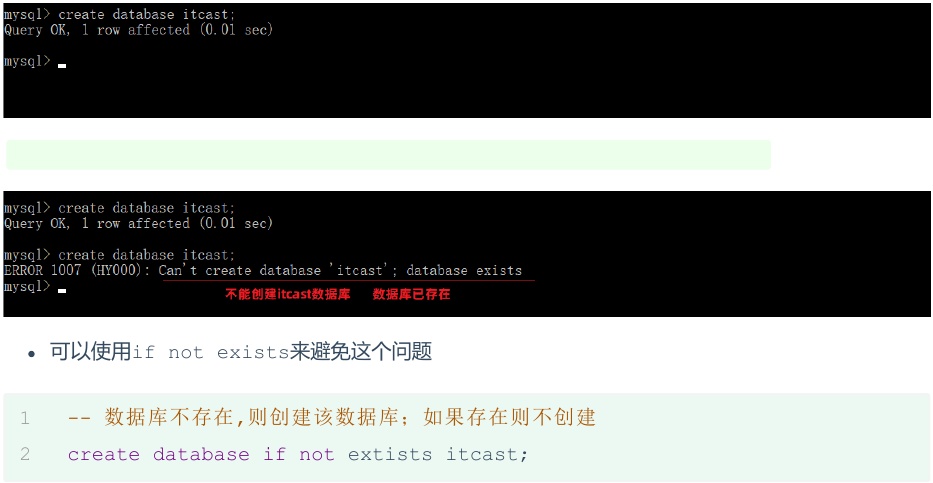


1

create database [ if not exists ] 数据库名;

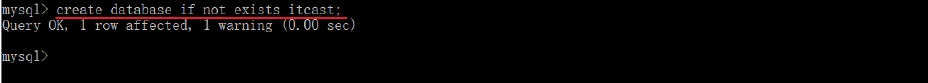
案例：创建一个itcast数据库。

命令行执行效果如下：



注意：在同一个数据库服务器中，不能创建两个名称相同的数据库，否则将会报错。

命令行执行效果如下：



**2.2.3 使用数据库**

**语法：**

use 数据库名;

1



我们要操作某一个数据库下的表时，就需要通过该指令，切换到对应的数据库下，否则不能操作。

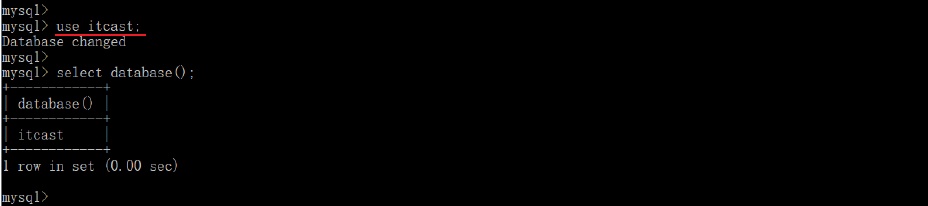
案例：切换到itcast数据



1

use itcast;

命令执行效果如下：



**2.2.4 删除数据库**

**语法：**

1

drop database [ if exists ] 数据库名;



如果删除一个不存在的数据库，将会报错。

可以加上参数if exists ，如果数据库存在，再执行删除，否则不执行删除。

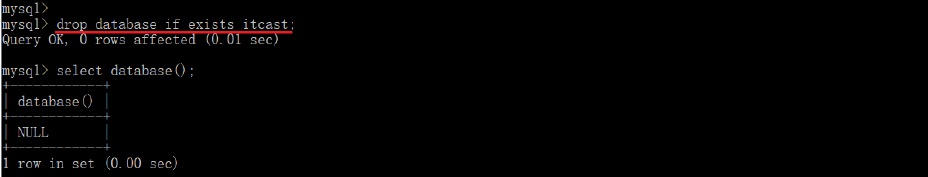
案例：删除itcast数据库



1

drop database if exists itcast; -- itcast数据库存在时删除

命令执行效果如下：



说明：上述语法中的database，也可以替换成 schema

如：create schema db01; 如：show schemas;



**2.3 图形化工具**

**2.3.1 介绍**

前面我们讲解了DDL中关于数据库操作的SQL语句，在我们编写这些SQL时，都是在命令行当中完成的。 大家在练习的时候应该也感受到了，在命令行当中来敲这些SQL语句很不方便，主要的原因有以下 3 点：

1. 没有任何代码提示。（全靠记忆，容易敲错字母造成执行报错）

2. 操作繁琐，影响开发效率。（所有的功能操作都是通过SQL语句来完成的）

3. 编写过的SQL代码无法保存。

在项目开发当中，通常为了提高开发效率，都会借助于现成的图形化管理工具来操作数据库。 目前MySQL主流的图形化界面工具有以下几种：



DataGrip是JetBrains旗下的一款数据库管理工具，是管理和开发MySQL、Oracle、PostgreSQL 的理想解决方案。

官网：**https://www.jetbrains.com/zh-cn/datagrip/**

**2.3.2 安装**

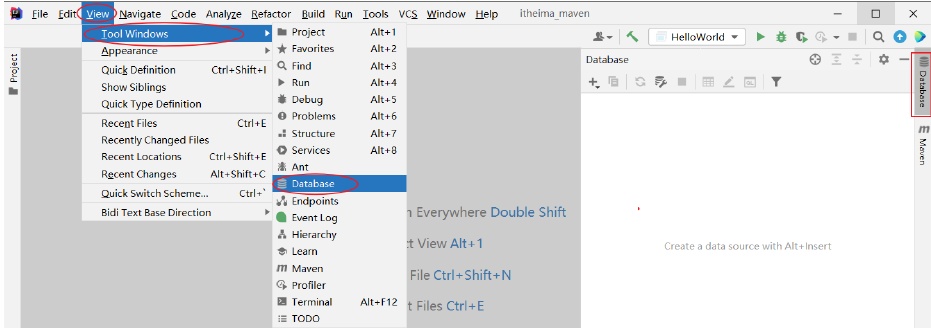
安装：参考资料中提供的《DataGrip安装手册》

说明：DataGrip这款工具可以不用安装，因为Jetbrains公司已经将DataGrip这款工具的功 能已经集成到了IDEA当中，所以我们就可以使用IDEA来作为一款图形化界面工具来操作Mysql 数据库。

