

数据库基础、使用 C 语言构建一个数据库

文章目录

一、数据库

1.什么是数据库

2.数据库的核心功能是什么

3.数据库的核心组件有哪些

4.数据库发展和展望

二、使用 C 语言实现一个简单的数据库

三、使用 C 语言连接数据库

四、SQL 语言和 MySQL 教程

1.SQL 是什么

1) DDL - Data Definition Language, 数据定义语言

2) DML - Data Manipulation Language, 数据处理语言

3) DCL - Data Control Language, 数据控制语言

2.常用数据库访问接口简介

ODBC

JDBC

ADO.NET

PDO

3.常用数据库大汇总（附带优缺点）

1) Oracle

优点:

缺点:

2) SQL Server

优点:

缺点:

3) MySQL

优点:

缺点:

4) Access

优点:

缺点:

5) DB2

优点:

缺点:

6) PostgreSQL

优点:

缺点:

4.MySQL 是什么? 它有什么优势?

MySQL 的特点、优势

1) MySQL 是开放源代码的数据库

2) MySQL 的跨平台性

3) 价格优势

4) 功能强大且使用方便

5. MySQL 适用于哪些场景?

1. Web 网站系统

2. 日志记录系统

3. 数据仓库系统

4. 嵌入式系统

6. 学 MySQL 前, 需要了解这些数据库专业术语

7. MySQL 客户端和服务器架构 (C/S 架构)

8. 明白了 MySQL 内部结构才能成为高手!

1. 连接层

2. SQL 层

3. 存储引擎层

4. 文件系统层

9. 如何学习数据库 (新手必看)?

10. 如何高效的学习 MySQL

1) 培养兴趣

2) 及时学习新知识

3) 多练习、多操作

4) 多编写 SQL 语句

5) 通过 [Java](<http://c.biancheng.net/java/>) 等编程语言来操作数据库

6) 数据库理论知识不能丢

11. 启动 MySQL 服务的两种方式 (图解)

通过计算机管理方式

通过命令行方式

12. 结合实例, 彻底搞懂数据库设计的三大范式

1) 第一范式

2) 第二范式

3) 第三范式

4) 反范式化

1) 范式化

2) 反范式化

一、数据库

1. 什么是数据库

数据库就是英文的“database”翻译来的, data + base, 顾名思义就是数据的根源, 数据的基础。那么为什么要有数据库呢, 数据库首先是个计算机软件, 在所谓数据库诞生之前, 常用方法可能是程序员自己写一个小程序来完成数据处理分析这样的工作。

伴随着计算机的普及, 越来越多的场景开始使用计算机, 产生了越来越多的数据, 也催生了越来越多的数据分析需求。为了降低数据分析的门槛, 让更多人能够更方便高效地管理分析数据, 工程师们就打造了一种专门的软件来帮助人们对数据进行合理的存储以提高存取效率, 提供易用的接口和丰富的分析算法以方便使用, 集成有效的管理工具以提高数据安全性等等, 这就是数据库, 也被称为数据库管理系统 (DBMS, Database management system)。

数据库是一整套数据管理体系，包括数据存储的模型、数据组织的架构、数据分析的算法、数据管理的工具以及数据访问的接口等等。

举个例子，粮仓。如果你有 1 亩 3 分地，产的粮食刚刚够一家人吃，吃不完的自己找个缸就放下了，这个缸也只需要方便自己家人使用就行了。随着你种的地越来越多比如 1 万亩地，生产的粮食根本吃不完，那就必须修建一个专门用来存放粮食的仓库，同时还要方便不同的商家来拉粮食，为了保证粮食存放的安全和效率，就必须对粮仓进行特殊的设计和處理，比如恒温恒湿、自动喷淋、传送系统等等。数据库也是类似的道理。

数据库起源于阿波罗登月计划，因为需要大量的数据分析人员对大量的数据进行分析，就不得不开发一款能够方便更多人使用的数据库管理分析软件。确实是人类当时的灯塔，研发出了多少好东西，不得不给 NASA 的工程师们点个赞。

2. 数据库的核心功能是什么

数据库会根据应用场景的不同而分为不同的类别，比如最经典的分类 OLTP（在线事务处理）和 OLAP（联机分析处理）。举个例子，你每天要使用信用卡支付来坐地铁，买午餐、买饮料、上淘宝购物等等，这每一笔交易都需要后台数据库准确地记录下来，这个数据库就是 OLTP 类型。

你也会通过系统去查询你上个月的消费情况，系统会根据你上个月的交易数据做个汇总发给你，并告诉你吃饭花了多少、交通花了多少、娱乐花了多少等等，支持这个场景的就是 OLAP 类型。

OLTP 主要处理短小的事务，要求事务吞吐量很高，因为每个人每天可能要支十几次，但每次需要处理的数据量比较小；而 OLAP，每个人可能每个月只用一次，但是每次要处理的数据量相对比较大，而且计算比较复杂。

