第19章 数据库编程

人类在进化的过程中,创造了符号、文字、数字等进行记录的工具，但是随着认知能力和 创造能力的提升，数据量越来越大，对于数据的记录和准确查找，成了一个重大难题。

计算机诞生后，数据开始在计算机中存储并计算，历经多年的发展后，设计出了数据库系统。 数据库是存储数据的仓库，数据库管理系统用于管理存储在数据库里的数据。数据库管理系统 解决的问题包摇：持久化存储，优化读写，保证数据的有效性等。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 目前流行的数据库管理系统属于各个不同的公司，图194是DB-Engines发布的2(H7年6 | | | | | | |
| 月份全球数据库排名， | | | 从排名可以看出，排在第一名和第二名的是Oracle和MySQL,这两个 | | | |
| 数据库管理系统都属于Oracle 司，第三名是微软公司的SQL Server数据库。 | | | | |  |  |
|  | Rank |  |  | . Scare | |  |
| *Jan* | May | Jun | BBMS Database Model | Jun | May |  |
| 2017 | 2017 | 2016 |  | 2017 | 2017 | 2015 |
| 1. | 1. | 1, | Oracbs，，或 DBV：S | 1351.76 | -2.5S | -97^5 |
| 2. | 2. | 2, | MySQL ；-. yT 飮姦g气祝 DBMS | 1345.31 |  |  |
| 3. | 3. | 3, | Microsoft SQL Server *可* EgE c?买临 | 1198.97 |  |  |
| 4. | 4, | 5. | PostgreSQL盘，*W* 歸Eg橱睥颱• | 368.54 |  |  |
| 5. | 5, | < 4. | MongoDB wZ Doc..,,：-ne；u sto-e | 335.00 |  | +30,38 |
| 6. | 6., | 6. | DB2. Ji Rdational DBMS | 187.50 | -1.34 | -1..07 |
| 7. | 7, | 5. | Microsoft Access Rfiiationa! DBMS | 126.55 | -3.33 |  |
| 8. | B. | < 7. | Cassan-dra Q Wide cobanm store | 124.12 |  | -7..0D |
| . 9. | q. | 蠻10. | Redis fj Key-value, stare | 118.89 |  |  |
| 10. | 10. | *'轟* 9.. | SOLite Reiaoonat uBMS | 116.71 |  | -MX y 2 |

图19』全球数据库排名

本书主要介绍在Python开发中使用较多的MySQL数据库。MySQL数据库排名一直在上升, 而且与第一名Ormk数据库之间的差距很小。

19ol My^Ol

MySQL是网络化的跨平台关系型数据库系统，除了商业版之外，还有免费版本供开发者 学习使用。由于具有免费使用、配置简单、稳定性和性能良好的优点，MySQL成为一个应用

广泛的数据库9与Python语言的结合使用也比较常见。

19D1J尊載粒曇礬

t革饑

可以到MySQL的官网上下载最新版本的MySQL数据库，下载地址是<https://www.mysql>. com/downloads/o

(1)打开下载页面，发现页面上有多个版本可选择，包括：

* MySQL Community Server ( GPL ):社区版本，该版本开源并且免费，但不提供官方技 术支持,是开发者的首选。
* MySQL Enterprise Edition：企业版本，比社区版拥有更多功能,需要付费使用,适合对 数据库可靠性和安全性要求较高的企业用户。
* MySQL ClusterCGE：高级集群版本,需要付费使用。

此外,在[https://dev.mysql.com/downloads/页面上可以看到给开发者使用的更多版本,包括](https://dev.mysql.com/downloads/%e9%a1%b5%e9%9d%a2%e4%b8%8a%e5%8f%af%e4%bb%a5%e7%9c%8b%e5%88%b0%e7%bb%99%e5%bc%80%e5%8f%91%e8%80%85%e4%bd%bf%e7%94%a8%e7%9a%84%e6%9b%b4%e5%a4%9a%e7%89%88%e6%9c%ac,%e5%8c%85%e6%8b%ac): 窗MySQL Cluster ( GPL ):免费的集群版。

* MySQL Workbench：为MySQL设计的GUI数据库设计和管理工具，在安装MySQL时 也可以选择安装。

我们通常使用的是社区版本o单击MySQL Community Server下的“Download"链接,进 入下一个页面，如图19・2所示。

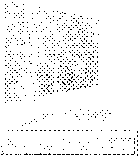
*" 峰夔部?感'拊熱点最瀚输囲峋"心噩藉-* 翥/W寻陌蜜节片猊携攵發磁;:巒感斜;涪

hlySCll Cooirnunity Server 5.7.1

J - Microsoft Windows

j Select Q5 Vers;-cp-： 顷二二 mm

Recommended

MySQL *Installer*

「「「「WlillwihdQfB

All MySQL Pg如砒队 For *Ml* Wmdom

*矚机瞄瞄* 伽嚣瓠 *甚盅覊.g囂％ 糰旗專 g谕瞄* mi

图19-2下载MySQL-选择操作系统

1. 选择操作系统和版本,并单击下载链接，这里以Windows安装包为例。

注意要选择MySQL Installer MSI安装包，它有安装向导，我们按照它的指导进行安装即可， 简单方便。单击"Go to Download Page"按钮，进入下一个页面，如图19；所示。

1. 图19・3的页面列举了离线和在线安装包，如果安装时没有网络连接，则建议下载离 线的完整安装包(mysql-installer-community-5.7.18.1 .msi),可以在不联网的情况下安装。单击 安装包后的"Download"按钮，进入下一个页面，如图19V所示。

Geoa：ra?■ ?y Ava-Jzab?e *勇"次膈独"踞* Development

Se:ec

! Micrasoft'VVindDWS

Windows MSI Installer

5.7..1S

M

*土\*过:必出濕农时忒施焉席\*03e：离編儘灯%専舄牌也*

1S.5M

5.7.18

-:，-r-h-

图19-3下载MySQL-选择安装包

Begin ¥our Downhaad

mysaJ-installepcomrnunity-5»7/^8/: .rnsi

Login 间Em *飘* Sign Up for *夺* f遗蓮 account,

An Or^cj-e登农e 怂:指:由笠provide; you- vvBch僧花 牡&卷痼 就以卷混咨寥％

*•f 益己:心浴*to r/(ySQL三疔的M法很己ck総E宾沉M

*-\*潛”虑电* Whru Papers c-nci ?resr.n:=3Bons

■- Post --oess^ges 潦 the 澆"3Q4 Cuscuss^Gn Por-urns

■■■ Repon:灘态"&去&一£5 k' the ^rySQL bug *"w：*

，Ccrnmero： 0 :che MySQL *E"5司毕*



MySQL.-: orn 賭 vs eng *E腿* SSQ for suth^r；r；c^^on-. bf yoi-； aire^dy h：ave Web

*e欢r；:RE：典吏* login G-nk. Cnh^wi-se. *gw.* £(gnup & 荏*牝渋况* by 界 E方宅 the Sfen

Up unk g fo<Giving the 滔•必用C弋’終.

*».豊栓翌芝邑上翌X竇\*貞3夷l•缨遂l弦9亜！L-*

图19・4下载MySQL-忽略用户注册

1. 如图19・4所示，在下载安装包时网页会建议使用Oracle账号登录，如果不想登录或 注册。皿尼账号，则单击左下角的直接下载链接即可。

2.寰装 「

下载完安装包以后，就可以安装MySQL 了。

(1 )双击安装文件(mysql-installer-community-5.7.18.1 .msi),启动安装，首先选择安装类型, 如图19-5所示。

安装类型有5种，分别是：

° Developer Default：默认开发版，安装开发所需的所有功能。

• Server only：仅安装服务器,也就是MySQL Servero

° Client only：仅安装客户端,也就是MySQL Shello

。Full：安装MySQL的全部功能。

° Custom：自定义安装。

Piesss select: the Setup Type that .suits your use case-.

MySQL Installer

g ； 陟'"“M

*普斂恐晉*Typ；c*专卷撰伊海貲*

*丢松加*ths ^>/S0l.長淳渋扑舟 *長：.Uwh* g

*»後普宣迎* 海:矿K:zySQ.L若憂営i让就泠口次玲怂成时泠咨‘忠「亍 ~"\〜七八：?二‘、：产"-~「 匸;

点戸"拭澆让冼号tor .■■•n 获康弟券 汹泠/‘,. {|

?，MySQL 5h-n :

*淬顎:*约"：MyS^Ql *uk:ai發神:拭疝膏\**燃 汚结關专芒， ;MySQL §毫\*會性*:酒点齢袞翌/；\*,* w陽法豚盘春拦芝金％ j；

j-; M>c?QL EE

*件芯"z品脳邕:*fester d<；s；r-c.n *沁* InrioDS 过出疫以 *\*心\** 您 靜宅 *E以糸戒* on必洱叔:就熟資 *\*戒鴨.*

MySOL W&ffch^nch

*狀泌聞密* 云拾戏针诚 密痘佥用怂 *涟上蕊' 汨曷"眼诟』混整滩*92拓W

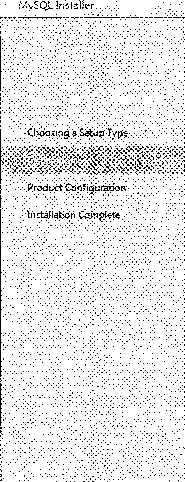
jlhc- GUI京戸賈訟滅;cm tcv盛財龙毎嶷 舔?刑淑 g由捋海拆由泛*E"知、*

也2 > ： *I* Cancel

图19-5安装MySQL：选择安装类型

在这里我们选择默认的开发版，单击“Next”按钮，进入下一步。

(2)下一个页面列举了要安装的所有组件，如图19・6所示。单击“Excute”按钮，开始 逐一安装这些组件，并显示组件的安装进度。

计5广，注

Press Execute to- upgrsde the faltering- predicts.

Clkk [Execute] ta install or update the faitoviing packages j < Back

| i ; Produd | Status |
| --- | --- |
| '\ Sir. e>- 5 7 .5 | R；eady to tetsll |
| *] 1* MySQL VtfQrkber<h S3..9 | Ready to Instaii |
| i :終：j MySQL N'-Kifier *11J* | *眼邕聲*to Ir^stsU |
| j 5X'-j： WSQL For E-xc«l 1..3,® | Ready to InstsH |
| *i* 盘上3關yS盘柘E'i泌营或山论*12.7* | Resdy t 爲 InstsU |
| | [" ： MySQL Utilities 1..6-.5 | Ready to |
| ! 技；」M^SQLSnen l；0-.9-■ | Ready to Irstsfi |
| \ 信上：険压啓Router *213* | Ready to tots JI |
| ! Uf'J *3ggg备L* 5.3.,7 | Ready to Install |
| | "：气 功眼由诚>+ l.l；-3 | Rea-dy to Install |
| *| wn* Coi-a'/S-dOf/J 5141 | Ready to tetslj |
| j ConnertGr.-NET 6.9?3- | Re^dy to F.胡 |
| ( ! '-7 *% !； c岩wa.吾点Hr捋- -iid'- /* 貝沒 | ..,......................段馬 £技.；切击泌引£..... |
| *mssssd* | KSSSOSOIOOOW |

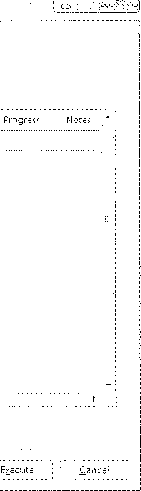
(a)

图19-6安装MySQL：开始安装和显示安装进度

|  |  |  |  |
| --- | --- | --- | --- |
| \ Produd  广二'"蘇志毎"”"二 Em 宀  *『襪*!"；.« K'fySQL *心膏* 5.7.18. | Status  Cc<mpleie | Progress. | 如也二j |
| j 新'?..'：\ \ MySQL VferktogP^ivtas | 協.痢购 |  |  |
| j j?；： { h%.'5QL ftotifier *1,11* | R-ead^ta Instaii |  | 顷 |
| | j)：\ j MySQL For Exts! 13,6 | *浦賊* t o |  |  |
| 1 y\ j MySQL for Visual Stvds 1.2,7 | Ready io Install |  |  |
| | ：{..；;.'■■■ *'1* M^5QL Utiiitiss 1\*.5 | Rea-^yt© Im-tall |  |  |
| I yO- } MySQL Stell 1.0.S | Resdy to Imtai? |  |  |
| I W MySQL SW 2,1,3 |  |  | 顷 |
| \ Co^redor/ODBC B3..7 | fe-Bdyiolnstsil |  |  |
| j 拦:言 CceWMA i.1« | Ready to InstaH |  | j |
| i 瘦片  j 毎三呉*S学戯云套戒*5.142 | Fww£危 |  |  |
| | 破匸即点荘寻妄•，泊 | Resave to *Install* |  |  |
| ） *'編;m*  ［；,,一.顶f丄痘由參航阳：崩建.恩孩揺，盡…“..…, | ,砖斌点由E\*丄“. |  | Bl®""」 |
| *| 宜g* Datails > ( |  |  |  |
|  |  |  | : C-ancel |