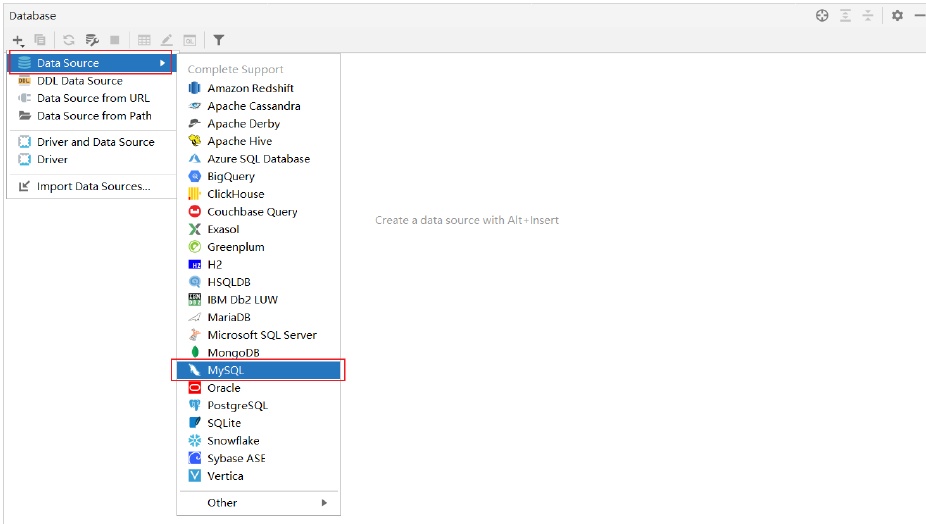
**2.3.3 使用**

**2.2.3.1 连接数据库**

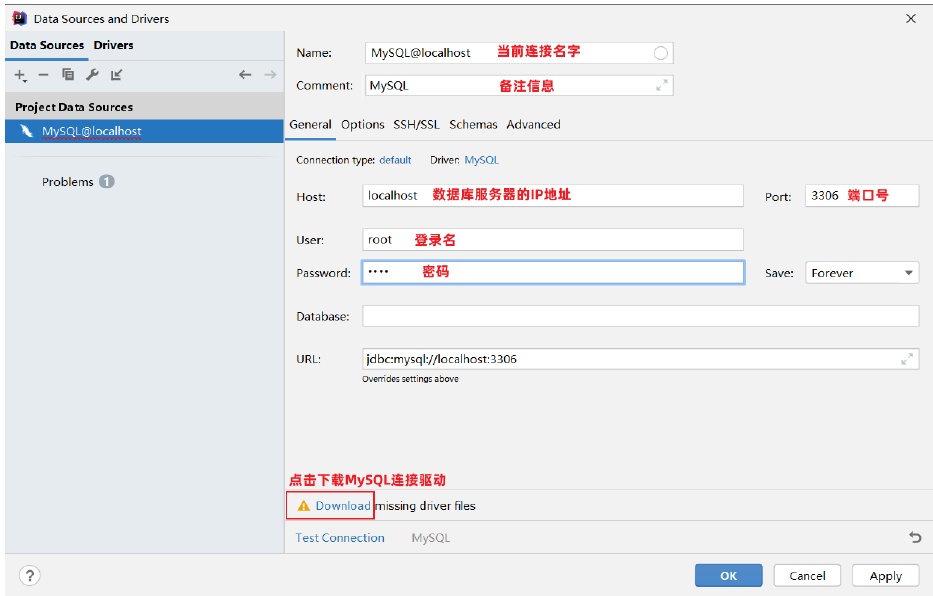
1、打开IDEA自带的Database



2、配置MySQL



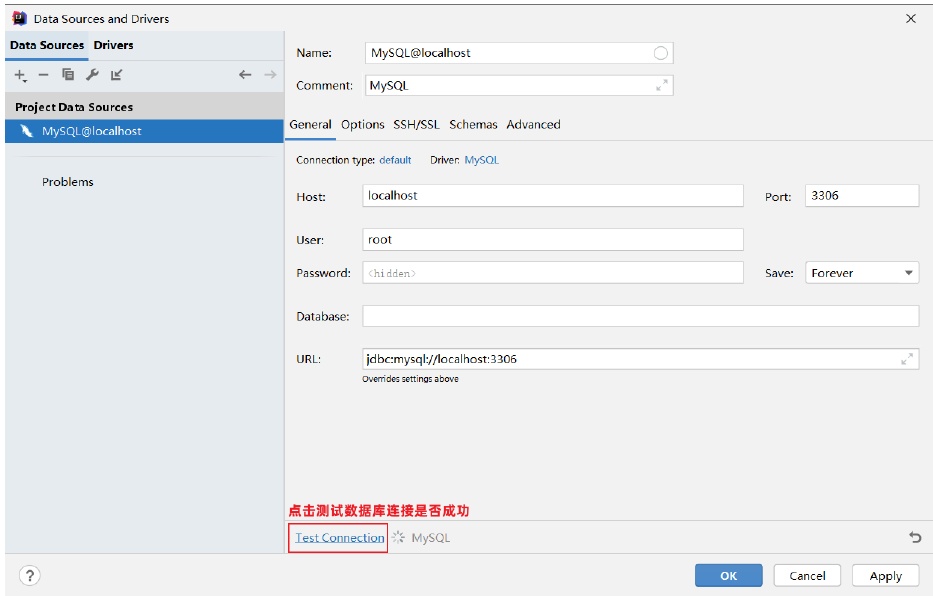
3、输入相关信息



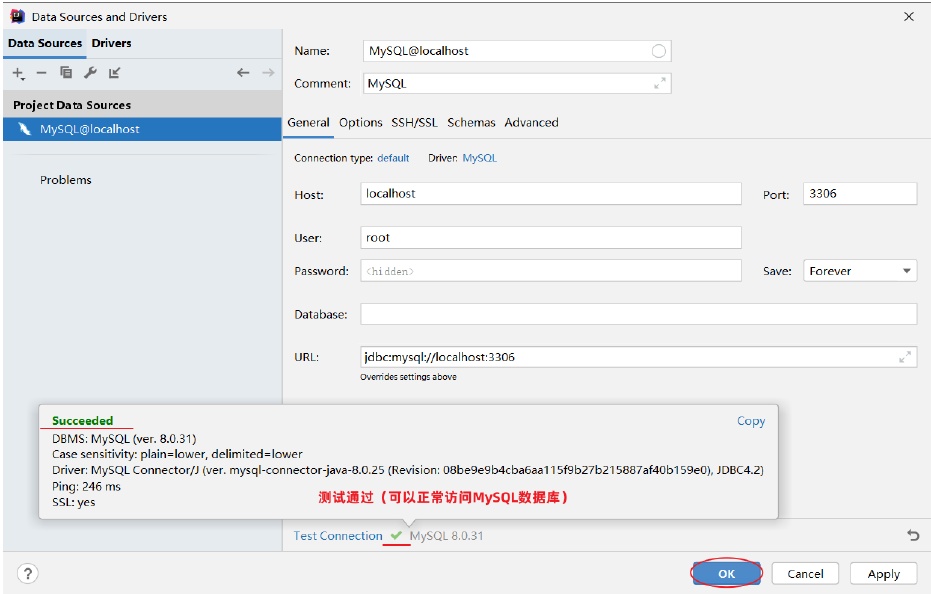
4、下载MySQL连接驱动



5、测试数据库连接



6、保存配置

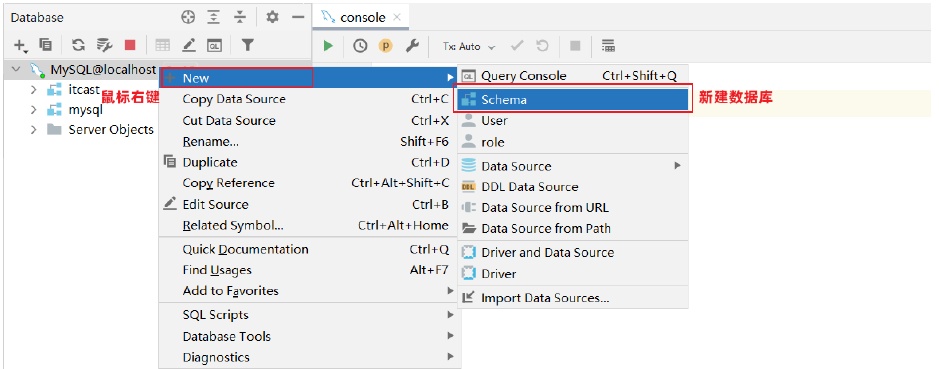


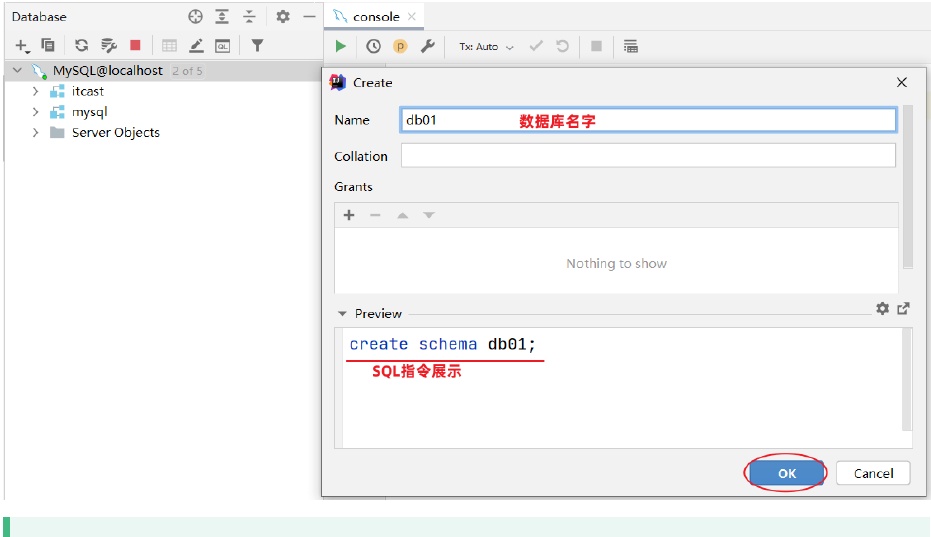
默认情况下，连接上了MySQL数据库之后， 数据库并没有全部展示出来。需要选择要展示哪些数据 库。具体操作如下：



**2.2.3.2 操作数据库**

**创建数据库：**

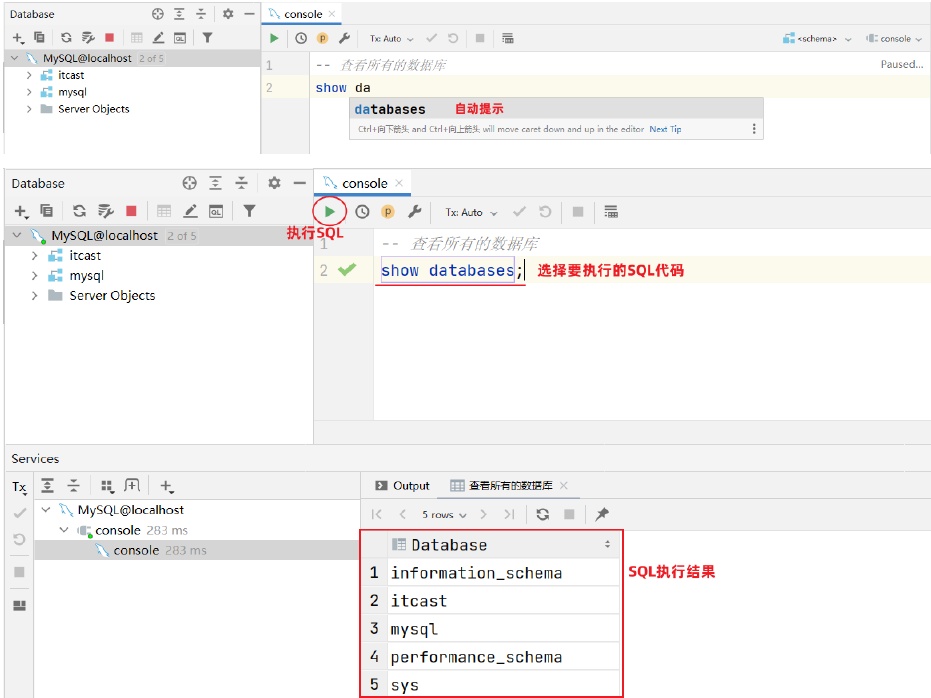




有了图形化界面工具后，就可以方便的使用图形化工具：创建数据库，创建表、修改表等DDL操 作。