近年来，伴随着人工智能、物联网、边缘计算等数字化场景的兴起，数据库的功能也产生了更多的分类，如 HATP（同时能够处理 OLTP 和 OLAP 的场景）、流式数据处理、时序数据处理、非结构化数据处理、跨平台数据处理、多模态数据处理等等。如何理解这些分类呢？

类似于不同功能的汽车，有货车，有客车，有 MPV，有 SUV，有皮卡，有燃油车，有新能源车等等。车的核心功能是一致的，只是为了适应不同的场景和需求，不同的车会有不同的架构设计和调教，如此而已。

那么数据库应该有哪些核心功能呢？

首先，数据库、数据库，必须要把数据保存下来。要把数据按照合理的格式，安全保存在可持久化的存储介质里面，要保证数据的正确性、完整性和安全性。这是所有数据系统最核心的功能。换句话说，把数据交给数据库，数据库要保证数据不丢、不错。这个是最最起码的要求。正如粮仓，不能粮食存进去都发霉了，被耗子吃了。

其次，数据库要尽可能提高数据存取效率。要用更有效率的方式存储数据，让数据存储得更

快，更易于使用者理解，更方便上层业务的使用。查询数据时效率更高，更快给出结果。就像有人来送粮食入库，要快速地称重、烘干、质检、打包、入库，不能让人家等一礼拜。有人要买小麦，有人要买玉米，必须按照要求快速找到相应的存放地点把粮食交给粮商。

再次，数据库要提供丰富的数据分析算法，尽可能把跟数据密切相关的计算在数据库中完成，减少数据传输的开销，减轻上层业务逻辑的计算压力。就像粮库要提供完善的粮食处理措施，比如称重、烘干、打包、品质分级等，方便粮食交易。

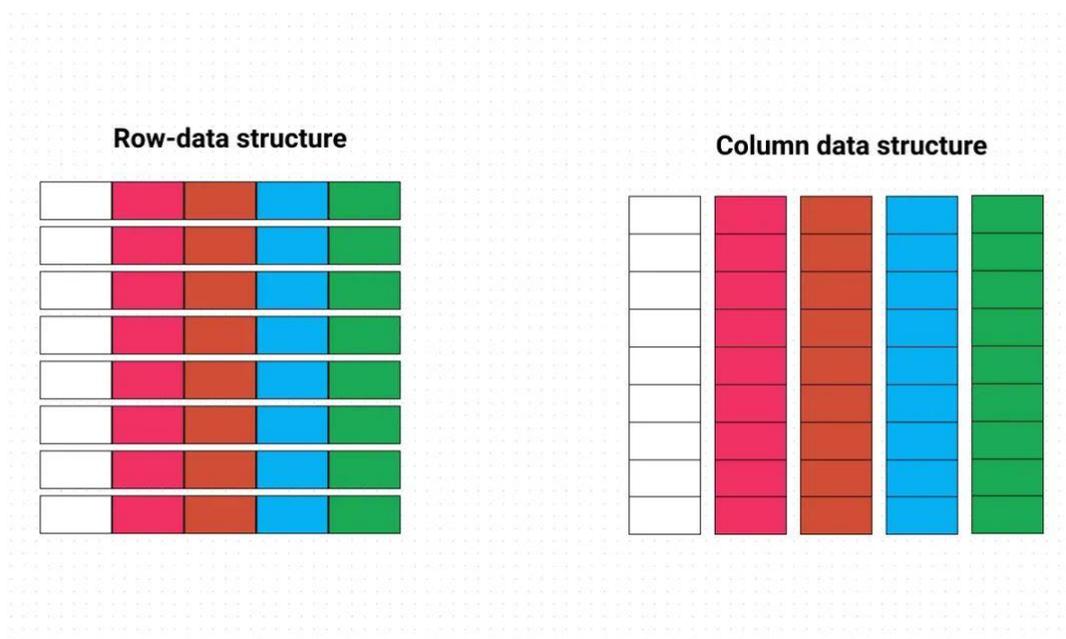
最后，数据库要提供易于使用的接口，降低数据分析人员的使用门槛，能够支持各种数据分析工具，让使用数据更加方便。就像粮库要有方便的停车场、清晰的指示牌、专业友好的工作人员等。

3.数据库的核心组件有哪些

为了实现这些核心功能，通常数据库会包括以下核心组件：

a. 存储管理

数据用什么样的方式来组织、存储，是 **key-value** 还是关系型，是按行存还是按列存，支不支持压缩，支不支持删除和修改，支持什么样的数据类型和存储接口，**Posix** 还是对象存储。是否要支持计算存储分离，是否要支持分布式存储，是否支持事物处理，是否支持多副本，采用什么算法来加速数据的检索（索引）等等。存储管理是数据库的核心组件，解决了存储管理问题，数据库的问题就解决了一半了。