Press Execute to- upgrade *the* feltewing pra-diKts;

图19.6安装MySQL：开始安装和显示安装进度（续）

*家*配置

安装完毕后，继续进行MySQL的配置，如图19・7所示。

Produce Co*賢*figuration

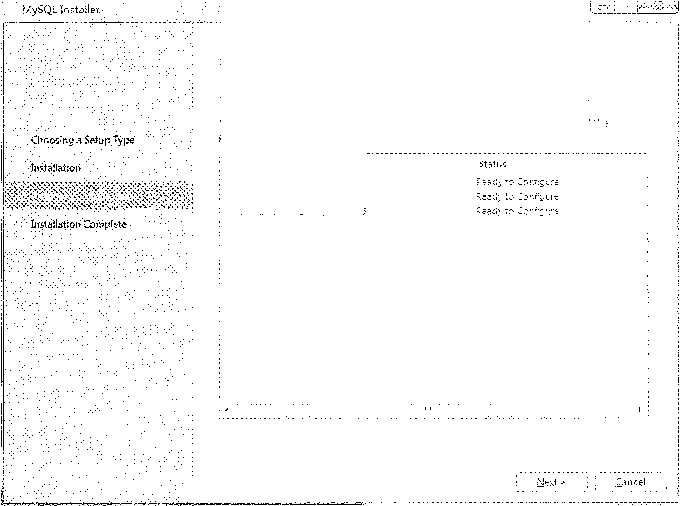
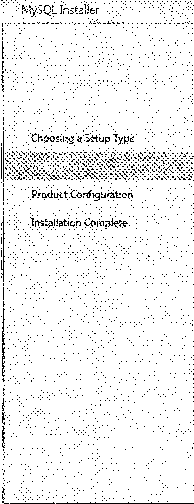
Produd

明诲 Server 5,713

MySQL Rootsf 213 玉束弟⑴抒 云鄭Examples 5-J.18

We'B now ?^alk through a- configursticn wizard fcr each cf the foftevving products.

You can cancel at any point if you wish to leave this wizard without configuring -alt the products,



可以看出，配置分为三部分:

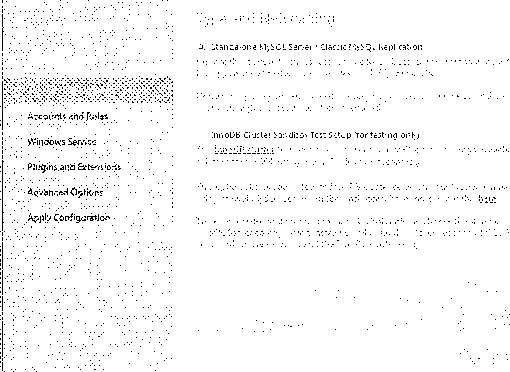
® MySQL Server 配置。

• MySQL Router 配置。

® Samples and Examples 样例配置。 具体配置步骤如下：

(1 )单击“Next”按钮，进入MySQL Server配置，如图98所示。首先进行类型和网络配置,

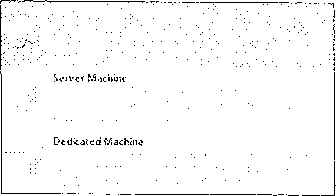
选择默认的a Standard MySQL Servern选项，然后单击.“Next"按钮，进入下一步。



Cancel

图19-8配置MySQL：服务器类型

(2 )下一步的显示界面如图19『9所示，这里可以选择数据库服务器的类型和网络连接方式。



*細攻mg心*

选择数据库服务器安装的类型有:

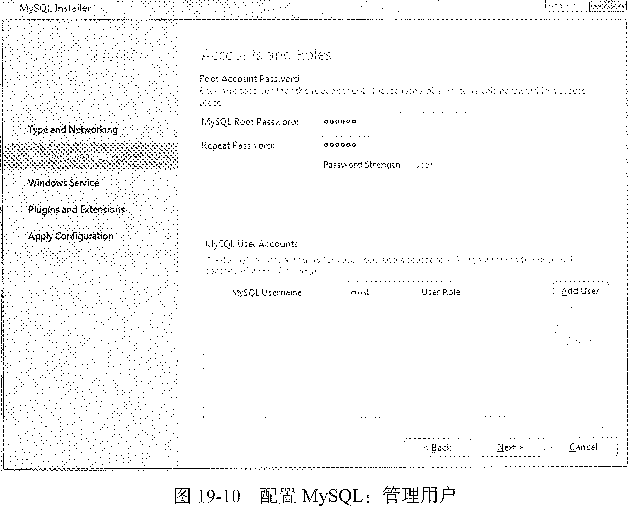
° Development Machine：数据库安装在开发机上,还会安装很多其他软件,这种类型下 MySQL数据库将使用所需的最少内存。

® Server Machine：数据库安装在服务器上，还会安装其他服务端程序，一般用于网络应 用或服务器端应用。这种类型下MySQL数据库将使用中等数量的内存。

。Dedicated Machine：数据库安装在数据库服务器上，不再安装其他服务器端程序。此时 MySQL会充分利用所有可用内存。

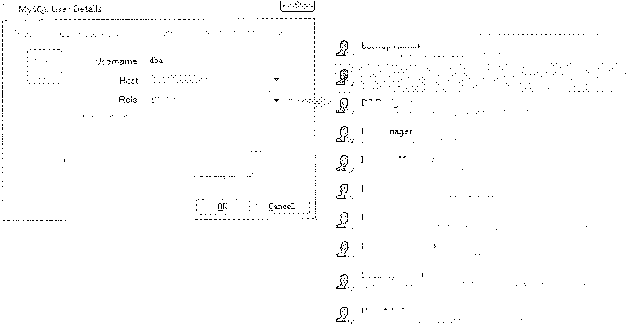
选择第一个"'Development Machine9',单击“Next"按钮,进入下一步。

（3）设置Roof用户的密码并添加用户，如图19』0所示。



在界面上方设置MySQL Root账号的密码。如果要添加新用户，单击"Add Us©产按钮，

进入添加新用户界面，如图所示。



Backuo Admin

MySQL

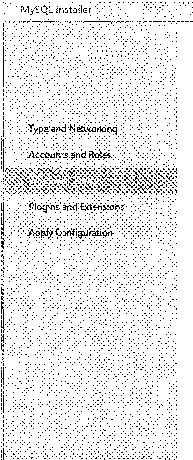
Aiitiientication

Password

Cc-nfinTi Password

User Admin

Password Strength：*"每*



增加用户时可以选择用户的角色,包括DB Admin, Backup Admin等。单击“OK”按钮, 可以看到，新加入的用户已经显示在界面上，如图19J2所示。

|  |  |  |  |
| --- | --- | --- | --- |
| ?V?ySQL Instalter |  |  | F 妙.宀 |
| *簞矚鑼靄讓靄霸囊讒嚴* |  | -- 1 1 ~ ~ —— — ~‘■— | !■： |
|  |  | Xvccounts an：d Rorf；s | [- |
|  |  | Root Account Pas^ord |  |
|  |  | 臼钊涪th-£-'步蕊.兌孕M\* *沁湾¥云：*呉:瘀:\*wz,滂丄 戶以編當节涝染'也:？聽提尊； | 祚咨％卩儀"溜点呂泊a *活 w件* |
| *嚨麗?聽聽麗囂畫讖麗噩靄叢\*囂?矚離矚飜i衆* |  | *成e* |  |
| *'囑讒蠶讒骚聽麗醫體覊矚囂囂竈囂讒爵* |  | MySQL Root Pssswar^ :麒g 翱 g |  |
|  |  | Repest Password; ‘ |  |
|  |  | Psssv/GFd StrengtT *中住當,* |  |
| J 緬诺场轴两s |  |  |  |
| *i* 語捋g 蛔 E滅江蔚拙 |  |  |  |
| 細遇 |  |  | (■; |
|  |  | MySQL User Accounts | [j |
|  |  | *Cii&:.i：：* MySQL*仙wn心;.爲待9対 迎；早部京*録钙浩由;挪 現帶邕滅泗北丄Assign | 岛口？m to*術迁七性混*打;•泓 § |
|  |  | Set Or*令訂品:呉牌.* | ........…......……*「―—.,[* |
|  |  | j MySQL Us-erna-m.fi- : Hast User Role | U Add User j ; |
|  |  | *\*囊縁蠶轟蘇觸鑫BIIHtl續蠶讓靄隸* | *fKffKi.* 打花廿楚C ! |
|  |  |  | ;\ Delek ； *1* |
|  |  |  | 里ext " J j； 卷晶国 ？ |

图19J2配置MySQL：显示用户列表

（4）单击图19J2中的“Next”按钮，会进入Windows Service设置的界面，如图19J3

所示。将MySQL服务器设置为Windows服务，就可以在Windows服务列表上进行启动/关闭 等操作。还可以设置为在系统启动时自动启动MySQL服务器。

Vvlndofvvs Service .

/ Configure MySQL Server 匹 己 界补钢匚執三 Sen ice

Window Sendee Details

*沔\*\* 咿相W* 法1浜点同W*三仲辭:g* r海我诺*W* 眺 寥点 氾曹代怎 *料典堀.*弛g溟 *泗泊\*/啲'买*弟幷密展

*熒貞吨热編:亵心微混南;混成討号臥* i

Windows Service MySQL57 ；

'Start the at System Startup /

Run Windows Semce as … j

T hz? MySOL. Se.r；/e.-/ needi to*如拦罚飞器叫*気苧海牡升上wm會::寸，消滅，政混簿#尚吃沙必.挣"小哭亨 ；

羿牌眼me編相of ：耕撲 玛节暗泠 汚以注臨呂 *扣*pick*臨州.*謗 时坯港宓泗旧bdc->7-

® Standard System Account /

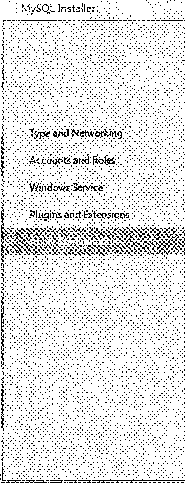
*袞迁気件汨冷姦買苴W貝* 鴻口7治成 K更性•訂湾& ；

#7- CLi5tcm User （

為承W辭共洪号 溢洲W；： #提仪 *匸落7*打&浇極£：农宓70"混扣;对拭:琮时三心手飞組隹法' ；

图19・13配置MySQL：配置Window Service服务

（5 ）下一步设置是否开启MySQL的文档存储功能，如图19-14所示。文档存储是MySQL 5.7



却尊蝴臨掬馈灿翌

*4毒5氐加株鴻血*

*寮眼瞄鸣*Sm4ce

*鬻麒螭娜黝mt/麗*

Apply腸曲華阀■瞄

L....<Back.. J〔…婆竺旗」L印叽」

推出的新功能*，*MySQL提供了 X插件支持文档存储功能,并且使用X协议与数据服务器进行 沟通。将MySQL设置为支持文档存储，就可以在数据库中存储JSON格式的文档,并对其执行增、 删、改、查等各种操作，包括创建索引等。

Piugtns and Extensions

MySQL a-s a DDcument Store

*曇:*ths *h* i抵沈必号 w羚d心绍.龙芸*睥 泌楓点like,* to 粕 汤沁•強八恐匚

r.) Enable X Prow col / MySQL g a Document Store

Port Number: *‘司云‘*…“…；

" Open F如的習W!卩心请 标门常稣泌険若亡:費英

Starting with MySQL Server 5.7, MySQL suppoits document store development, In order to provide a coniplete document stQre/WoSQL experience there 饪 a new comfflunk.atiQns pFQtaeol calkd the X ProtGcoL The expanded capabilities of the K Protocol enable us to provide modem, dewloper APIs with- features such as asynchronous cafls.. pipelining, and： more. In addition ta Implementing docirment caUe^ion：;, the new X DevAPI also suppers re^tionai and combined docitmenf storeAeMional capabHities, Mow developers, designers and DBAs can deploy M^SQL da'Ubasesthat implement document store, relation^ or hybrid ciacLiment/rel-Btion mede^.