其实工具底层也是通过DDL语句操作的数据库，只不过这些SQL语句是图形化界面工具帮我们自动 完成的。

**查看所有数据库：**



**2.3 表操作**

学习完了DDL语句当中关于数据库的操作之后，接下来我们继续学习DDL语句当中关于表结构的操作。 关于表结构的操作也是包含四个部分：创建表、查询表、修改表、删除表。

**2.3.1 创建**

**2.3.1.1 语法**



案例：创建tb\_user表

对应的结构如下：

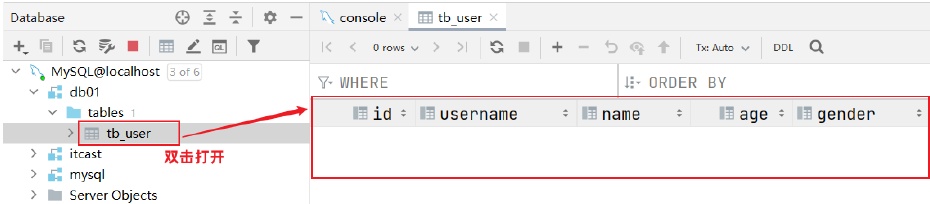


建表语句：

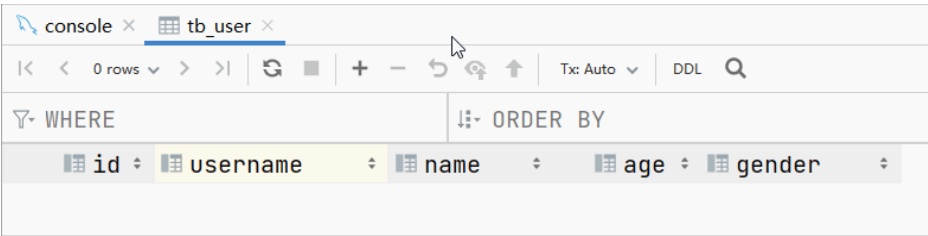
数据表创建完成，接下来我们还需要测试一下是否可以往这张表结构当中来存储数据。



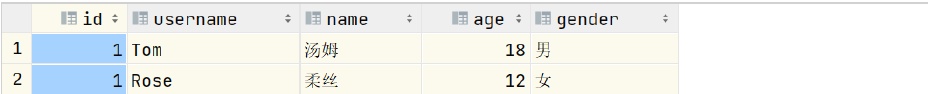
双击打开tb\_user表结构，大家会发现里面没有数据：



添加数据：



此时我们再插入一条数据：



我们之前提到过：id字段是一行数据的唯一标识，不能有重复值。但是现在数据表中有两个相同的id 值，这是为什么呢？

其实我们现在创建表结构的时候，id这个字段我们只加了一个备注信息说明它是一个唯一标识， 但是在数据库层面呢，并没有去限制字段存储的数据。所以id这个字段没有起到唯一标识的作用。