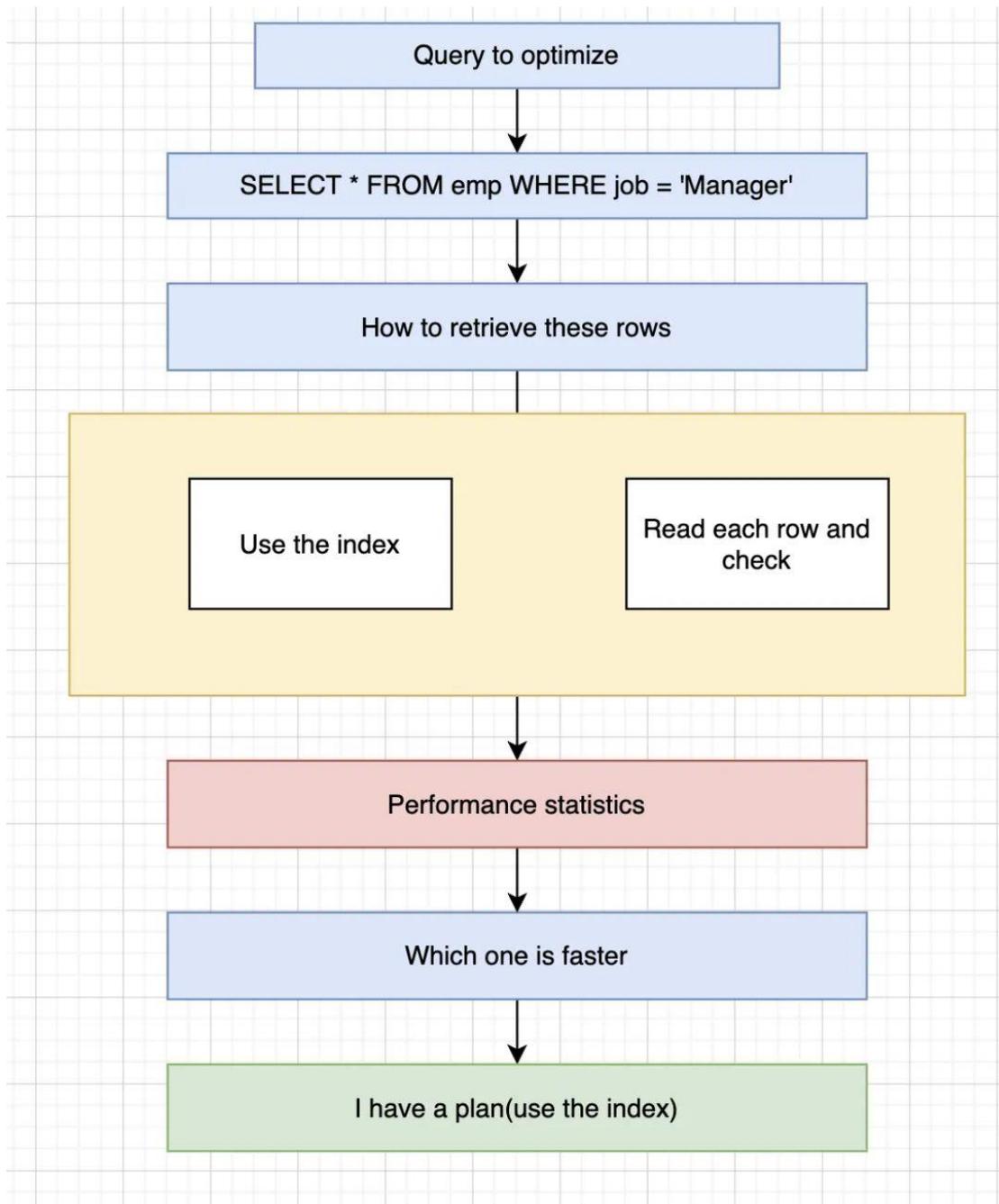


b. 查询优化器

要提高数据查询的效率，数据库必须找到一条最优化的执行路径，比如，查询时是否需要使用索引，如果有多个索引，应该选择哪一个，如果数据分布在不同的存储单元（表、集合等）

里，应该按照什么顺序来访问效率最高等等。优化器面对的问题可能是一个极其复杂的路径规划问题，需要它在很短的时间里计算出最优路径，需要大量核心优化算法。属于数据库中复杂程度最高的部分。

举个例子，你要带着全家人，包括老人、小孩一起从上海去海南旅行，要制作一个性价比最好、家人满意度最高的计划，那么在计划时需要考虑哪些因素呢，首先，怎么去，是开车去，还是火车去，还是飞机去。开车，路上要花多久，中间需要休息几次，你和太太有没有时间，老人孩子是不是受得了，汽油费用，过路费用；飞机，怎么去机场，行李有多少，带不带的下，机票有没有打折，下了飞机怎么办等等。住什么酒店，去什么景点，老人喜欢去人多的人文景观，太太喜欢安静的地方和方便购物的地方，小孩喜欢有游乐场的地方，要不要酒店+景点一起订，会不会有优惠，要不要租车，租什么车…说到这里，是不是可以体会一个查询优化器需要考虑的问题有多少？