£lkk 區竺荃圣£坦匚华箜並表坦延盘咬施圭此曇觎z

111^1111^110103

图19-14配置MySQL-设置文档存储

1. 接下来应用这些设置，如图19J5所示。

App(y C.o nfi gurati on

lExecute] to apply the chsnge?

J Configurotion Steps 払擾愛检滅j

I *Q* Stopping Server [if necessa-iy]

1 Writing configuration file

： 虹U Updating firewaH

i Q Af^Ltstrng. Windows service [if necessary!

j 、 InrtUlizing Database [if necessary]

j O Starting S.er-/er

\ 0 Applying secyrity settings

Creating user accounts

Updating Start Menu Link

Updating. Fireball for MySQL Document Data Feature Set

图1945应用设置

MySQL Ir^<3；ler

*Product* Corrncruratiori

vVell now w-3lk thrcugh *瑟* configurstion wizard for each of the follD-A\*ing pro-dircts.

Choosing 起 Setup Type

Instsfedon

论密冨fer慾典

You can cancel' at any point if you wish ta= leave this -wizard witho-ut configuring- sll the praducts.

Product

Status

C?rnot/3i7 3r Cz^s'plet\* Ready 电,Cor；figur-£-

*Rwg l法 &*

Next

Caned

图19・16服务器设置完成

第二项是设置MySQL Router,用于数据库的负载均衡，这里保持默认设置即可，如 图19』7所示。

QL Iz-jstfiiler 1 阮協E

*鸞驪讖* ■MySQL Route?- Co<ifigiH"atfon

Configure M> SQL Router for InnaDB cluster,

*If警鬢麗麗麗靄離驚養寵鸞養矚?靄靄骚麗麗麗饗鸞囂？簷矗靄f蠢 竜足*捋注眾滔£辎 膈出抄冲the苗罗泓Routei-混7溢时 燮知耕玳:"空M'.y;GQ：L明举凯關泓哗必r■湧&

蕊辎蟀 M'y;SQL*契q忒薄* 貞强罗務，*異\*耶\*忑注\*导與% IM：.芸財择钮电法.*to杪件 做投恣f旧趙 *牴* 舉就心終应技微徳 *盘我W节点 凭*

礴聞 *:訟:蘇球部後* 快典密 斜 R/C?*魚會誓每帝 m* thft *臨哄电..*

■ ■ . ^ ;

*I囑麗矚矚嬲矚爨囑矚欄矚\*飄矚嬲聲矚矚 E%志翌;*終絲拋如*藏*w滔辫混戏独*却忿*粕史◎酬f认眼就E必&泛*In*沏滋尸陌以弥E the MySQl箭炽建淀 j

*V養爵麗書靄靄鬢麗爵麗疊鸞麗靄普麗饗饗擾\*\*i蕾麗麗噩靄善素靄矚囂3* 陽幣E辻財頌遗u；;e*加*汨汚爾财点萍酒法消盘諄財理of th^尊關史如 %

， 緇緇餾 Hastname;:*辄拠密点*st. :;

Port： -3310 %

| Management th巴'：：洁咨温 j

Password: :- j j

舗懲 MySQL *\*心佥顼託牛Mg* :spednGrbor? of a i皿网声混\*怂就容再h £0 snd *W：：3Z],* Trus \*明？迂拦w紅•；*睥：* ；

躍爛 *戏\*混w*危就挤，"寤•：, °財消狀•;：沁?巽，了臨□泠苗•浄加七：*睥赚 5琶後* 的总•技"也題\*混挖f击矽 *g炊*po-fL If *冲 ■:*

*, 汚曲心睥：;我剛編拓恆恩部睜壬海^局貝京鑫抵：財\*紅\*世奇狀睥,臨肅*p&rL ij

Classic MySQL protocol Lannectsons to [nnoDB cluster :

\ Read/Write; *\*向感■- ;*

\ Read Only；専狷？ : ■:

M^QL X Protacol connections to InnoDB duster； j

Rencl/VVrite-: jz544S '；

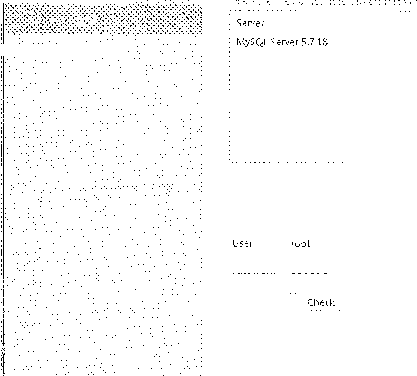
Read Only-：關耳奨 J

! Finish ! ? Cancel- / j

图19-17 MySQL设置：设置MySQL路由

Connect To Server

MySQL InstaHer-

务圮榆*"光M* 犧衆号.球写扱愍成论 診岐 *加：血；部我.*(neecfa 部 *品总忒* 再成®总应豎拦誨"OE：疫驀財初 抡 汗郷如楚 *g會.*［切貯 書終嫩. ^

Apply*铃忡枷诚5*

Connection successfuL

Architedufe

X64

Pas 5-WQ.Frf? *: ®. ®. ® a >-#*

Status

Running

图1948 MySQL设置：设置样例和示例

Cancel

1. 单击“Next”按钮开始应用设置，设置完成后的界面如图19-19所示。

We,H now walk th-rou^h a ca-nfigii-ratio-n wizard for each of the fo-ltewing products.

You can cancel: at any point if you wish to leave this wizard without canfiguring, all the prcdurf<.

iiWit¥'ion Complete

Produd

FJ^QL Ser 5 715

MySQL Renter 2.^3

&顔必卩电5 •采诺*Exg旨y* 5J.18

Status

Initial Co脯堂uration-

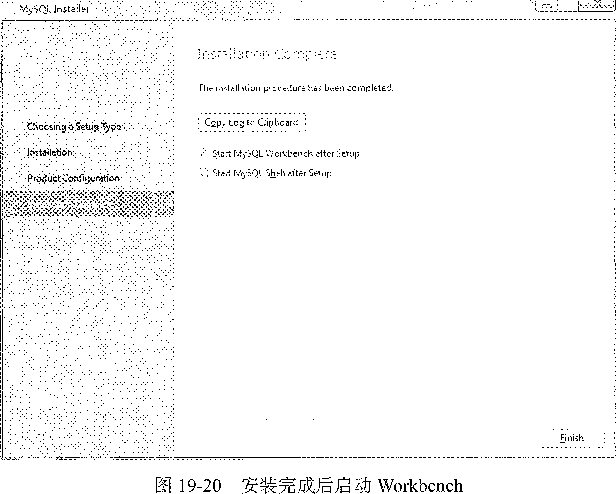
C^rtipii3iJC-r C^jrpfe^

Next >

Csncd

图19-19 MySQL设置：设置完成

1. 安装完成后,启动MySQL Workbench和客户端MySQL Shell,如图19-20所示。



1®J.2 « MySQL W©rkb@o©h

MySQL Workbench是一款专为MySQL设计的数据库GUI管理工具,在安装MySQL Community Server时会自动安装MySQL Workbench,除此之夕卜,还可以从官网上单独下载。

使用MySQL Workbench可以设计和创建新的数据库、对数据进行操作，以及迁移和备份 数据库等,并且可在Windows、Linux和Mac系统上使用。

下面以MySQL Workbench 6.3 CE为例,说明如何使用这个软件。

(1 )双击打开MySQL Workbench,打开后的界面如图19-21所示。

Welcome to MySQL V^/orkbench

MySQL *Is* the graphical (GUF)扣& for MySQL. It a^ows you to design,

create and b-rawse your database schemas. *混* database ejects E insert data as w或 as

design and 拽■滔 SQL to y/ork wU-h sic-red dais. ¥o^ ran *also* ；Y^gr&te *云冲仲表* and data 和食芸 other

database vendors to yotir Mv6QL database.

?淫门扑"在二心了只尖注氏?3罚,注混交汚就”如菱損强点/■ 法:净二3Z W雄?-'■■■/■/ /,■?；■?■：-■■：- ■.-

MySQL ConnectiGns ® ® 岛苔E o?湘此如5

*l-QC&i* ?\4ySQL57

至炒



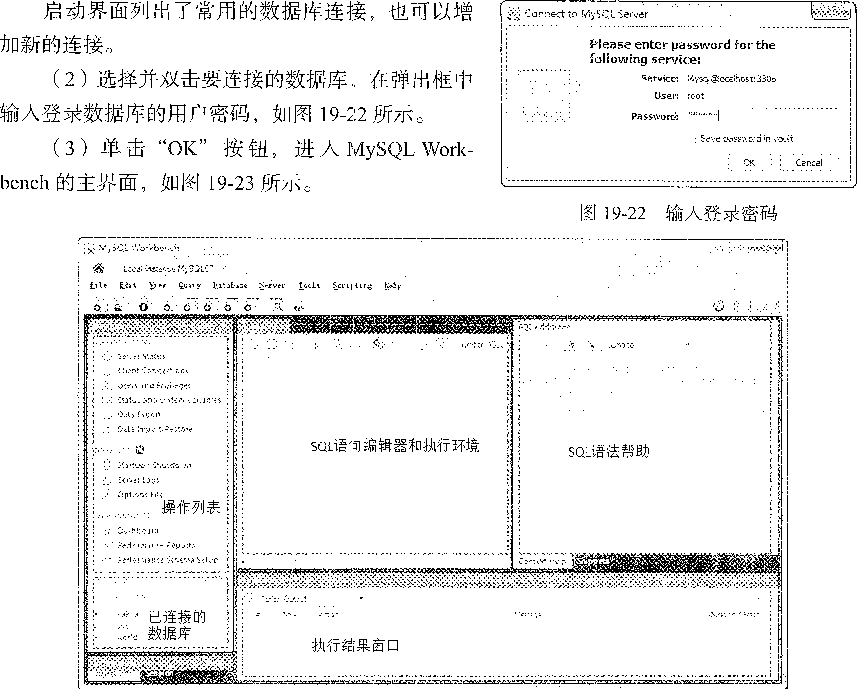


图 19-23 MySQL Workbench 界面

该界面主要有以下几个区域:

•左下角区域是Schemas,列出了已连接的数据库以及数据库包含的内容。

。左上角区域是对数据库的操作列表，包括用户管理、数据导入导出、数据库实例的启动 和停止、日志、性能报告等。

•中间的区域是SQL语句的编辑器和执行环境，可以在这个窗口编写SQL语句。

-最下面的区域是SQL语句的执行结果窗口，对数据的操作结果会显示在这个区域。

•最右边是SQL语法帮助窗口 ,包含各种SQL语句的语法规则和示例代码。

19J.3 WO DlySQL 「

要想卸载MySQL,最好的办法是使用MySQL的安装程序来实现。步骤如下：

(1 )双击 MySQL 安装程序(mysql-installer-community-5.7.18.1 .msi),出现图 19-24 的 界面，显示增加、修改、升级和删除等操作类型。单击“Removes”按钮，进入删除界面，如 图19・25所示。



MySQL, InstsHer

MySOL

Installer

*i* Product

j :- MySQL Server

5 ;MySQL Workbench

4 ■; hlySQL Noti fi er

;j 加拓QL For Excel

I MyZQL for Visud. Studio

I MrZQL Utilities

]MySQL Shell

j : MySQL Rout er-

：| Coimectoi'/ODBC

d iConnsctoe/匚++

j ；Connector/J

*i* iConnector/NET

§ Corinectoi'/Python (3. 4) 2. 1.6

d ：- MySQL Cormector/C

J i MySQL Documentati on

1 ^SampleE artd Examples

| V | ereion | ；Architecture | :Qui ck Acti an |
| --- | --- | --- | --- |
| 5 | 7. 18 | X64 |  |
| 6 | 3.9 | X64 |  |
| 1 | 1.7 | X86 |  |
| 1 | 3.6 | X86 |  |
| 1 | 2.7 | X86 |  |
| 1 | 6.5 | X64 |  |
| 1 | 0.9 | X64 |  |
| 2 | 1.3 | X64 |  |
| 5 | 3.7 | X64 |  |
| 1 | 1.8 | X64 |  |
| 5 | 1.41 | X8B |  |
| 6 | 9.9 | X86 |  |
| 2 | 1.6 | X64 |  |
| 6 | 1.9 | X64 |  |
| 5 | 7. 18 | X86 |  |
| 5 | 7.18 | X86 |  |

Add ...