想要限制字段所存储的数据，就需要用到数据库中的约束。



**2.3.1.2 约束**

概念：所谓约束就是作用在表中字段上的规则，用于限制存储在表中的数据。

作用：就是来保证数据库当中数据的正确性、有效性和完整性。（后面的学习会验证这些） 在MySQL数据库当中，提供了以下5种约束：



注意：约束是作用于表中字段上的，可以在创建表/修改表的时候添加约束。



案例：创建tb\_user表

对应的结构如下：



在上述的表结构中:

id 是一行数据的唯一标识 username 用户名字段是非空且唯一的 name 姓名字段是不允许存储空值的 gender 性别字段是有默认值，默认为男

建表语句：



1 create table tb\_user (

2 id int primary key comment 'ID,唯一标识',

3 username varchar(20) not null unique comment '用户名',

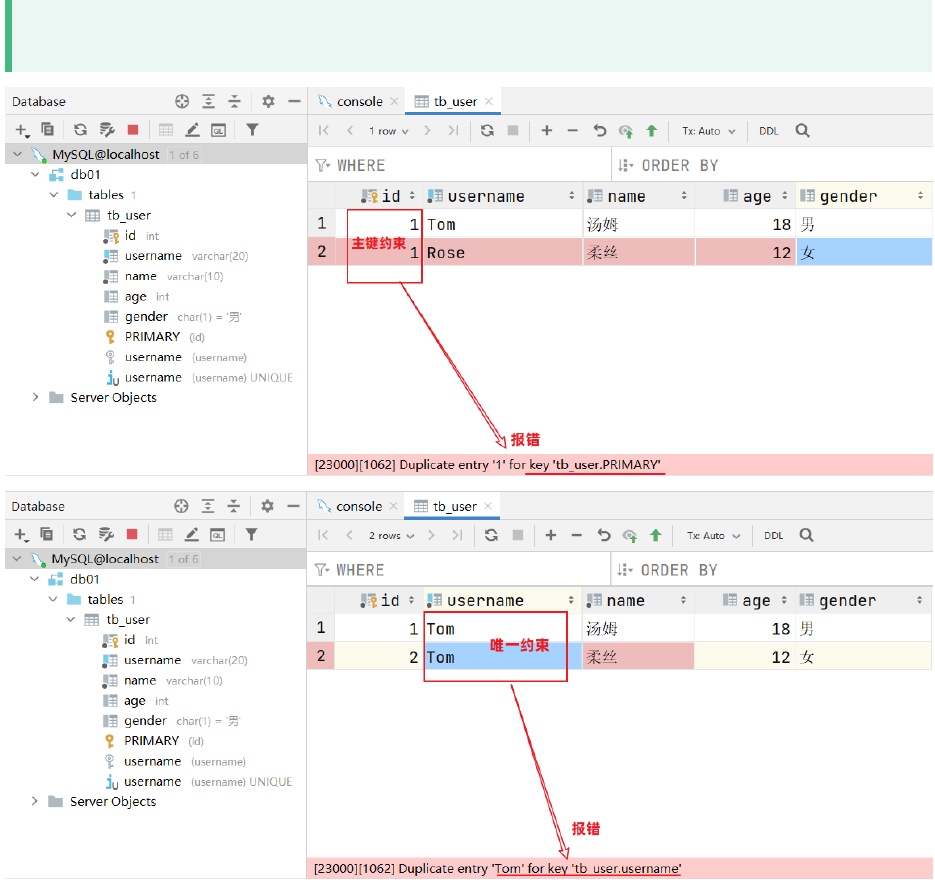
4 name varchar(10) not null comment '姓名',

5 age int comment '年龄',

6 gender char(1) default '男' comment '性别'

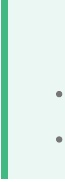
7 ) comment '用户表';

数据表创建完成，接下来测试一下表中字段上的约束是否生效



大家有没有发现一个问题：id字段下存储的值，如果由我们自己来维护会比较麻烦(必须保证值的唯一 性)。MySQL数据库为了解决这个问题，给我们提供了一个关键字：auto\_increment（自动增长）

主键自增：auto\_increment



每次插入新的行记录时，数据库自动生成id字段(主键)下的值 具有auto\_increment的数据列是一个正数序列开始增长(从1开始自增)

测试：主键自增

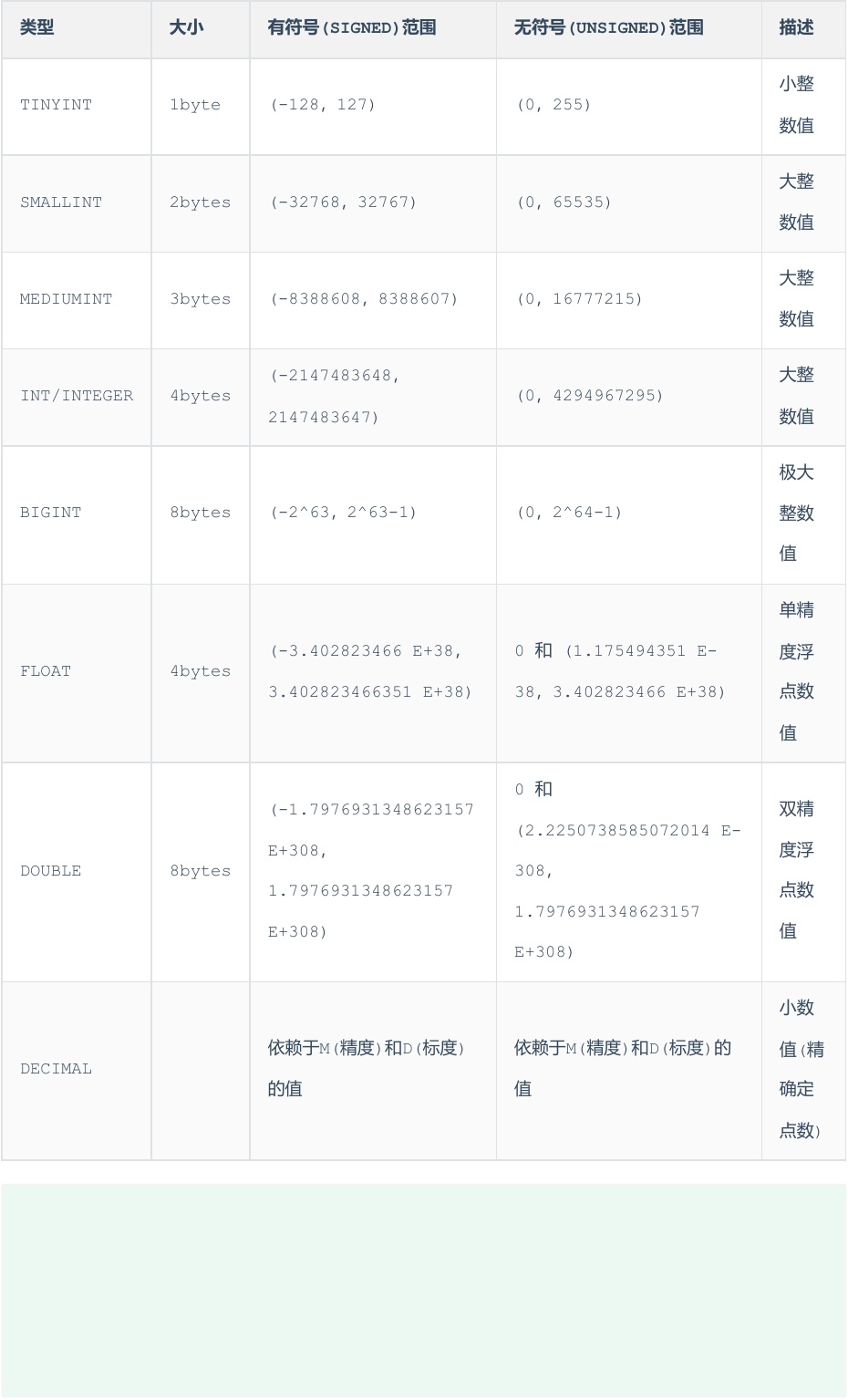


**2.3.1.3 数据类型**

在上面建表语句中，我们在指定字段的数据类型时，用到了int 、varchar、char，那么在MySQL中 除了以上的数据类型，还有哪些常见的数据类型呢？接下来,我们就来详细介绍一下MySQL的数据类 型。

MySQL中的数据类型有很多，主要分为三类：数值类型、字符串类型、日期时间类型。

**数值类型**



1

示例:

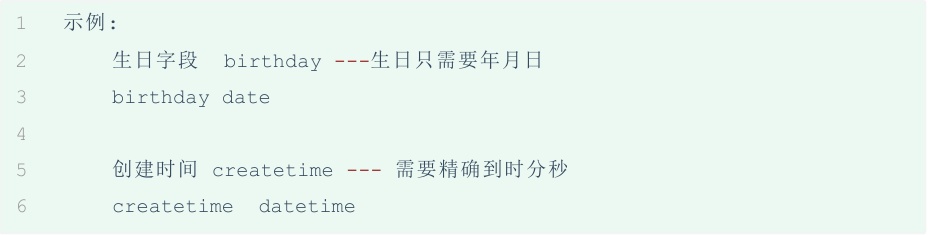
**字符串类型**



char 与varchar 都可以描述字符串，char是定长字符串，指定长度多长，就占用多少个字符，和 字段值的长度无关。而varchar是变长字符串，指定的长度为最大占用长度 。相对来说，char的性 能会更高些。

**日期时间类型**





**2.3.1.4 案例**

需求：根据产品原型/需求创建表((设计合理的数据类型、长度、约束)

参考资料中提供的《黑马-tlias智能学习辅助系统》页面原型，设计员工管理模块的表结构 暂不考虑所属部门字段



产品原型及需求如下：







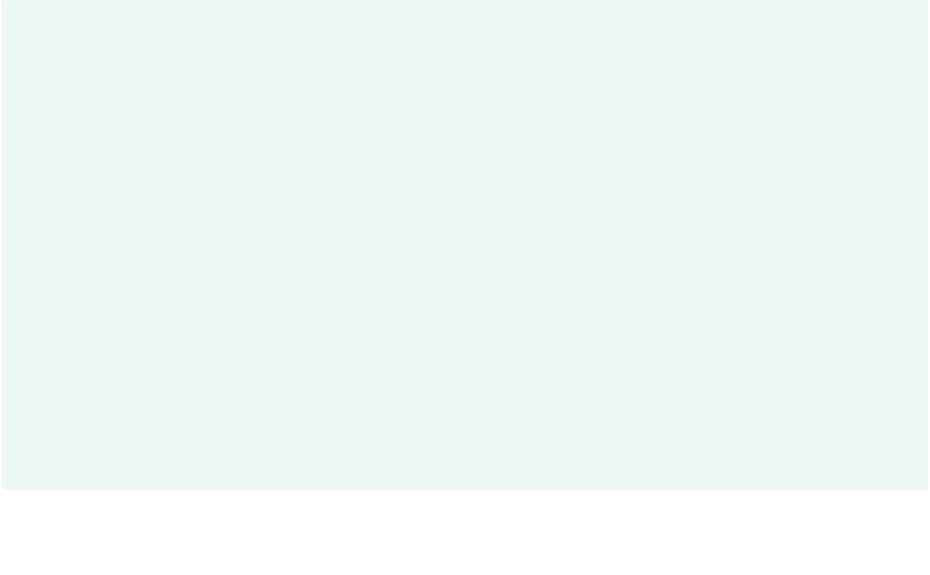
**步骤：**

1. 阅读产品原型及需求文档，看看里面涉及到哪些字段。

2. 查看需求文档说明，确认各个字段的类型以及字段存储数据的长度限制。

3. 在页面原型中描述的基础字段的基础上，再增加额外的基础字段。

使用SQL创建表：



1 create table emp (

2 id int unsigned primary key auto\_increment comment 'ID',

3 username varchar(20) not null unique comment '用户名',

4 password varchar(32) default '123456' comment '密码',

5 name varchar(10) not null comment '姓名',

6 gender tinyint unsigned not null comment '性别, 说明: 1 男, 2 女',

7 image varchar(300) comment '图像',

8 job tinyint unsigned comment '职位, 说明: 1 班主任,2 讲师, 3 学工主  
管, 4 教研主管',

9 entrydate date comment '入职时间',

10 create\_time datetime not null comment '创建时间',

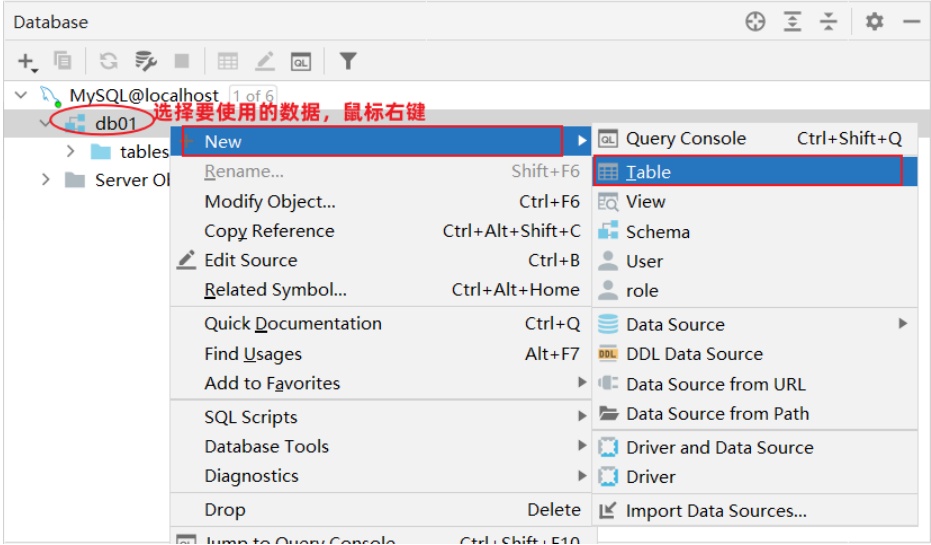
11 update\_time datetime not null comment '修改时间'

12 ) comment '员工表';

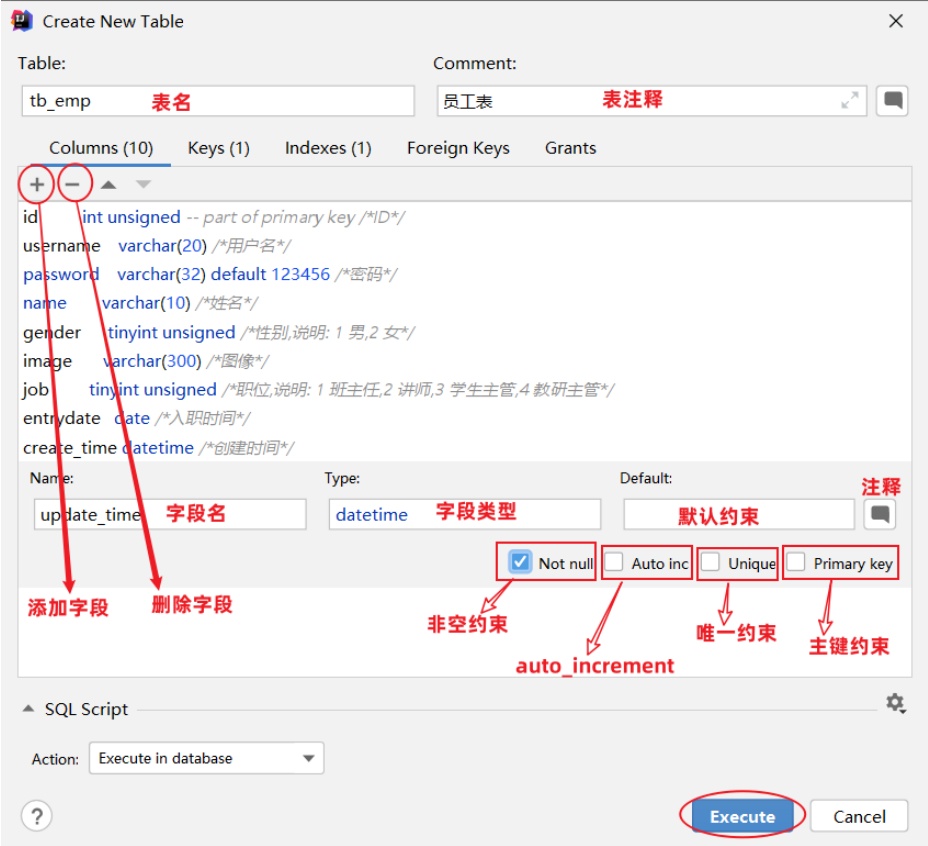
除了使用SQL语句创建表外，我们还可以借助于图形化界面来创建表结构，这种创建方式会更加直观、   
更加方便。