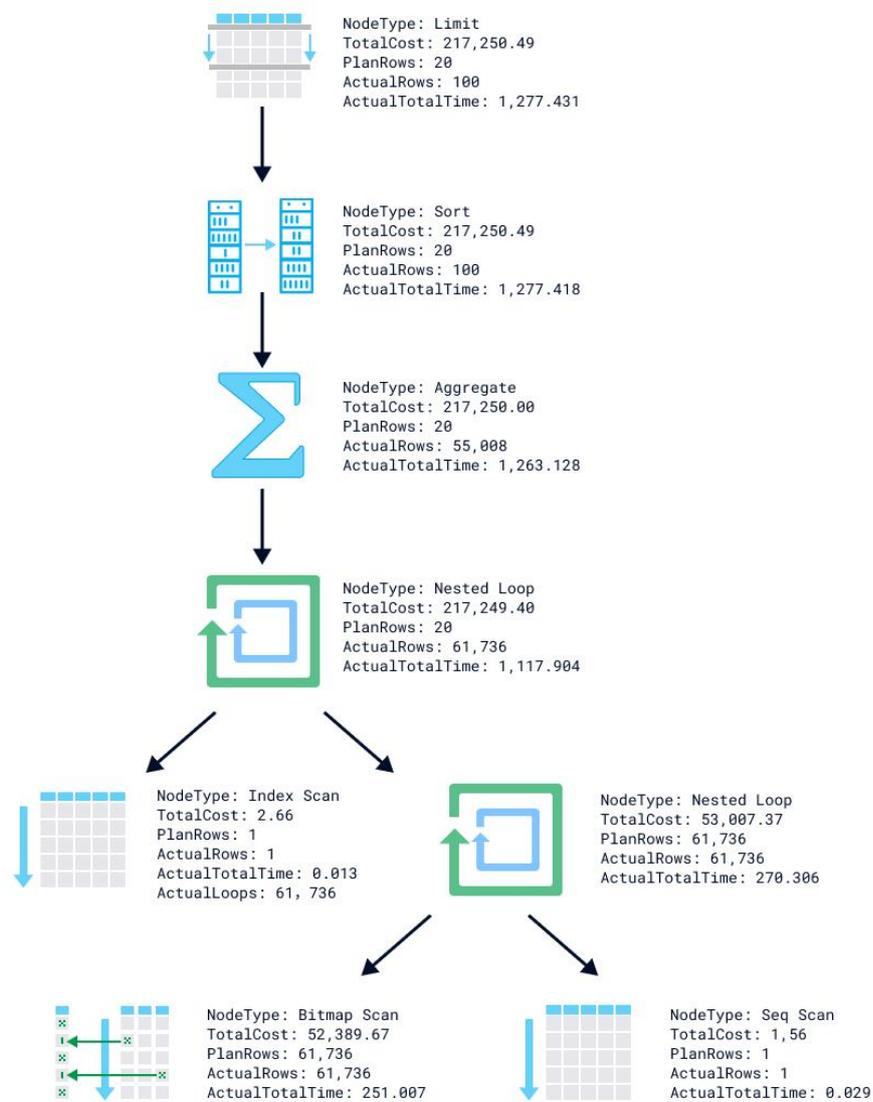


当然，这部分工作可以有相对简单的实现（基于规则），比如太太说了，时间确定、飞机来回、五星酒店、带私人沙滩。这样计划就会简单很多，也可以复杂到难以想象（基于机器学习、基于实际开销等等），太太说你全权负责，具体时间不确定，大概在 8 月-9 月，要少花钱多办事，多做调研，找一个最优方案。那么做这个计划就会非常复杂，需要的支持决策信息就会非常多。这样做出来的决策大概率相对会优化，比基于规则实现的计划能适应更多场景。