Modify ...

IJpgrade ...

Remove ...

\\*-ySQL InsU !er

图19-24选择操作类型

Rer?)ove Products

Ptease check the pro-ducts, you woul'-d like tc- have removed.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Status Progress | 二］ |
|  | ?[£ M^SOL Sen/er 5718 *螂聽\* i* | Read^ to Rgmsve |  |
| *瞬感* Couplets | \ 国即L bench g 5,9 | Ready to Remo\?e | 1 ! |
|  | MySQL Notifier 1.17 | Re-sdyts Remsve | i |
|  | j ； MySQL For E>ceil3(6 | Ready to Remove | 1 s |
|  | :7 MySQL for Visual StudiG 1,2,7 | Ready to Renw/.<e | ill ! |
|  | ! 1 M^QL Utilities 1,6.5 | Readv to Remove | i i |
|  | !MySQL Shell 1,0,9 | Ready to Remove | II |
|  | j ；2 MySQL Router 2,13 | Resdy to Remove | 1 i |
|  | | 1 Ccnneear/ODBC 53.7 | Ready to *钦sg已* | || I |
|  | ，/ ConnectGr?C-- 1.1.3  i . ■. , | Ready to Rerfiove | 遏! |
|  | "也 Connedor/J 51.41 | Readyto Remove | i : |
|  | j ， Lonnedoi/UETS,?.? | Ready to Remove | 1 i |
|  | *\ 燮* Cannertar/T^thon (3,4} 2.1.6 | Ready ta Recno^e | if i |
|  |  |  | ZJ i |

Click (Execute] to remote the seleded pack-sqes.

Execute [ ； Cancei

图19-25选择删除的组件

(2 )在图19^25的界面中，选择要删除的组件。勾选全部复选框，然后单击"Execute"按钮。 系统会弹出一个确认删除的对话框,在对话框中确定以后，就开始删除各个组件了。删除过程中， 可以看到各个组件依次被删除的进度，如图19.26所示。

MvSQL

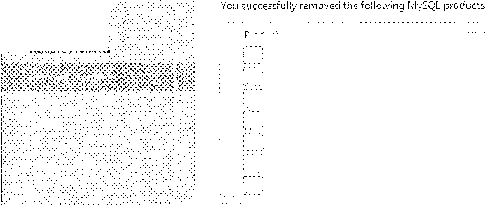
|  |  |  |  |
| --- | --- | --- | --- |
| Ples^e ch^ckthe products you would iiketo have removed. | | | |
| 技負 Produd: | Statiis |  | \_—BQ\* • |
| \ MySQL Server 5.7.18 | Cornpiete |  |  |
| | MySQL. Workbench 5.3,9 | Complete |  |  |
| | MySQL Notifief 1.1.7 | Complete |  | ■ 「：|j , |
| | MySQL For Excel 13,6 | Complete … |  |  |
| | Mj'SQL for Visual Stuciio *1,2J* | Renwdng j | 52% j |  |
| 1 MySQL Uhfities 1,6.5 | Ready to Remove *卜* | .! | ：项. |
| \ MySQL Sheli 1.0,9 | Resdy to Remove |  |  |
| \ MySQL Router *2.13*  *i* | Ready is Remove. |  |  |
| *I*  | Conned-QF/ODBC 53.7 | Ready to Rem-Gve |  | ..M |
| ] ConriedoF/C^ 彳 1.1,6 | Re^dy to Remove |  | ..胃• |
| \ 51.41 | *和心龍* |  |  |
| j Connedar/NET 6,9.9 | Readi/ to Remove |  | *\* |
| \ Connedsr/Python p,4i 2,1.6 | Ready to Remove |  | .「j |
| | Shew DetsHs. > i |  |  |  |



图19・26删除的进度

（3）删除完成后，单击“Next”按钮进入下一页，可以看到删除组件的日志。选中“Yrs, reboot when done'5复选框，然后单击“Finish”按钮。系统重启，删除就彻底完成了，如

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 图19・27所示。 |  |  |  |  |
| VlySQl InsieHsr |  |  | *号養S?g竇?睡g饕g魅* | : 护爾时 |
|  | 、- …- ——… -- |  |  |  |
|  | rer^:r/e Prc-nucn |  |  | 1 |
|  | Ptesse c hec k the praducte you wcuki like tc- hsve remcv&d. | |  |  |
| *協謫編艦蠢Kl/tt蟲* | j h；.Cl Product | St-StUG | Progress | *―不* |
| *欄』豔普薰鑼籠讓\*Mt廳解\** | | M^fSQL For Excel 1.3.6 | Complete |  | ,"彳 |
| \ 爲敏泗詞£o呻心 | | MySQL for \*i汹引 Studio 1.2.7 | Comptete |  | j |
| *"靄麗觸讓矚讒轟讒囂讓躊讒靄麗羸* | j MySQL Utilities 1.63 | Complete |  | 匚尊？ ! |
| *草!囊靄繼編讓""睡讒* | j MvSCU Shell l.Df9 | Complete |  |  |
| *瞼* | | MySQL Router 2.13 | Complete |  |  |
|  | \ Connector/GDBC 53J | Complete |  |  |
|  | \ Connedor/C\* - 11>3 | Complete |  |  |
| *"""繼"* | \ Conned:sr/J 5141. | Coinpiete |  |  |
| *"""薰* | \ Cannedor/NET 6.9.9 | Compfete |  |  |
| *奮蕾聾麗畿靈驚韻鳶養憲麗鶴養矚囂\*囂麗?畫蕾龍蠢驚饗養謂?最榛醫體蠢.* | ] ConnertGr./Pjthon 2.1.6 | Compiete |  |  |
|  | ! MySQL CGnnector/C6tE9 | Complete |  | 技繍 *1* |
| *驚縁疆鬻繼嚴讒潔薑靄轟* | j MySQL Documentation 571.3 | Complete |  | 日 j |
| *暨聽歸矚驪醴髒讓聽聽體編矚驪書* | *I* Samples and Examples 5.7.13 | Complete |  | 1 |
|  | *1* piWhg録B就流忘ww |  |  |  |
| *嘱關繼囊矚矚矚靄縁讓縁蠢* |  |  | j Ne^t > ； { | Gance! ! I -….二…---……*1* j |



MySQL 玷 mt 詞 er

Remov；3：i Com：ptete

Remove Prt-4a<h

55.18

MySQL 5,718

MySQL Workbench： S3-9

S3 5

? MySQL Notifier 1.-.17

11J

1.3.S

；；MySQL For Ex^l 1-3.8-

SQL for 驻捎H 丄厂

2 .-.2-7

MySQL Utilities *1&5*

1.DS

京二丄，知.W

You ha\^ remoyed aH your Community MySQL products, Would you like to uninstali the MvSQL Installer too?

^rodud

Version

*'饪丄* uninstall the LhfSQL Installer

One or mare of the produ^s rerna'^ed require a reboot to compiete. Do you 'Want to rebs：tGt vout computer v^hen denied

*■j* ¥esr reboot vvhen done

Finish' j

图19-27删除完成(续)

19«2 SO土语言

SQL是结构化查询语言(structure query language )，是目前应用广泛的关系数据库标准语言， 具有易学易用、功能丰富的特点，它已被国际标准化组织(ISO)采纳成为国际标准的关系数 据库语言。MySQL数据库也支持SQL语言的操作。

SQL语言操作数据库的主要功能包括：

®数据定义语言(Da馆Ddinition Language, DDL )。使用SQL语言可以定义数据库的各 种对象，以及数据的完整性约束等。

'•数据操作语言(Data Manipulation Language, DML )。SQL语言可以对数据进行操作, 包括数据的查询、增加、删除和修改。

®数据控制语言(Data Control Language, DCL )。SQL语言还可以实现对数据库的控制, 包括数据一致性、完整性和安全性控制，以及数据库的备份和恢复等。

19o3麴1库墓本操作

一个数据库对应MySQL中的一个Schema,数据库是存储数据库对象的容器，包括表、视 图、存储过程、函数、触发器和事件等。可以使用SQL语言进行数据库操作，包括创建和删除 数据库。

19d3J

创建数据库的SQL语句使用Create Database实现，语法格式如下:

create database db\_\_name

其中，dbjiame参数表宗要创建的数据库名称。数据库名称需符合以下条件; •不得与现有数据库重名。

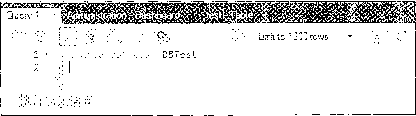
•名称可包含字母、数字、下画线、@、#和$符号。

•首字母不能是数字或$符号。

%不能使用MySQL的保留字。

•不能有空格和其他特殊字符。

。长度小于128位。

如果数据库名称不符合要求，则创建数据库会失败。 打开MySQL Workbench,在中间的SQL 语句编写区域一个SQL语句，创建一个名为 DBTest的数据库，然后单击区域上方的闪电 按钮，执行SQL语句，如图19・28所示。，

|  |  |  |  |
| --- | --- | --- | --- |
| 可以在执行结果区域看到这个SQL语句 的执行结果，如图19-29所示。 | | | 图19-28创建数据库 |
|  |  |  |  |
|  | Jg|| .fetton Oiiput *糖t蠹/\*餐饌?囊篆懿續蠢,*  # T齢 m Actto-n  ，［‘3 1 15:^3:15 crests database DBTest | 1恐狀矿sffect&d | Dw 齢聲 Fetch  O/COO sec |

图19-29 SQL语句执行结果

|  |  |
| --- | --- |
| 图中的结果窗口显示SQL语句执行成功。単击左侧Schemas | Mavigstor ■ |
| 区域的刷新按钮，可以看到新的数据库dbtest已经显示在数据库列 | SCHEMAS *♦'辭* |
| 表中了，如图19-30所示。 | A, rfits? objects  dbtest |
| 19d3d2 | ” \_ sakih  ► -i-g sy-s |
| 删除数据库使用Drop Database语句实现，它的语法格式如下： |  |
| drop database db\_name | 图19-30显示新建数据库 |
| db\_name参数表示要删除的数据库名称。 |  |
| 现在在MySQL Workbench的查询窗口里输入SQL语句: |  |
| drop database dbtest |  |

执行该语句，可以看到结果窗口显示执行成功。在Schemas区域可以看到，该数据库已经 从列表里消失了。

要注意的是，要谨慎使用数据库删除功能，因为数据库一旦删除，它包含的所有数据库对 象也全部都删除了。

19o4表的基本操徘

数据库中的数据都存放在表里，表中的数据按照行和列来组织，一行数据是一条记录，表 的列对应记录的一个字段。

19.4J拿麗靈驚 一

MySQL支持的字段类型很丰富,主要包括以下几种:

（1 ）数值类型，如表194所示。

表19-1数值类型

|  |  |  |
| --- | --- | --- |
| 整数类型 | 字节数 | 范 围 |
| TNIYINT | 1 | 有符号：-128-127 ；无符号0-255 |
| SMALLINT | 2 | 有符号：-32 767-32 768 ；无符号"0〜65 535 |
| MEDIUMINT | . 3 | 有符号：-8 388 608-8 388 607 ；无符号:。〜！ 677 215 |
| INT | *4* | 有符号；-2 147 483 648-2 147 483 647 ；无符号：0〜4 294 967 295 |
| BIGINT | 8 | 有符号:-9 223 372 036 854 775 808-9 223 372 036 854 775 807 3  无符号：0〜18 446 744 073 709 551 615 |
|  | 字节数 |  |
| FLOAT | 4 | 単精度浮点数 … |
| DOUBLE | 8 | 双精度浮点数 |
| DECIMAL | 可变 | 高精度定点数 |
| 位类型 | 字节数 |  |
| BIT(M) | 1〜8 | BIT(1>BIT(64) |