操作步骤如下：

1. 在指定操作的数据库上，右键New ==> Table



2. 编辑表结构的相关信息



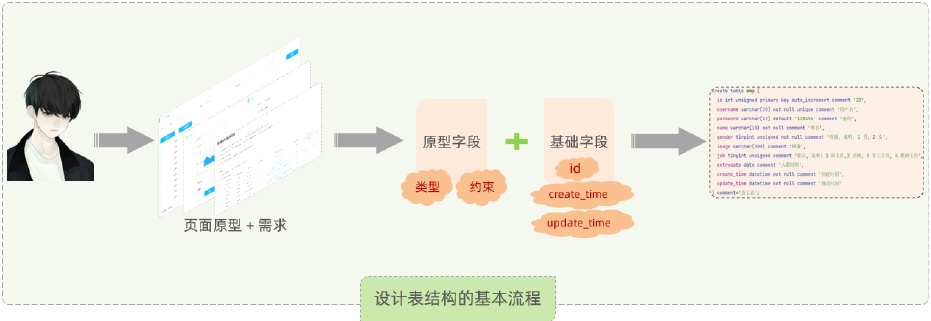
**2.3.1.5 设计表流程**

通过上面的案例，我们明白了，设计一张表，基本的流程如下：

1. 阅读页面原型及需求文档

2. 基于页面原则和需求文档，确定原型字段(类型、长度限制、约束)

3. 再增加表设计所需要的业务基础字段(id主键、插入时间、修改时间)



说明：

create\_time：记录的是当前这条数据插入的时间。 update\_time：记录当前这条数据最后更新的时间。

**2.3.2 查询**

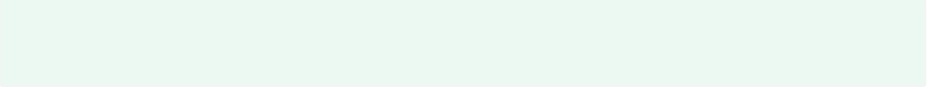
关于表结构的查询操作，工作中一般都是直接基于**图形化界面操作**。



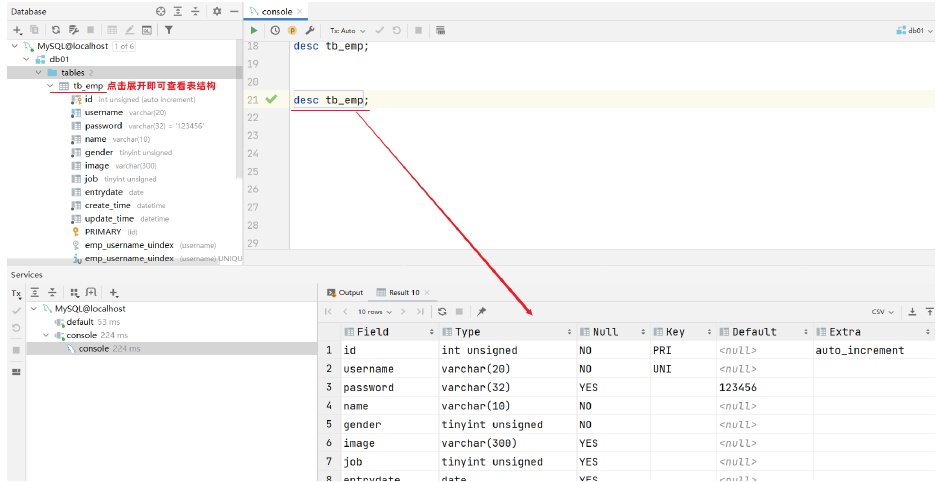
**查询当前数据库所有表**



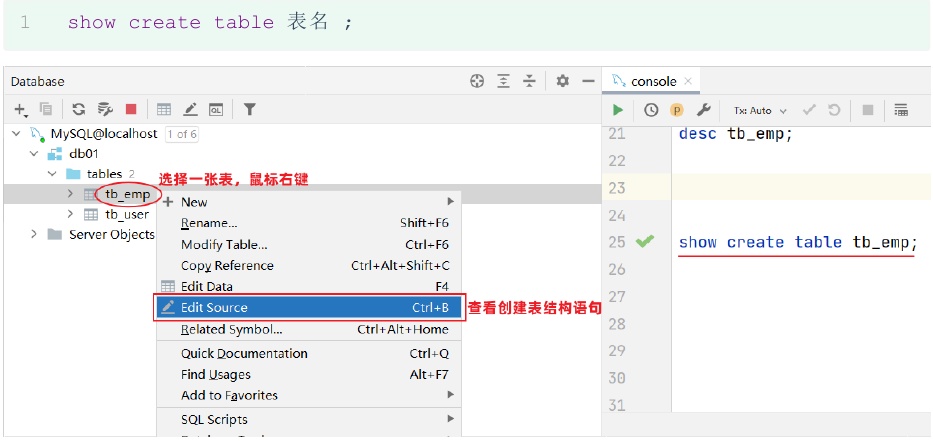
**查看指定表结构**



1 desc 表名;#可以查看指定表的字段、字段的类型、是否可以为NULL、是否存在默认值   
等信息



查询指定表的建表语句





2.3.3修改

关于表结构的修改操作，工作中一般都是直接基于图形化界面操作。

添加字段

alter table表名add 字段名 类型(长度) [comment注释] [约束];

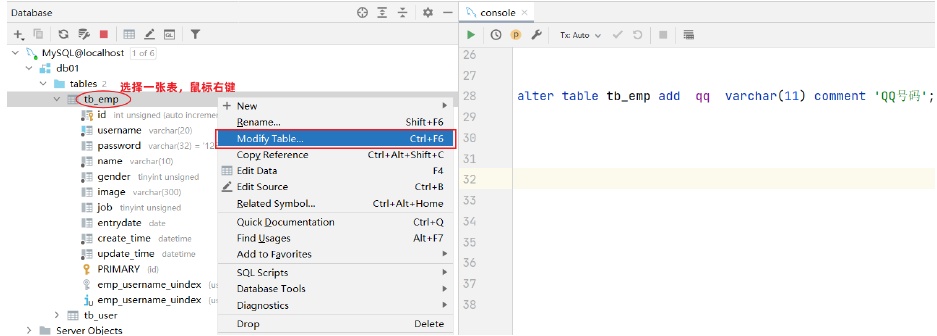
1

案例: 为tbemp表添加字段qq,字段类型为varchar (11)