c. 执行模块

优化器做好了执行计划后，接下来就会有执行的模块按照执行计划对数据进行相关的计算，

包括数据的存取、常规的加减乘除、排序、平均值、哈希，也会包括一些机器学习的算法，数据的压缩/解压缩，最后将计算完成的结果返回给客户端。



d. 内部管理和调度

数据库要正常的工作，还会需要一些内部协调管理的模块，比如，内存和存储同步，存储空间整理，元数据管理，集群状态检测，容错和故障恢复等。

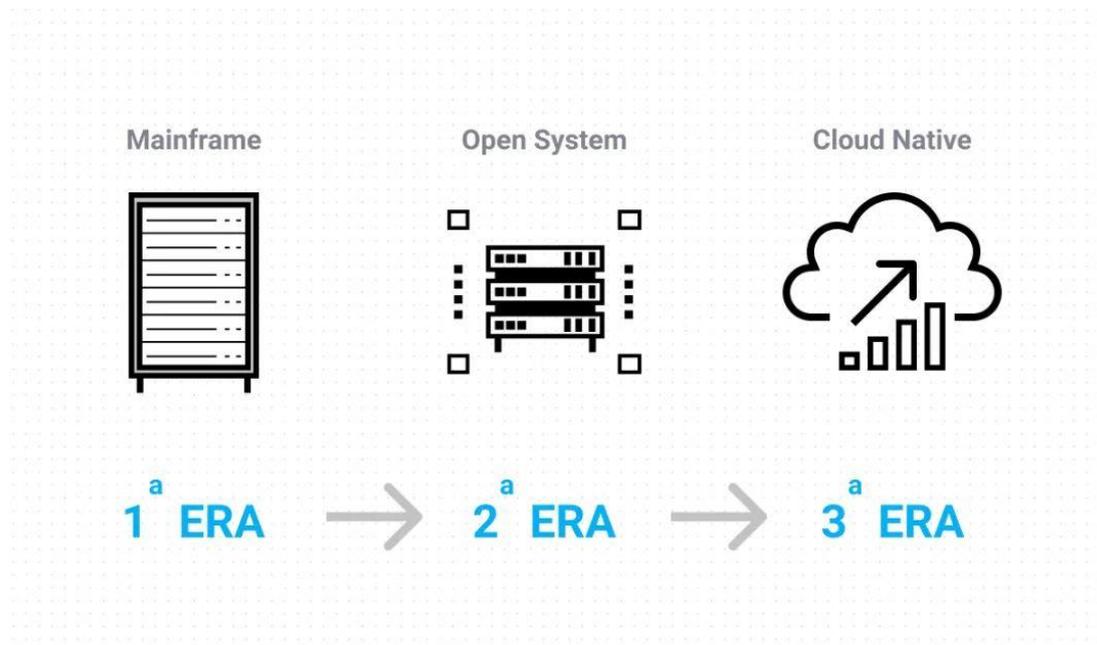
e. 管理工具和接口

为了提高易用性，数据库都需要提供一套管理工具，比如备份/恢复、状态检测、运行时监

控、资源隔离、权限管理、安全审计、自定义接口、各种数据访问接口等。

4.数据库发展和展望

数据库的发展是伴随着计算机体系架构的发展而不断演进的，从主机，到个人电脑+网络（x86），到现在的云服务，数据库也经历了一系列的演化历程。



a. 主机时代

最初的计算机和数据库只是在航空航天、军事领域使用，只需要支持专业的数据分析人员进行数据分析。到了上世纪 70 年代末，伴随着计算机进入更多的商业场景，产生了大量的数据分析的需求，数据库就需要面对更为普遍的用户需求。在 IBM 最早发布的关系型数据库的论文中，最强调的一点就是希望能够让数据库的用户不用再去操心数据应该如何存储和组织，而能够高效率使用这些数据进行分析。

为了方便用户的使用，SQL（结构化查询语言）被定义了出来，按照这样的语法，数据库用户只需要关注数据该如何分析，不需要关注底层的数据分布和存储等。

为了要支持大量用户的并发数据操作，数据库事务特性被定义了出来，保证在并发的数据操作下，用户能够看到符合业务逻辑的数据内容。

为了保证数据库的高效率和安全性，数据库重做日志（事务日志）被设计出来，包括当前数据库中经常出现的一系列概念，比如回滚日志（Undo Log）、提交日志（commit log）、检查点（checkpoint）等等。

主机时代由于硬件成本极其昂贵，不论是存储、内存还是 CPU 资源，相对来说都很稀缺，那么数据库在设计和使用上就会采用各种算法和架构来降低对内存的使用，减少数据的冗余，提高数据的检索效率，因此各种数据索引类型，功能强大的查询优化器，数据缓存算法等在

数据库中得到了极大的发展。同时在使用数据库时,也要对数据进行各种复杂的模型设计(3范式模型,星型模型,雪花型模型等等)以降低数据的冗余程度,当然这样也会增加数据库应用的开发难度。

b. x86 时代

伴随着 x86 服务器的广泛使用和网络技术的发展,把 N 台 x86 服务器通过网络组建成一个集群,利用这个集群的计算、存储能力来取代昂贵的主机也就更加具有性价比。在这种趋势下,也就设计出了各种能够使用集群能力的分布式数据库系统,这些系统的核心思想就是把数据分散在不同的节点上,利用多个节点的计算和存储资源提高对数据的存储和分析能力。在分布式的处理架构下,数据一致性协议、多副本机制、高可用机制、数据分片机制、扩容/缩容机制等等也都成为了分布式数据库必须要设计和解决的问题。