（2）日期和时间类型，如表19『2所示。

a 19-2日期和时间类型

|  |  |  |
| --- | --- | --- |
| 日期和时间类型 | 字节数 | 范 围 |
| DATE | 4 | 1000-01-01 至 9999-12-31 |
| DATETIME | IIIIlli—1 | 1000-01-01 00:00:00 至 9999-12-31 00:00:00 |
| TIMESTAMP | 4 | 1970010101080001至2038年的某个时刻 |
|  | 3 | -838:59:59 至 838:59:59 |
| YEAR | 1 | 1901—2155 年 |

（3）字符串类型,如表19-3所示。

|  |  |  |
| --- | --- | --- |
| 表@3字符串类型 | | |
| 字符串类型 | 存 储 | 备 注 |
| BINARY | 255 | 定长 |
| VARBINARY | 65 535 |  |

续表

|  |  |  |
| --- | --- | --- |
| 字符串类型 | 、 存 储 | 备 注 |
| SMALLBLOB/BLOB | *L+2 B,L< 216* |  |
| MEDIUMBLOB | Z+3B,£<224 |  |
| LONGBLOB | *L+4 B,L<* 232 |  |
| TINYTEXT | (£+l)B,£<28 |  |
| SMALLTEXT/TEXT | *(L+2) B,L<* 216 |  |
| MEDIUMTEXT | *(L+3) B,L<224* |  |
| LONGTEXT | *(L+4) B,L<* 232 |  |

彳凱凰卷tin

在设计表的字段时，可以添加约束，常用的约束有：

•主键(primary key):设置某个字段为主键，主键能够唯一标识一条记录，所以主键不 能重复，一般也不能为空。

-非空(硕曲1)：设置某个字段不能有空值，如果添加空值为引发错误。

•唯一 (unique):设置某个字段的值唯一，即不能有重复的值。

•默认(default):设置某个字段的默认值，当没有为该字段设置值时，自动采用默认值。

⑥外键(foreign key):假设A表的某个字段B是唯一的，而C表的D字段的值引用了 B 字段的值，则这个引用关系称为外键。D字段的值只能来自于B字段已存在的值。

19.4.3劎選襄

创建表就是在数据库中增加一张新表，创建表的SQL语句使用Create Table实现，它的语 法格式是：

create table table\_name(

字段名字段美型［约束］,

字段名字段类型［约束

字段名字段类型［约束］

)

上述定义中，tablejiame是要创建的表名称，该表的名称不能与已存在的表重名。表的字 段定义语句在括号里，包括字段名称、字段类型，还可以加上约束。每个字段定义之间以逗号 分隔开。

例如，使用建表语言在数据库DBTest中创建一个用户表，取名为USR『S。之前已经使用 SQL语句创建了数据库dbtest,现在使用use dbtest语句,表示后面的SQL语句是针对dbtest 数据库发生的。具体的SQL语句如下所示。

use dbtest;

create table users (

UserID int primary keyz

UserName VARCHAR(20) unique^ UserPwd VARCHAR(20) not null

)

在上述语句中，首先使用use语句选择了 dbtest数据库，然后使用create table语句创建了 users表。该表有三个字段,分别是UserID. UserName和UserPwd,这三个字段都设置了约束, 其中UserID字段被设置为主键，而Us说Name的值是唯一的不能重复，而UserPwd字段的值不 能为空。这些约束也可以在创建完表以后再次添加或修改。

在MySQL Workbench中执行这段SQL语句,可以看到提示说语句执行成功,如图19-31 所示，其中绿色背景的勾号代表执行成功，红色背景的叉号说明语句执行失败。

Q 5 18:20:33 create table users; LteeriD 矚 primsty key, UserName VARCHAR；22； unique,... C

图19-31执行结果

在Schema导航器里可以看到已经列出了新建的表歹和表的列信息，如图19『32所示。

需要注意的是，在执行SQL语句时要使用use语句指定数据库，否则这段SQL语句将在 默认的数据库中执行。如果要多次使用同一个数据库，也可以将该数据库设置为默认数据库，

方法是选中该数据库，然后在右键菜单中选择"Set as Default Schema" o如图19-33所示。



dbtest '

T'WTate

卜 ' 曷

*. ...w to*

} l;||~,users f'

I CGlumris

I ♦ U-serlD

I UserMarne

j ♦ JJserFwd

*尊撃* indexes

► ;1|i Foreign Keys

*i'* Triggers

图19-32表信息

| J Obtest | fl 2 , B ■. |
| --- | --- |
| '矿 Tables | Load Spati:?.1 Data s |
| a | Set as Dsfa-alt Schains . |
| *如*b | Fi 11er co This Schema |
| V .• •• users |  |
|  | Schema InEpector |
|  | Tabl2 Data Import isss-d  Copy to Clipboai'd >  Send to SQL Edi tor \* |
|  |  |
| .''-Tng | Create Schein a... |
|  | Altar Schema.,. |
| Stored -TD£ |  |
| functions | Drop Schema... |
| :' sakHa | Sesi-ch Tabls Data... |
| S¥5 |  |
|  | Befresh All |
| W :涕W TA 私 *I EH* | *侧緇；部敍据餓針群謎魂潮\*\*熙湍飆麒喝展善緇憶頌曜錢験蛍溯院験密滅滥曉溺懿淡㈱瀚麥挤熙淡％* |

图19.33设置默认数据库

19b4q4 «!!«

表创建出来以后，还可以删除，删除表使用Drop Table语句，它的定义如下：

drop table table\_name

其中table\_name指的是要删除表的名称。

现在我们使用Drop Table语句将刚创建的表 典邙删除，语句如下：

use dbtest;

drop table users

上述语句中，首先使用use dbtest语句选择数据库(如果dbtest数据库为默认数据库，则可以省略use语句)，然后使用Drop Table语句将users表删除。在MySQL Workbench中执行 这条语句之后，可以看到执行结果为成功，然后表users马上从数据库中消失了。

要注意的是，如果表a的列被别的表b引用，那么为了数据的完整性，则必须先删除表b， 才能删除表ao

19o 5教鋸的矇作

19n5J

要想在数据库里保存数据，首先要将数据添加到表中。使用insert into语句添加数据，它 的定义如下：

insert into table\_name (fieldl r field2 f . •。….fieldn)

values (valuelz value2z e <, valuen);

例如，往users表里添加2行数据:

insert into users(userid^ usernamezuserpwd)

values (1^ ,xiaoMing 七"123456');

insert into users(useridrusernamefuserpwd)

values *{2f* 1xiaoHong ? r 5 6543211);

在MySQL Workbench中执行这段SQL语句,可以看到提示说语句执行成功。然后使用查 询语句查询表里的记录言

select \* from dbtest.users

UserID UserName UserPvvd '1""""""”…'''SSSing'''123456 " 加關&馆.""ills黑"

图19；4查询结果

执行查询语句，查询结果如图19・34所示。

可以看出，数据已经成功添加了。

10h5d2

更新数据是指对数据库中已经存在的数据记录值进行更改，是常见的数据操作之一。SQL 语句中使用update语句来实现对记录的修改。upd砒©语句的定义如下所示。

update table\_name

set fieldl = valuel7

field2 = value2 e field3 = value3 ....

where condition

在定义中,table name表示要更新记录的表名,field表示要更新数据的字段,value表ZK 新数值，condition指定要更新的数据满足的条件。

例如,将表users中UserID*为*1的用户名称改为“小华”,则可以使用以下SQL语句: update users set UserName = n 小华"where UserID *= 1;*

SELECT \* FROM dbtest.users;

执行SQL语句，可以看出，名称已经成功的更改了，…・…*检……*…

1 小华 1.2345b 如图 19-35 所7K。 12 xte-sHong S5432.1

图335更新结果

注意：要更新表中所有的数据,可以去掉where子句或者确保where子句里的条件包含了

所有的数据。例如:

update users set UserPwd = 1 654321 \* where 1 = 1;

删除数据指从数据库中将记录彻底删除，删除记录使用delete语句，它的语法如下:

delete from table name wher㊀ condition

其中？ table\_name是要删除记录的表名，condition是指要删除记录符合的条件。例如，从

槌哙表中将第1条数据删除，可以使用以下SQL语句:

delete from users where UserID = 1;

UserID UserbJame UserPwd

2 xiscHong 65^321

select \* from dbtest users;

其执行结果如图19『36所示。

图19.36删除数据后的结果

*i* 注意：如果要删除所有数据,可以去掉where子句,或者确保where子句里的条件包含

了所有的数据，例如：

delete from users where 1=1

当一个业务逻辑需要多个SQL完成时，如果其中某条SQL语句岀错，则希望整个操作都 退回。使用事务可以完成退回的功能，保证业务逻辑的正确性。

事务具有四大特性(简称ACID):

•原子性(Atomicity)：事务中的全部操作在数据库中是不可分割的，要么全部完成，要 么均不执行。

• 一致性(Consistency):几个并行执行的事务，其执行结果必须与按某一顺序串行执行 的结果相一致。

。隔离性(Isolation):事务的执行不受其他事务的干扰，事务执行的中间结果对其他事 务必须是透明的。

®持久性(Durability):对于任意已提交事务，系统必须保证该事务对数据库的改变不被 丢失，即使数据库岀现故障。

要注意的是，表的类型必须是innodb或bdb类型，才可以对此表使用事务。可以使用以下 SQL语句查询表的创建信息:

show create table table\_name

其table\_name表示表的名称。例如，查询U迥s表的创建信息，可使用以下SQL语句：

show create table dbtest s users

在建表信息中可以看出表的类型。

还可以使用以下语句修改users表的类型,例如:

alter table dbtest <. users engine = innodb

对于符合条件的表，可以使用事务操作，事务操作的语句包括以下三种：

° Begin：开启事务。

• Commit：提交操作。

° Rollback：回滚操作。

，19o6单表查询

数据的查询是指从数据库查询要求的数据记录，这个操作是数据库最常用的基本操作歹非 常重要。MySQL提供了丰富的数据查询方法，以满足不同的业务需求。

MySQL中使用SQL语句的SE心语句查询数据，它的格式为：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| select fieldl z field2 f . . . . . ^fieldn from table\_name  其中field表示字段名称,table\_name表示表名。  现在我们在数据库中添加1张employees (员工)表，用来示 例查询操作。它的表结构如图19-37所示。  其中，主键是emlD字段。可使用如下SQL语句创建员工表：  create table employees (  emID int(11) primary key, emName nvarchar (50) not nullf emBirthDate datef  emLevel int *(11) f*  emDepID int (11)); 图  为了示例数据库的查询，需要往员工表里添加若干数据，添加的!  表;19』员工表的数据 | | | | , ■ - - . employees  PK emID INT(ll) emName NVarchar(50)  emBirthDate Date emLevel INT(ll)  emDepID INT(ll)  19-37 employees 表的经  数据如表19・4所示。 | | 湘 |
| emID | emName | emBirthDate | emLevel | | emDepID | |
| 1 | 小明 | 1990/7/9 | 3 | | 1 | |
| 2 | 小红 | 1993/8/8 | 2 | | 2 | |
| 3 | 小花 ： | 1994/10/25 | 2 | | 3 | |
| 4 |  | 1990/11/30 | 3 | | 2 | |
| 5 | 小娟 | 1992/12/2 | 3 | | 3 | |
| 6 |  | 1988/3/5 | 5 | | 1 | |
| 7 | 老杨 | 1987/8/10 | 4 | | 2 | |
| 8 | iiwuw | iiiiuiimi— | 4 | | 3 | |
| 9 | 小周 | 1995/5/6 | 1 | | 1 | |

19.6.2最简暴資療 •

最简单的查询是查询所有的数据，它的语法格式如下：

select \* from table\_name

其中,table\_name表示要查询的表名,\*表示所有字段。

例如，从employees表中将所有的数据都查询出来，可以用以下语句：

use dbtest;

select \* from employees;