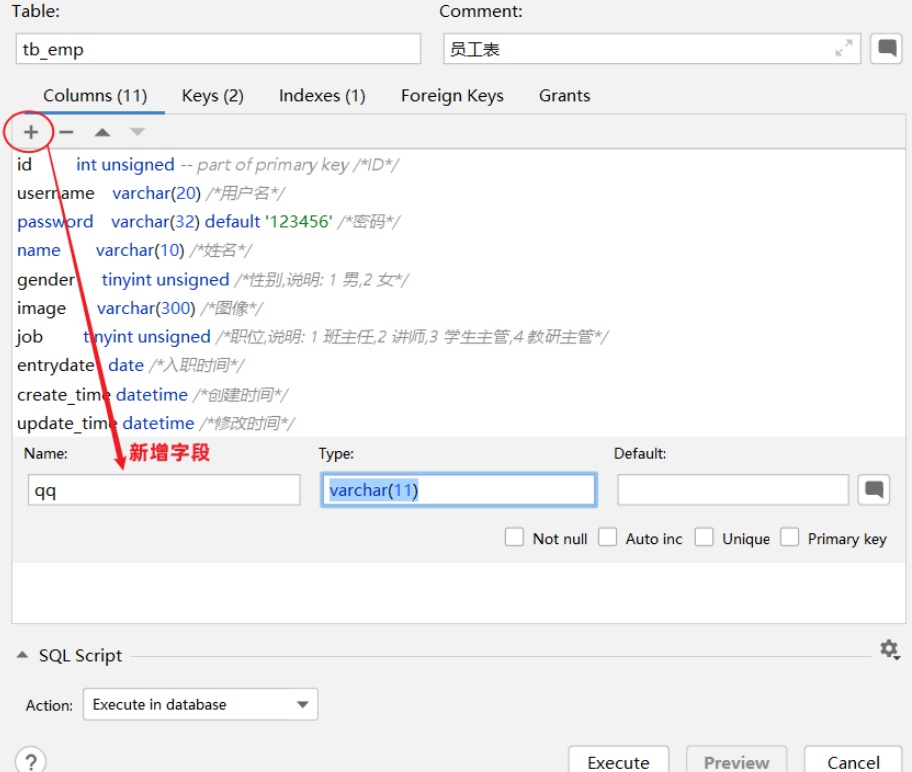
1

alter table tb\_emp add qq varchar (11) comment 0Q号码';

图形化操作：添加字段

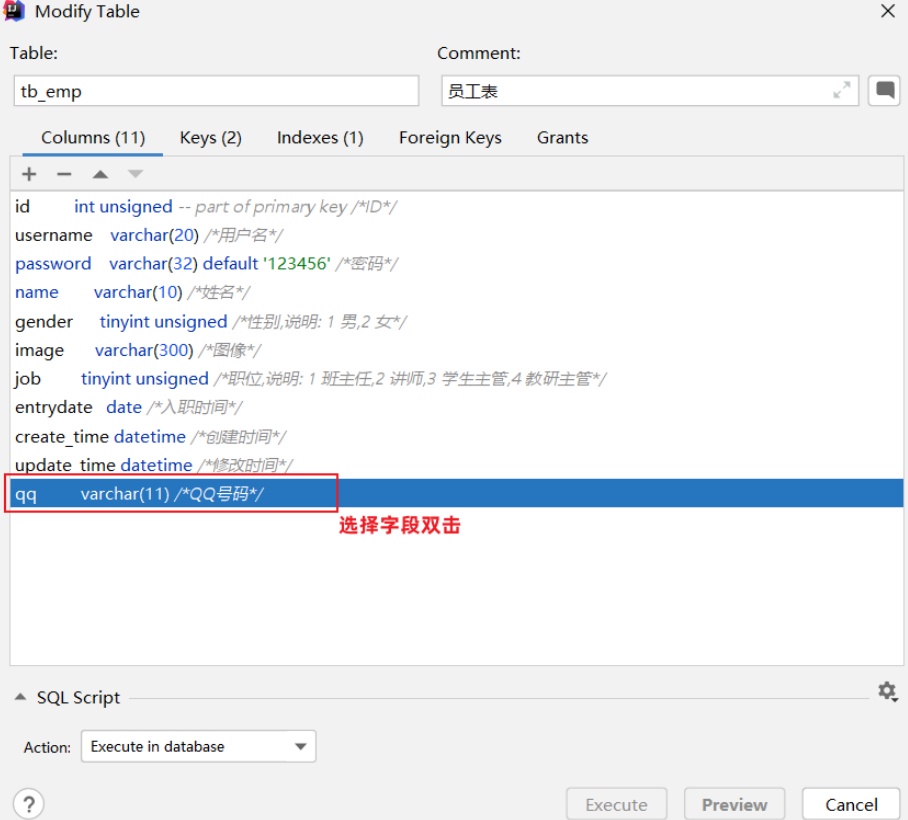


Modify Table



修改数据类型





Modify Table



SQL Script

实

Action:

Execute in database

و

Rename in:

Comments and stringsText occurrences Loaded sources

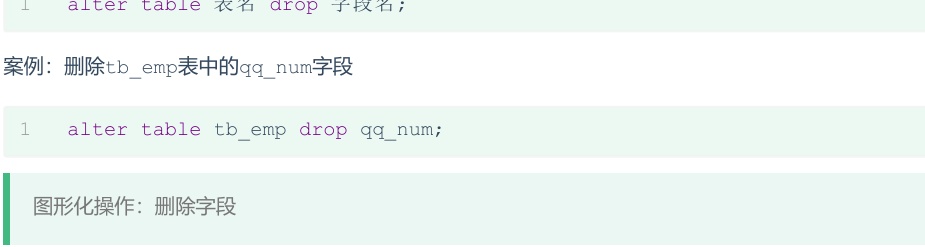
(?)