在 x86 时代,由于硬件成本的大幅下降,用户更多关注数据分析的灵活性和交付的效率。因此,使用数据库时更多会关注如何加快数据分析的过程、如何让数据更易于人类理解,而不需要为了降低数据的冗余而进行复杂的模型构建。

c. 云时代

随着技术的进一步发展,通过把传统硬件虚拟化/容器化等技术,提高硬件资源的使用效率,降低生产运维成本的云服务被越来越多企业采用。为了更好地适应云服务的技术体系,数据库也设计出了相关的云特性,比如存储计算分离、弹性伸缩、微服务化、跨域数据同步等等。

云时代,用户更加关注数据分析的效率和投入产出比,更加关注产品是否能够提供便利的一体化数据处理服务,让业务开发者能够更加专注于业务本身,而数据库服务也在朝着标准化云服务的方向不断演进。

d. 展望

不同的数据库架构和部署方式不是一个简单的迭代和取代的关系,而是在很长一段时间里会同时存在并且逐步迭代的过程。时至今日,依然有不少金融机构会选择使用在主机上的数据库产品,只是新的业务和场景非常有限。而基于 x86 服务器的数据处理产品,还是当前企业数据库的主流选择。与此同时,云数据库的市场份额也在逐步增长和扩大。采用何种数据库产品要根据自身的业务需求来决定,合适的就是最好的。当然从技术演进的方向上看,云技术(包括公有云和私有云)会是大势所趋,因为云能够提供更高的效率。

数据库作为信息产业的三大基础技术(还有芯片和操作系统)之一,在相当长的时间里,不论从资本还是技术方面都非常火热,国内近几年来也出现了相当多优秀的数据库产品和企业。在人类迈向数字化文明的进程中,必定会产生越来越多的数据,也需要从数据中挖掘出更多的价值,而数据库作为承载数据的核心,也必将持续发挥重要作用。有幸一直在从事这个领域的工作,期待与广大同仁一道为人类数字化技术的进步贡献力量。

二、使用 C 语言实现一个简单的数据库

这边推荐几篇博客:

(C 语言) 数据库简单实现

[小项目]c 语言实现数据库操作(低仿)

c 简易实现数据库

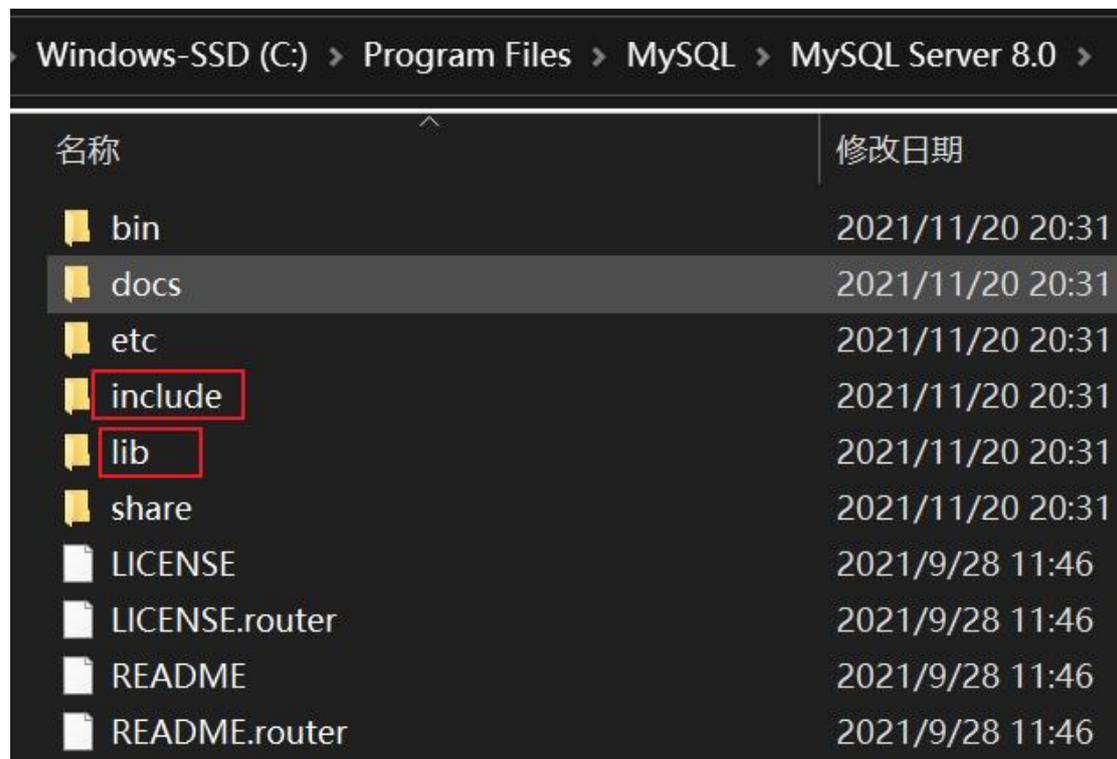
三、使用 C 语言连接数据库

当然大家基本不会使用自己写的数据库，一般都是使用专业的数据库。

下面我们看看怎么使用 C 语言来连接数据库。

先安装一下 MySQL，可以参考这篇博客：[超级详细的 mysql 数据库安装指南](#)