在MySQL Workbench里执行这条语句,可以看到查询结果,如图19-38所示。

可以看到所有的字段，所有的数据都查询出来了。

如果不需要所有的字段，可以显式地指明要查询的字段，多个字段之间以逗号分开，字段

|  |  |  |  |
| --- | --- | --- | --- |
| 的顺序也可以打乱。例如，从employees表中查询员工编号和员工姓名，那么SQL语句为 select emID7 emName from dbtest.employees;  其执行结果如图19-39所示。 | | | |
|  |  | ;emID | emhlame |
|  | 渤密証出詞s emLeyd eniCiepH) |  | 小明 |
| 1 小明 | *'' '"谥W蚤""*3…'”一…1"…”……" | ：；2 | 小红 |
| 2 小红 | 1933-08-08 *2 2* |  | 小花 |
| 3 小花 | 1994-10-25 2 3 |  | 小华 |
| T 小华 | 199£Ml-3a 3- 2 |  |  |
|  |  | h | 小婿 |
| <5 小娟 | .1992-12-02 *3- 3* |  |  |
|  | 1983-03-05 .5 1 | ,；6 |  |
| 7 寇杨 | 1987-08-10 4 2 | 7 | 寇杨 |
| 启 寇严 | 1987-Q2-14 4 3 | 祷 | 惡严 |
| ? 小周 | 1995-05^6 1 1 |  | 小周 |
| 图3-38查询结果 | | 图19・39 | 查询结果 |

技主意" ’―

1. 在SQL语句中,可以给表起个别名,在SQL表达式的其他部分可以使用别名来引用 这张表。例如:

select a o emlD^ a <, emName from dbtest „ employees a;

在这一语句中,a就是表dbtest.employees的别名。

1. 还可以使用as关键字给查询的列名起个别名，使得查询结果中的列名显示为该列的 别名。这在列名较长或较复杂时非常有用，例如：

select a . emID as 员工编号，a . emName as 员工姓名 from

dbtest o employees a;

在显示结果中，列名不再显示emID,emName,而是分别显示“员工编号”和"员工姓名”。

1S)応\* •

查询全部数据只适用于数据量很少的表，大部分表的数据量是非常巨大的，没有必要一次 性将全部数据都查询出来，而且消耗也会很大。现实中往往只需要查询满足条件的一组数据即 可，此时用到的就是条件查询。

SQL语句使用select语句和where子句进行条件查询,它的格式如下所示。

select fieldl f field2 f .fieldn

from table\_name

where condition

其中,conditon表示条件表达式。条件表达式由字段、运算符和数值组成。

（1 ）比较运算符

MySQL支持的比较运算符如表19-5所示。

select \* from employees where emDepID = 1

:;emID

它的查询结果如图194）所示。

（2）逻辑运算符

小明

小周

19S8-03-05

,1.§9.5-05-06

■emievd

emDepID

表19-5比较运算符

|  |  |
| --- | --- |
| 比较运算符 | 含 义 |
| > | 大于 |
| < | 小于 |
| = | 等于 |
| !=或者。 | 不等于 |
| >= | 大于等于 |
|  | 小于等于 |

例如，在employee表中查询人事部的员工，可以使用以下SQL语句:

MySQL支持的逻辑运算符如表19-6所ZK。

| emID | emName | emBirthDate | emlevel | esrDepID |
| --- | --- | --- | --- | --- |
| *2* | 小红 | '1993-08-Q8… | ""2."""" | *2* |
| 3 | 小花 | 1994-10-25 | 2. | 3. |
| 4 | 小华 | 1990-11-30 | 3 | 2 |
| 5 | 小娟 | 1992~12-02 | 3 | 3 |

图19-41查询结果

|  |  |
| --- | --- |
| 逻辑运算符 | 含 义 |
| AND (&&) | 逻辑与 |
| OR (ID . | 逻辑或 |
| NOT (!) | 逻辑非 |
| XOR | 逻辑异或 |

图19-40查询结果

表仲书逻辑运算符

使用逻辑与和逻辑或运算符，可以实现多条件联合的数据查询。

例如查询所有非rr部门

并且是1990年以后出生的员工，可以使用以下SQL语句:

select \* from employees

where emDepID != 1 and. emBirthDate > 11990-00-00!;

上述语句的执行结果如图1941所示。

（3）模糊查询

除了精确数据值的查询之外，还可以进行模糊 查询。模糊查询使用like关键字和通配符来实现。它 的定义如下：

select fieldl f field2 f

..fieldn from table\_name

where field like value

其中，vahi©表示匹配的字符串，一般包含了通配符，通配符可以匹配1至多个字符，这 样模糊查询的时候，只要满足了条件的记录都可以被查询出来。

like关键字支持的通配符包括和“％”，它们的含义如表37所示。

表 @7 通配符

通配符 含 义

\_ 匹配单个字符

例如，查询所有名字以“小”开头的员工，可以使用下面的SQL语句：

select \* from employees

where emName like ' 小 \_' ;

其查询结果如图19-42所示。

（☆注意：like关键字可以与逻辑非运算符（NOT） 一起使用，表示所有模糊查询结果之外 的记录。

（4）连续范围查询

•如果想查询某个范围内的数值，比如编号等于2、3、4的员工，那么可以使用范围来表示 查询条件。

MySQL中使用between...and…表示一个连续的范围,它的语法是:

select fieldl z field2 7 *。？* fieldn

from table\_name

where field between valuel and value2

表7K查询字段值在valuel和value2之间（包含valuel和value2 ）的记录,并且,valuel 必须小于等于value2,才是一个有效的范围。

例如，查询编号为2、3、4的员工，可以使用以下SQL语句实现：

select \* from employees where emID between 2 and 4;

其运行结果如图19-43所示。

JD

| ■emBirthDate mlevd emDepID | |
| --- | --- |
| 19.90-07-093 | ''.1 |
| 1.993-08-08 2. | *2* |
| 1994-10-25 2 | 3 |
| 1990-1.1-30 3 | 2 |
| 1992.-.12-D2 3 | 3 |
| 1995-05-06 1 | 1 |

<液哗蘭周

湖 小

| j emID emNsme | emBirtiiDate | erolevd | eriiDepID |
| --- | --- | --- | --- |
| 虹獎 商 | 1993--08'-08 | *"'"'2* | "2………"…"" |
| *'卩* 小花 | 1994-10-25 | 2 | 3 |
| 4 小华 | 1990-11-30 | 3 | 2 |

图19-43查询结果

图19-42查询结果

扌注意：连续范围可以与逻辑非运算符（NOT）一起使用，表示在该连续范围之外的记录。 它的定义格式如下：

select fieldl z field2 7 .

*h* fieldn

from table name

where field notbetween value 1 and value2

(5 )不连续范围查询(集合查询)

除了连续范围之外，还可以使用由关键字表示不连续的范围，也称之为集合。它的语法如 下所示。

select fieldl 7 field2 ..... F fieldn

from table\_name

where field in (value 1 f value2 f valuen)

在in关:键字后面的括号中定义了一个数据的集合，字段值在该集合内的记录会被查询出来。

例如，查询工号是2、4、6的员工记录，可以使用如下SQL语句：

select \* from employees where emID in (2z 46);

其运行结果如图19-44所示。

｛仃注意： 匚

集合查询可以与逻辑非运算符(NOT) 一起使用，表示在该集合之外的记录。它的定义格 式是：

select fieldl z field2 r fieldn

from table\_name

where field not in (value 1 z value2 z valuen)

(6)空值查询

可以使用is null关键字查询数据库里的空值记录，它的定义格式如下：

select fieldl f field2 z 。。。。。。， fieldn

from table\_name

where filed is null

它将查询出字段为空值(Null)的记录，要注意的是，空值(Null)和空字符串(»是不 同的，空字符串不符合空值的条件。

例如,我们先往employees表中添加2条出生日期为空的数据,然后再用空值查询将它查 询出来，SQL语句如下所示。

insert into employees (emID, emName) values (10,"迪丽热血”);

insert into employees(emIDfemName) values(16f" ;

select 大 from employees where emDepID is null

［片注意：两条SQL语句之间以分号(；)连接，它的运行结果如图19^45所示。

| emID emMeme | emBlrthDate | emlevd | er^DepID |
| --- | --- | --- | --- |
| ! " 2 小红 | 1993-08-03 | *2* | 2 |
| '4 小华 | 1990-11-30 | *3* | 2. |
| \ 5 宠張 | .1988433.-05 | *5* | 1 |

\ ernlD emMsme emBirthDste emlevd emDepID 何10 迪丽热血

16 小路

图19-45查询结果

图19-44查询结果

:気?注意：空值查询好像可以与逻辑非运算符(NOT) 一起使用，表示该字段非空的记录。

*电瞄顚携W*

它的定义格式如下：

select fieldl f fie 丄 d2’ *。……*.fieldn

from table\_name

where fieldis not null

1®擔w釐面茶蠶曩贓谒聂

有时需要将查询结果中重复的字段去掉，此时可以用distinct关键字来实现，只需要用di\* tinct关键字修饰重复的字段。它的格式如下:

select distinct fieldl f field2, . . . z fieldn from table\_name

例如，从员工表中查询所有的部门歹可以使用以下SQL语句：

select distinct emDepID from employees

其查询结果如图19・46所示。

要注意的是，如果在distinct后面跟多个字段，则只会去掉多个字段的重复组合值。例如， 以下查询语句查询了 emDepID和emName字段的非重复组合值。

select distinct emDepID, emName from employees

| ；emDepID |  |
| --- | --- |
|  | 小明 |
| ；|2 | 小红 |
|  | 小莅 |
| *\2* | 小华 |
| h | 小娟 |
| ""i— |  |
| |2 |  |
| ■js | 志严 |
|  | 小周 |
|  | 迪丽热血 |
|  | 小路 |

其查询结果如图19・47所示。

图19-47查询结果

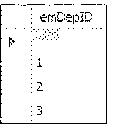


图19-46查询结果

1⑨応点霊念賣嵐

在查询语句时，可以使用聚合函数快速得到统计数据，MySQL支持5个聚合函数，它们 的作用如表19・8所示。

表19书聚合函数

|  |  |
| --- | --- |
| 聚合函数 | 含 义 |
| count(\*/field) | 査询记录总数 |
| max(field) | 查询某一列的最大值 |
| min(field) | 查询某一列的最小值 |
| sum(field) | 査询某一列的值的总和 |
| avg(field) | 査询某一列的平均值 |

使用聚合函数的格式如下所示。

select fieldl, field2 z , fun\_name from table\_name

其中,fiin\_name表示聚合函数。

例如，我们使用聚合函数count()查询employees表里的所有记录总数，可以使用如下SQL 语句：

select count (\*) from employees;

其查询结果如图19-48所示。

19q®b6 翁韻賛逾 ： 图性48查询结果

可以按照字段分组，表示此字段相同的数据会被放到一个组中。分组后，只能查询出相同 的数据列，对于有差异的数据列无法出现在结果集中。

分组经常与聚合函数结合使用，以对分组后的数据进行统计。

SQL语句使用group by关键字进行分组,分组查询的定义如下:

select fieldl, field.2 f . . . , fun\_name

from table\_name

group by fieldl r field2 f . . . [with rollup] [having condition]

其中：

° fun name：表示聚合函数。

® group by：表示要按照哪个字段进行分组。

• with rollup:可选项，表明是否对分组聚合后的结果再汇总。

。having：表示对分类后的结果再进行条件过滤。

例如，将employ笑表中的数据按照部门进行分组，计算每个组的人数，然后计算总人数。 可以使用如下SQL语句：

use dbtest;

select emDepIDf count (\*)

from employees group by emDepID with rollup;

其执行结果如图19・49所示。

例如，统计每个部门级别最高和最低的人，可以使用以下SQL语句：

select max(emLevel) as maximin(emLevel) as min,emDepID

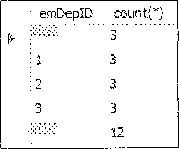
from employees group by emDepID;

图19-49查询结果

其执行结果如图19・50所示。

例如,统计平均级别大于2的部门,可以使用以下SQL语句:

select emDepID7 avg(emLevel)from employees group by emDepID having avg(emLevel) >2; 其执行结果如图19^51所示。

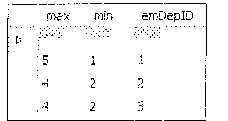


图19-50查询结果

|  | ；emDepID |  |
| --- | --- | --- |
|  |  | 3.0000 |
|  | 2 | *3MX* |
|  | 3 | 3,0800 |

图19-51查询结果

注意： ^

® where是对分组查询前对记录进行过滤。

® having条件判断是对分组后的记录进行过滤。

为了方便查看数据，可以对数据进行排序。在SQL中，排序使用order by »字。如果要 指定排序顺序，按从小到大的升序.排列使用asc关键字，按从大到小的降序排列使用desc关键字, 默认是按升序排列。排序的语法如下所示。

select \* from table\_name

order by fieldl asc | desc7 field2 asc | descf .