Execute

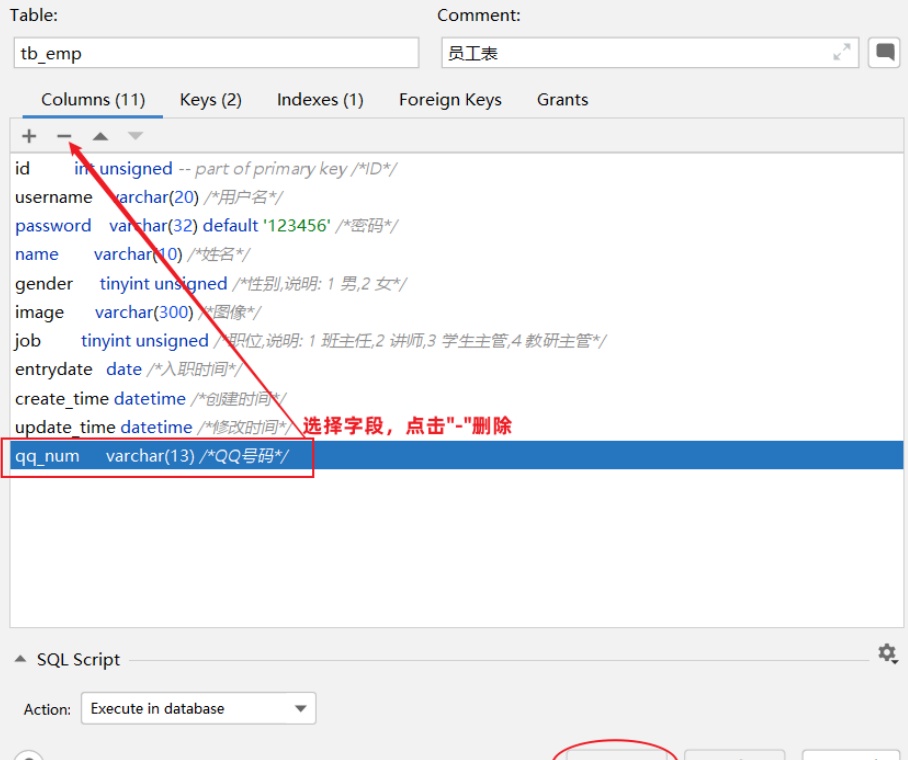
Preview

Cancel

删除字段



Modify Table



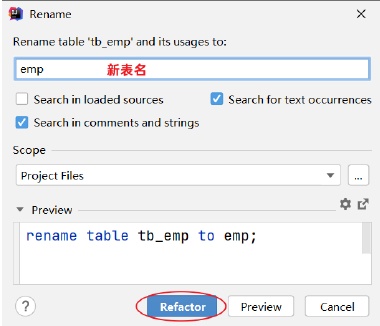
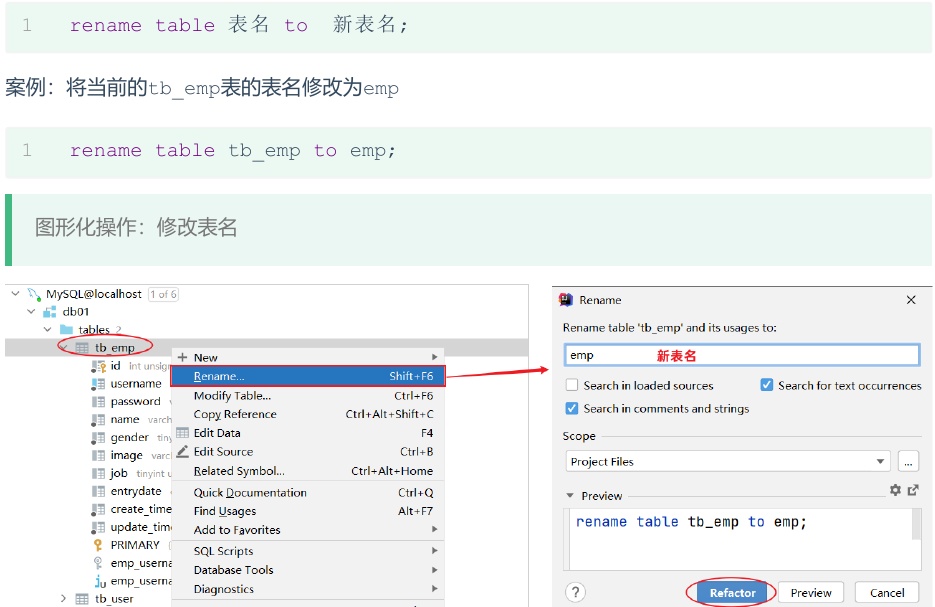
Execute

?

Preview

Cancel

修改表名



Drop

১

Server Objects

Delete

2.3.4删除

关于表结构的删除操作，工作中一般都是直接基于图形化界面操作。

删除表语法:

drop table [if exists ] 表名;

1

if exists ：只有表名存在时才会删除该表，表名不存在，则不执行删除操作(如果不加该参数

项，删除一张不存在的表，执行将会报错)。

案例:如果tbemp表存在，则删除tbemp表

1

drop table if exists tb\_emp; --在删除表时,表中的全部数据也会被删除。

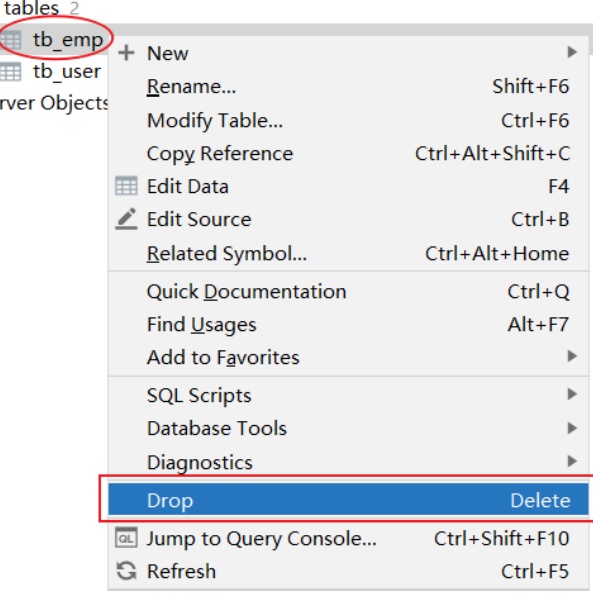
图形化操作:删除表

ا

MySQL@localhost 1 of 6

db01

ا



১

3. 数据库操作-DMI

DML英文全称是Data Manipulation Language (数据操作语言),用来对数据库中表的数据记录进 行增、删、改操作。

添加数据(INSERT) 修改数据(UPDATE) 删除数据(DELETE)

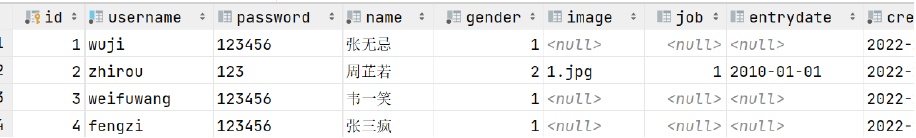
3.1增加(insert)

insert语:

向指定字段添加数据



图形化操作:双击tb\_emp表查看数据



Insert操作的注意事项:

1. 插入数据时，指定的字段顺序需要与值的顺序是——对应的。

2. 字符串和日期型数据应该包含在引号中。

3. 插入的数据大小，应该在字段的规定范围内。

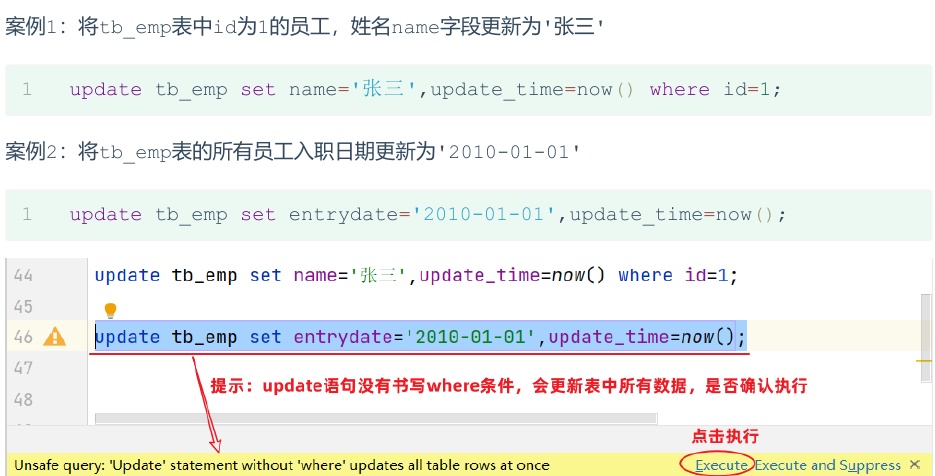
3.2修改(update)

update:

update表名set字段名1=值1,字段名2=值2,...

1

[where条件];



注意事项:

1. 修改语句的条件可以有，也可以没有，如果没有条件，则会修改整张表的所有数据。

2. 在修改数据时,一般需要同时修改公共字段update-time,将其修改为当前操作时间。

3.3删除(delete)

delete:

案例1:删除tb\_emp表中id为1的员工

delete from tb\_emp where id = 1;

1

案例2:删除tbemp表中所有员工

1

delete from tb\_emp;

注意事项:

DELETE语句的条件可以有，也可以没有，如果没有条件，则会删除整张表的所有数据。

DELETE语句不能删除某一个字段的值(可以使用UPDATE,将该字段值置为NULL即 可)。

当进行删除全部数据操作时，会提示询问是否确认删除所有数据，直接点击Execute即 可。