我的安装路径为：C:\Program Files\MySQL

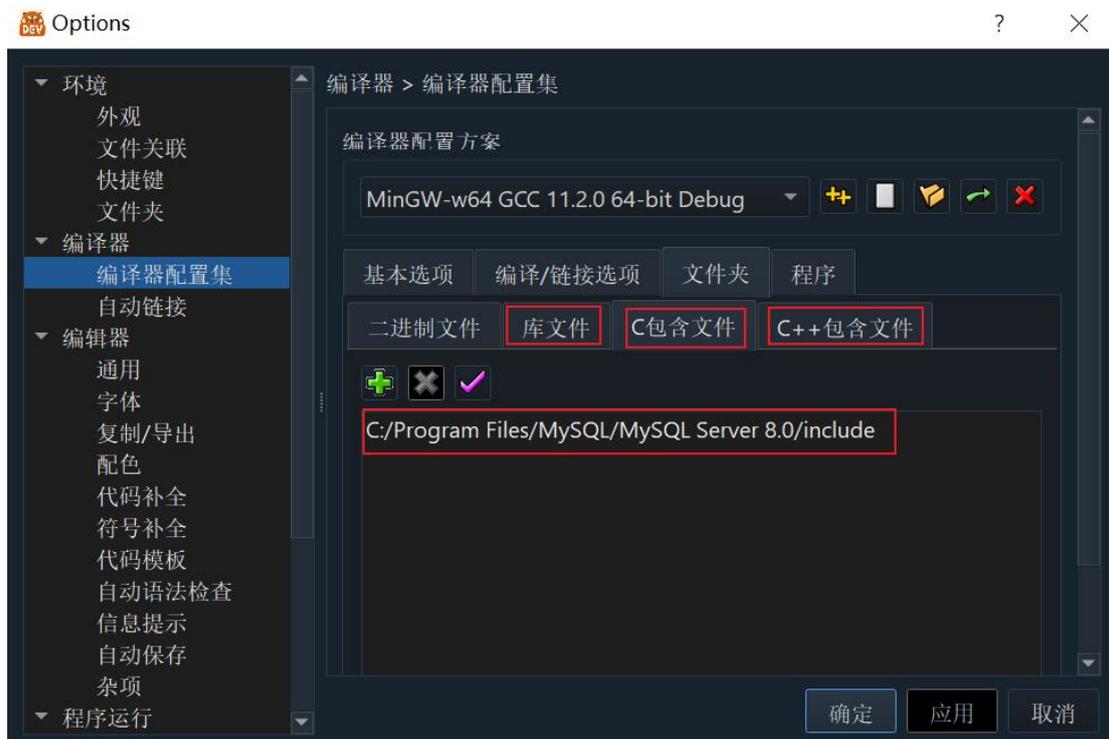


include 目录中就包括我们 C 语言连接数据库所需要的头文件。

vs-SSD (C:) > Program Files > MySQL > MySQL Server 8.0 > include >

名称	修改日期
mysql	2021/11/20 20:31
openssl	2021/11/20 20:31
errmsg.h	2021/9/28 11:46
field_types.h	2021/9/28 11:46
my_command.h	2021/9/28 11:46
my_compress.h	2021/9/28 11:46
my_list.h	2021/9/28 11:46
mysql.h	2021/9/28 11:46
mysql_com.h	2021/9/28 11:46
mysql_time.h	2021/9/28 11:46
mysql_version.h	2021/9/28 14:04
mysqld_error.h	2021/9/28 14:06
mysqlx_ename.h	2021/9/28 13:49
mysqlx_error.h	2021/9/28 13:49
mysqlx_version.h	2021/9/28 13:49

要想使用 C 语言来连接 MySQL 数据库，我们需要将头文件的路径和库文件的路径加入进来。



可以看看下面的文章来看看一些具体的操作：

C 语言操作 mysql 范例（增删查改）

c 语言连接数据库以及对数据库操作

四、SQL 语言和 MySQL 教程

1. SQL 是什么

SQL 是一种操作数据库的语言，包括创建数据库、删除数据库、查询记录、修改记录、添加字段等。SQL 虽然是一种被 ANSI 标准化的语言，但是它有很多不同的实现版本。

ANSI 是 American National Standards Institute 的缩写，中文译为“美国国家标准协会”。

SQL 是 Structured Query Language 的缩写，中文译为“结构化查询语言”。SQL 是一种计算机语言，用来存储、检索和修改关系型数据库中存储的数据。