| emlD |  | emBifihDate |  | emDepID |
| --- | --- | --- | --- | --- |
|  | 迪丽热血 |  |  |  |
| he | 小路 |  |  |  |
| ：6 |  | 1983-03-05 | 5 | 1 |
|  | 小明 | 1990-07-09 | 3 | 1 |
| 9 | 小周 | 1995-05-06 | 1. | .1 |
|  |  | 1987M38-.1Q | 4 | 2 |
| '4 | 小华 | .1990-1.1-30 | 3 | 2. |
| *\2* | 小红 | 1993-08-08 | 2 | 2. |
|  | 寇严 | 1987-02-14 | 4 | 3 |
| ；5 | 小韻 | 1.992-12-02 | S' | 3 |
| ""切’ | 小花 | 1994-10-2,5 | 2 | 3 |

其中，order by后面是要排序的字段，如果有多 个排序字段，则先按照第一个字段的值进行排序，然 后对第一个字段值相同的记录再按照第二个字段排序， 依此类推。

图19^52查询结果

例如，对employee表中的记录按照部门和级别排 序，可以使用如下SQL语句：

select \* from dbtest«employees order by emDepIDz emLeveJ. desc;

其执行结果如图19^52所示。

在查询时，如果不希望将所有符合条件的数据一次性查询出来，还可以规定查询的记录数 量上限。规定记录数量使用limit关键字，要注意的是，这是MySQL对SQL语句进行扩展后 的关键字，不能用于其他数据库。

使用limit限制数量的定义如下所示。

select fieldlzfield2z .。.

from table\_name

limit startz count

. 其中，start表示起始记录号，默认是0, count表示共要查询多少条数据。

例如，从employees表中查询5条从第3条开始的记录，可以编写以下SQL语句：

| emID emMame emBirthDate emLevd emDepID | | | |
| --- | --- | --- | --- |
|  | 不花 | ""1994-10-25… | ""2"""""2"………""" |
| j4 | 小华 | 1990-11-30 | 3 2 |
|  | 小狷 | 199242<Q | *3* 3 |
|  | 寇张 | 1388-03-05 | 5 .1 |
| *\'7* |  | 19S7-0S-W | 4 *2* |

select \* from dbtest»employees limit 2,5 ;

图19-53查询结果

其查询结果如图19书3所示。

限制语句limit经常用于分页显示数据时的记录查询。

例如，已知每页显示m条数据，当前显示在第n页， 那么可以计算出总页数和第n页的数据。

(1 )求总页数：

首先查询出总页数pl,然后使用pl除以m得到p2；。如果能整除，则p2为总页数；如果不能整除，则p2+l为*总页数。*

1. 求第nt页的数据：

使用1 imit(n-即可。例如查询第4页的3条数据，可使用以下SQL语句：

select \* from dbtest o employees a limit 9, 3

1 ⑨思応 '

总结一下，单表查询的完整wlect语句定义如下所示。

select distinct fieldlf field2z 。.

from table\_name

where condition..

group by fieldl f field2 z » having condition

order by field2 r field2 f <.。o

limit startz count

这个定义里面，很多子句都是可选的，但是如果包含了这些子句，那么子句的顺序是固定的。

这些子句的执行顺序如下所示:

* from table\_name
* where condition...
* group by fieldl9field23.o.
* select distinct field 19 field2v •.
* having condition
* order by fieldl, field2,…
* limit start, count

，19s7多表査询

除了查询单表数据外，有时候需要将有关系的多个表里的数据同时查询出来，此时要用到 连接查询。

|  | employees |
| --- | --- |
| department | *PK.* era^'D |
| PK de© | Warchar(50) |
| deisms | Date |
|  |  |
|  | ernDep© |

现在我们往数据库中再添加一张departement

图 19-54 表 department 和表 employees 的结构

primary key,

(部门)表,与已有的employees表一起,用于

示例多表查询操作。其结构和关系如图19・54所示。

其中,department表的主键是delD字段,

employees表的主键是enilD字段。

使用如下SQL语句创建部门表言

create table department (deID int(11)

deName nvarchar(100) not null)

为了示例数据库的查询，再往部门表里添加若干数据，添加的数据如表19-9所示

>19^9部门表的数据

|  |  |
| --- | --- |
| deiD | deName |
| 1 | IT部 |
| 2 | 人事部 |
| 3 | 财务部 |
| 4 | 市场部 |
| 5 | 业务部 |

.19a7o1 { Gliner ]©m )

A表内连接B表，那么查询结果是A表和B表所有的符合连接条件的数据记录，舍弃不 匹配的数据记录。

内连接的语法格式如下所示。

select \* from table\_namel

|  |  |  | er(?Levd emDepID deiD deName | | |
| --- | --- | --- | --- | --- | --- |
| > i 1 | 小期 | 199CH37-a9 | 3 1 | 1 | ”浦…… |
|  | 小红 | 1993-08-aa | 2 2 | 2 | 人事部 |
| p | 小花 | 1994^10-25 | 2 3 | 3 | 财务部 |
|  | 小华 | 1990-11-30 | 3 2 |  |  |
| 1.5 | •1矚 | 1992-12-D2 | 3 3 | 3 | 财务部 |
| % 6 | 憲弓长 | 1988-D3-Q.5 | 5 1 | 1 | 賓部 |
| -j 7 | :卷杨 | 1987-08-ID | 4 2 | 2 | 人.事部 |
| 物 | *卷普* | 1987-02-14 | 4 3 | 3 | 财务部 |
|  | 小周 | 1995-05-06 | 1 1 | 1 | IT部 |
|  | 迪丽热血 | BEffl | 4 3 | 3 | 财务部 |
| j 16 | 小路 | 德勰 | *2* 3 | 3 | 财务部 |

inner join table\_name2 on table\_\_namel . f ieldl = table\_\_ name2 °fi㊀丄d2 .......

图19-55查询结果

以 employees 表 和 department 表 为例,

以部门相等的条件进行内连接查询，查询语 句如下所示。

select \* from employees e

inner j oin department d on e.emDepID = d.deID 其执行结果如图19『55所示。

1@o7n2 绑檐囊(©ut®r j©m )

外连接分为左连接和右连接，具体介绍如下：

(1 )左连接(left join)

A表左连接B表，则A表和B表匹配的数据会出现在结果中，外加A表中独有的数据， 未对应的数据使用null值填充。

例如，以部门相等的条件进行左连接查询，查询语句如下所示。

select \* from employees e

left join department d

on e o emDepID = d., delD;

它的查询结果如图19^56所示。

从结果可以看出? employees表中不符合条件的数据也被查询出来。

(2 )右连接(right join)

A表右连接B表，则A表和B表匹配的数据会岀现在结果中，外加B表中独有的数据， 为对应的数据使用null值填充。

例如，以部门相等的条件进行右连接查询，查询语句如下所示。

select \* from employees e right join department d on e.emDepID = d o delD;

它的查询结果如图19,所示。

|  |  | emBirthDate |  | emDepID | delD | defame |
| --- | --- | --- | --- | --- | --- | --- |
|  | 小明 | 1990-0709… | 上 | E | ~i | IT部 |
|  | 宠张 | 1988-G3-D5' | 5 | 1 |  | 貝部 |
| *景* | 小周 | 1995-)35-06 | 1 | 1 | 1 | 賓部 |
| *ii* | 小红 | 1993-D&-0S | 2 | 2 | 2 | 人爭部 |
| |4 | 小华 | 1990-1.1-30 | 3 | 2 | *2.* | 人事部 |
|  | 宼畅i | 1987-08-1.0 | 4 | .2 | *2* | 人箏部 |
| "3 | 小花 | .1994-10-25 | 2 | 3 | 3 | 财务部 |
| ;；5 | 小狷 | 1992'12-)32 | 3- | 3 | 3 |  |
| J8 | 走严 | 1987-02-14 | 4 | 3 | 3 | 财务部 |
| :ha | 迪丽惣血 | AMI | 4 | .3- | 3 | 财务部 |
| i 16 | 小蛋 |  | 2 | 3 | 3 | 财务部 |
| 「旧— |  | *VBA* |  | *-竺塑* | *諾囂鹽■* |  |

图19.56左连接查询结果

| j |  | emBr'Wate enfepID delD defeme | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| ► -； | 小明 | 1990-07-09 | '3… | '…1""""'— | *"1*'…" | 訂部 |
| *2* | 小红 | 1993-)38-08 | 2 | 2 | 2. | 人事部 |
| *3* | 小花 | 1594-10-2.5 | 2 | 3 | 3 | 财务部 |
|  | 小华 | I99G-11-3G | *W* | 2 | 2 | 人蚤部 |
| 才5 | 小娟 | 1992-12-02 | 3 | 3 | 3 | 财务部 |
| •布 |  | 1988-83-05 | 5 | 1 | 1 | 开部 |
| 如 | 表杨 | *IS37-Q8-1Q* | 4 | 2 | 口 | 人事部 |
| is |  | 1987-02-14 | 4 |  | 3 | 财务部 |
|  | 小周 | 1995-05'OS | 1 | 1 | 1 | 商部 |
| '；；1G | 迪丽热血 |  | 4 | 3 | 3 | 财务-部 |
| 15 | 小路 | ffiSH | 2 | 3- | 3 | 财韻. |
|  |  |  | BBB |  | 4 | 市场部- |
| | -pg |  |  | MH | *嚴輙* | 5 \_ | …业务都一 i |

图19-57右连接查询结果

从结果可以看出9 department表中不符合条件的数据也被查询岀来。

I 19O8 置麺教 ：

为了让开发者更好地使用数据库里的数据，每种数据库管理软件都提供了各自的内置函数。 MySQL软件也为一些常见操作提供了内置函数，开发者可以直接使用这些函数达到事半功倍 的开发效果。这些内置函数包括字符串函数、数学函数、日期时间函数和类型转换函数等。

字符串函数主要用于处理字符串。字符串操作是数据库的最常用操作，掌握字符串函数对 开发者的工作是大有裨益的。MySQL中常用的字符串函数包括：

(1 )拼接字符串函数 contat(strl?str29...?strN)

将strl9str29^9strN拼接为一个字符串，例如:

select concat ("itcast--" f emName) from dbtest. employees;

其执行结果如图19^58所示。

(2 )获取字符个数length(str)

length函数能获取字符串毗中所包含的字符个数。例如：

select length(? abc ?;

其执行结果如图19.59所示。

1. 截取字符串

截取字符串有3个函数：

• left(str, len):返回字符串sti•最左端的长度为len的字符串。

® right(str, len):返回字符串str最右端的长度为len的字符串。

® substring(str5 pos, len):获取字符串str的起始位置为pos?长度为len的字符串。

j csnc.at：ntcast-,ffi,,-emHame)

杵itG55<-小明

波希劳-小红

統％、.小彳它

itcasi-小华

前,亲房小娟

,g就■苞长

段E?-耄杨

htrsst-^ 严

：rtcast-小周

住航-迪丽热血

length Csbd}  
|'F…'…”…""'“

图19-59查询结果

取知-小路

图19-58查询结果

例如，从字符串uabcl23n中获取起始位置为2,长度为3的子字符串须可使用如下SQL 语句：

select substring (\* abcl2 3 *\ 2,3)* 其执行结果如图19・60所示。

1. 去掉空格

去掉空格也有3个类似的函数；

® Itrim(str)：返回去掉了左侧空格的str字符串。

° rtrim(str):返回去掉了右侧空格的str字符串。

。trim(str)：返回去掉了左右两侧空格的str字符串。

例如，将字符串fS abc “左右两侧的空格去掉，可以使用以下SQL语句:

select trim(n abc ");

其执行结果如图19・61所示。

| j sijbstringf.abcl2.3I#2.„r3}

j trim-r abc 1

> sbc

图19-61查询结果

I |bcl

图19-60查询结果

1. 去掉首尾端字符

trim(方向strl from str2):返回从某侧删除字符串strl后的字符串str,方向包括了 both, leading和trailing,分别表示两侧、左侧和右侧。

例如,从字符串"wwwltcastwww”中去掉首尾端的“w”字符,可使用以下SQL：

select trim(both \*w' from 'wwwltcastwww ?;

- 其执行结果如图19・62所示。

-(6)替换字符串

' 替换字符串使用replace(str, from\_str5 to\_str)函数，表示将字符串str中所有的子字符串 fn)m\_s加替换成to\_s加。例如，将字符串“我要学习iOS”中的“iOS"改成"Python”，可使用 如下SQL语句:

select replace (" «# 习 iOS、"iOS七 “Python®')“

其执行结果如图19-63所示。

? '对'磚捋斂扯爲tw耕g reuses：；我要学习汇热‘ ◎，，㈣血门｝

h 3 卜我要学习F削

图19-62查询结果 图19-63查询结果

(7)其他常用字符串函数 ^

MySQL还提供了其他常用字符串函数，如表19J0所示。

S 19-10其他常用字符串函数

|  |  |
| --- | --- |
| 字符串函数 | '功 能 |
| iower(str) | 将粉中所有字符转换成小写 |
| upper(str) | 将粉中所有字符转换成大写 |
| space(n) | 返回由n个空格字符组成的字符串 |
| repeat(str, x) | 返回字符串str重复x次后的结果 |
| insert(str, pos5 length, instr) | 将字符串str中从pos开始长度为length的子字符串替换成Eslr |
| strCmp(strl, str2) | 比较字符串strl和str2 :  。如果sirl大于str2,则返回1 5  。如果strl小于str2 9则返回-1 5  。如果strl等于str29则返回结果0 |
| 19q®d2數掌圖數 「 ’’  对数据的操作中，数值的处理也占比很大,我们有必要学会和灵活使用数值处理函数。例如，  血(n)可用于对数值n求绝对值。举例来说，对负数-25求绝对值，可使用以下SQL语句：  select abs(~25); 一 " *「H""*  其执行结果如图19-64所示。 土 p5  常用的数值处理函数如表19-11所示。 图1X4查询结果 | |

表19^11其他数值函数

|  |  |
| --- | --- |
| 数值函数 | 功 能 |
| mod(m,n) | 返回m除以n的余数，同运算符% |
| floor(n) | 地板函数，返回不大于n的最大整数 |
| ceiling(n) | 天花板函数，返回不小于R的最大整数 |
| round (n, d) | 求四舍五入值  。n表示原来的数  。d表示保留小数点位数，默认为0 |
| pow(x,y) | 返回x的y次嘉 |
| PK) | 返回圆周率 |
| rand() | 随机函数，返回。〜1.0之间的随机浮点数 |
| tnincate(x,y) | 返回将x截断为y位小数的数值 |

在开发中，往往要将数据库中保存的时间以不同形式显示出来，此时可以使用日期时间函 数进行处理。

1. 获取日期的子值，包括：

* year(date):返回date的年份(范围为1000〜9999 )。

° month(date)::返回date中的月份数值。

* day(date):返回date中的日期数值。
* week(date):返回date的日期是一年中的第几周。
* hour(time):返回time的小时数(范围为0〜23 ) °

° minute(time):返回time的分钟数(范围为0〜59 )。

® second(time):返回 time 的秒数(范围为 0~59 ) o

例如，返回日期“2017・06.30”的年份，可使用以下SQL语句： *塑S5SR.*

select year(f2017-06~30\*) ; *»「項*

其执行结果如图19^65所示。 图垃65查询结果

1. 日期格式化

日期格式化使用函数date\_format(date, format),其中format规定了日期的格式,它的可用 值包括：

•获取年％Y,返回4位的整数。

•获取年％y,返回2位的整数。

•获取月％m,值为1〜12的整数。

•获取日％d,返回整数。

•获取时％H,值为0〜23的整数。

•获取时％h,值为1〜12的整数。

•获取分％i,值为0〜59的整数。

•获取秒％s,值为0〜59的整数。

例如，返回日期“2017・06・30”的年月日时分秒数值，可使用以下SQL语句：

select date\_\_format(12017-06-30 17 :40:30 S 1%Y %m %d %H %i %s J);

其执行结果如图19-66所示。

1. 返回当前日期和时间

返回当前日期和时间的函数有3个，分别是：

* current\_date()：返回当前日期
* current\_time():返回当前时间
* now()：返回当前日期时间

例如，返回当前的日期、时间和日期时间：

select current\_date()7 current\_time()7 now();

其执行结果如图19-67所示。

|  |  |
| --- | --- |
| date JcrmSitCSO 17-06-30 17:40:30'.,!%¥ 擎期撃d 购 %s!}  2017 06 *3Q* 1740 30 | \ current„dateO £wrent\_ime0  ► 2D17-Q7^4 1.8.： 20:08 2D17-S37-Q418:20:08 |
| 图19-66查询结果 | 图19-67查询结果 |

(4)日期计算的函数

MySQL软件提供了日期计算函数，常用的如表1942所示。

|  |  |
| --- | --- |
| 表財 | |
| 日期函数 | 功 能 |
| to\_days(date) | 计算日期da馅与默认日期(0000年1月1日)之间相隔的天数 |
| from\_d ays(number) | 计算从默认日期(0000年1月1日)后间隔number天的日期 |
| datediffifdatel, date2) | 返回日期date］和date2之间相隔的天数 |
| adddate(date5 n) | 返回日期date加上11天后的日期 |
| subdate(date, n) | 返回日期date减去11天后的日期 |
| adddate(date, interval exp type) | 返回日期date加上一段时间后的日期，其中：  Exp :表示间隔时间的长度  Type :表示间隔时间的单位 |
| subdate(date, interval exp type) | 返回日期date减去上一段时间后的日期，其中：  Exp :表示间隔时间的长度  Type :表示间隔时间的单位 |

例如，计算日期“20174)7-03”后1天的日期，可以使用以下SQL语句：

select adddate(?2017-07-03 \*,interval 1 day);

其执行结果如图1948所示。

可以看岀，计算得到的日期是“20174)74)4” o

其实,adddate(date, interval exp type)和 subdate(date9 interval exp type)方法可使用加号(+ ) 和减号(-)来替代，使用起来更简便。例如，计算日期“20174)7・03”后1年2个月的日期， 可以使用以下SQL语句：

select ? 2017-07-03? + interval year\_month;

其执行结果如图说69所示。

j adddat£C2017-07^3sAnt£rvsl 1 day) j 52017-G7'-03s + interval ?1；25 yearjwnth

*雌* j 2017437-04 > ^(2018-09-03

图19-68查询结果 图369查询结果

从结果可以看出，计算得到的日期是“20184)9-03” o

adddate方法和subdate方法中type参数的可能取值如表1943所示。

« 19-13 type参数取值

|  |  |  |
| --- | --- | --- |
| type取值 | 表达式样式 | 含 义 |
| Year | YY | 年 |

|  |  |  |
| --- | --- | --- |
| 续表 | | |
| type取值 | 表达式样\*—―; | 含 义 |
| Month | MM | 月 |
| Day | DD | 日 |
| Hour | hh | 小时 |
| Minute | mm | 分钟 |
| Seconde | ss | 秒 |
| Year\_month | YY和MM之间以任意符号间隔 | 年和月 |
| Day\_hour | DD和hh之间以任意符号间隔 | 日和小时 |
| Day\_minute | DD和mm之间以任意符号间隔 | 日和分钟 |
| Day\_second | DD和ss之间以任意符号间隔 | 日和秒 |
| Hour\_minute | hh和mm之间以任意符号间隔 | 小时和分钟 |
| Hour\_second | hh和ss之间以任意符号间隔 | 小时和秒 |
| Minute\_\_second | mm和ss之间以任意符号间隔 | 分钟和秒 |
| MySQL中的类型转换函数有两个，具体如下：  cast(value as type)  convert(value, type)  其中，vMiie表示要转换的值，type表示目标类型，目标类型的取值可以是：  •二进制 binary o  •字符型char,可指定字符个数如char(10)o  •日期date。  •时间timeo  ®日期时间型datetime °  •浮点数decimalo  •整数 signedo  •无符号整数unsigned o  例如，将字符串”2345”转换成整数类型，可使用以下SQL语句：  select cast ("123,45" as signed) ; » j £351(\*123.45®as signed)  其执行结果如图19-70所示。 !► ：!123  图19-70查询结果  19- 9 MySQL Python | | |

在学会使用SQL语句之后，就可以在开发中使用Python »的模块与MySQL进行交互， 这是Python程序访问数据库的方法。

由于Python 3.x不向前兼容,导致我们在Python 2.7中使用的MySQLdb组件在Python 3 以后的版本无法使用。要在Python 3.x中连接MySQL,可以使用PyMySQL组件。PyMySQL 遵循了 Python数据库API 2.0规范。

19SJ PyMySQL

在Windows系统下,可直接进入命令行窗口,使用pip安装PyMySQL,如下所示。

pip install pymysql3

在Python文件中引入模块:

import pymysql

如果不出错，说明已经安装成功了。

1. 凱新2 PyilySQL

由于Python统一了 MySQL的访问接口,所以PyMySQL的常用对象和接口与MySQLdb 类似。

1. Connection 对象

Connection对象用于建立与数据库的连接。

可以使用connect (参数列表)方法来创建连接对象，其参数包括：

° host：连接的MySQL主机地址，如果是本机可使用?localhost\*o

* port：连接的MySQL主机的端口，默认是3306。
* db：数据库的名称。
* user：连接的用户名。
* passwd：连接的密码。
* charset：通信采用的编码方式，推荐使用utfB。

例如，使用SQL语句创建一个本地连接：

conn = pymysql.connect(

host= ?localhost',

user='root1z passwd='12 345 6 L db=*斗*dbtest5 z charset= s utf8 ?)

Connection对象的常用方法包括:

* closeQ:关闭连接。

@ commit():提交当前事务。

* rollback。：回滚当前事务。
* cursorQ:创建并返回游标。

2. Cursor

顷S成称为游标，它负责执行SQL语句。可以使用连接对象的印球双方法来创建，例如： cursor = conneet <. cursor ()

游标对象的常用方法包括旨

• close():关闭游标对象。

° execute(operation [, parameters ])：执行 SQL 语句,返回受影响的行数。

。fetchoneO：执行查询语句时，获取查询结果集的当前行数据，返回一个元组。

° next():执行查询语句时，获取当前行的下一行。

。fetchallQ：执行查询时，获取结果集的所有行，一行构成一个元组，再将这些元组装入 一个元组返回。

® scroll(value[,rnode]):将行指针移动到某个位置。mode表示移动的方式。

° mode的默认值为『elativ饥 表示基于当前行移动到value5 vain©为正则向下移动，value 为负则向上移动。

Q mode的值为absohrts表示基于第一条数据的位置，第一条数据的位置为0。 游标对象的常用属性包括：

° rowcount：只读属性，表示最近一次execute()执行后受影响的行数。

° connection：获得当前连接对象。

彳凱凱芻PyMySQLwfl

现在我们用一个示例演示如何使用PyMySQL来连接和操作MySQL数据库。在Pycharm 中新建一个文件,输入以下代码：

import pymysql

#连接数据库

connect = pymysql.connect(

host=1 localhost, ’

user=1 root17 passwd='123456 \* z db=1dbtest \* z charset="utf8 \*)

#获得游标

cursor = connect.cursor ()

#插入数据

sql = "insert into employees (emID, emName, emLevel, emDepID) values (%d, 1%s1z %d7 %d)"

data = (15,'小园3, 3)

cursor.execute(sql % data)

connect。commit()

#修改数据

sql = "update employees set emName = 1%s' where emID = %d"

data = ( 1 小丸子 1 z 15)

cursor.execute(sql % data)

connect.commit()

#查询数据

sql = "select emIDr emName from employees where emDepID = 3° cursor.execute(sql)

for row in cursor.fetchall():

print (° 员工工D:%d 姓名:?%sln % row)

print ('财务部\_共有%d个员工' % cursor»rowcount)

#删除数据

sql = "delete from employees where emID = %d limit %d° data = (15, 1)

cursor o execute(sql % data) connect.commit()

print ('共删除％d 条数据'% cursor . rowcount)

#事务处理

sql\_one= "insert into employees (emIDf emNamef emLevelz emDepID) values (16,，小涵 \2,2)"

sql\_two = "update employees set emName = \* 小路! where emID = 16" sql\_three = "update employees set emDepID = 3" try:

cursor.execute(sql\_one) cursor.execute(sql\_two) cursor.execute(sql\_three) except Exception as e:

#添加\_条数据

#修改这条数据

#再次修改

#事务回滚

#事务提交

connect.rollback() print ("事务处理失败"z e) else :

connect.commit() print ("事务处理成功")

#关闭游标,关闭连接

cursor.close() connect *。*close()