SQL 是关系型数据库的标准语言，所有的关系型数据库管理系统（RDBMS），比如 MySQL、Oracle、SQL Server、MS Access、Sybase、Informix、Postgres 等，都将 SQL 作为其标准处理语言。

此外，SQL 也有一些变种，就像中文有很多方言，比如：

微软的 SQL Server 使用 T-SQL；

Oracle 使用 PL/SQL；

微软 Access 版本的 SQL 被称为 JET SQL（本地格式）。

SQL 的用途

SQL 之所以广受欢迎，是因为它具有以下用途：

允许用户访问关系型数据库系统中的数据；

允许用户描述数据；

允许用户定义数据库中的数据，并处理该数据；

允许将 SQL 模块、库或者预处理器嵌入到其它编程语言中；

允许用户创建和删除数据库、表、数据项（记录）；

允许用户在数据库中创建视图、存储过程、函数；

允许用户设置对表、存储过程和视图的权限。

SQL 简史

1970 年，IBM 的 Edgar Frank “Ted” Codd（埃德加·弗兰克·科德）博士描述了关系型数据库的模型，他因此被称为“关系型数据库之父”。

1974 年，IBM 希望把 Codd 的想法变成现实，着手开发一款名为 System R 的数据库，并研发出一套结构化查询语句 SEQUEL，这就是 SQL 的雏形。System R 数据库于 1978 年第一次发布，用于科研和实验。

1979 年，Oracle 公司首先提供商用的 SQL，随后 IBM 公司也在 DB2 数据库中实现了 SQL。

1986 年 10 月，美国 ANSI 采用 SQL 作为关系型数据库管理系统的标准语言，紧接着国际标准组织（ISO）也将 SQL 采纳为国际标准。

1989 年，ANSI 发布了 SQL 标准的重大更新版本，以弥补旧版的不足，称为 ANSI SQL 89，该版本也被 ISO 采纳。

目前，市场上主要的关系型数据库都有自己的 SQL 变种，但是它们都遵守 ANSI SQL 89 标准。

SQL 执行过程

当你在任何一款 RDBMS 中执行 SQL 命令时，系统首先确定执行请求的最佳方式，然后 SQL 引擎将会翻译 SQL 语句，并处理请求任务。

整个执行过程包含了多种组件，比如：

查询调度程序；

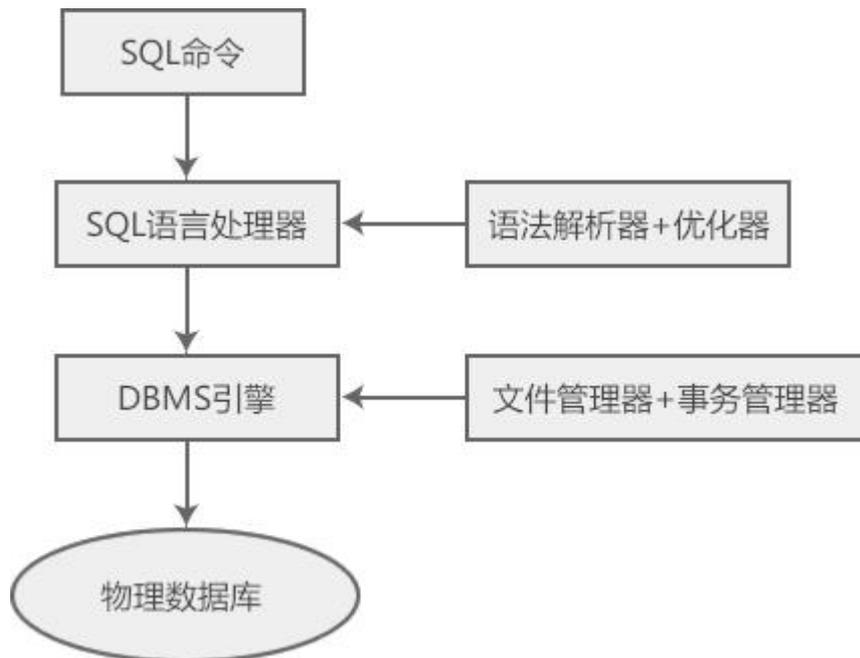
优化引擎；

传统的查询引擎；

SQL 查询引擎。

传统查询引擎能够处理所有的非 SQL 命令，但是 SQL 引擎并不能处理逻辑文件。

下图展示了 SQL 的体系结构：



SQL 命令

与关系型数据库有关的 SQL 命令包括 CREATE、SELECT、INSERT、UPDATE、DELETE、DROP 等，根据其特性，可以将它们分为以下几个类别。

1) DDL - Data Definition Language，数据定义语言

对数据的结构和形式进行定义，一般用于数据库和表的创建、删除、修改等。

命令 说明

CREATE 用于在数据库中创建一个新表、一个视图或者其它对象。

ALTER 用于修改现有的数据库，比如表、记录。

DROP 用于删除整个表、视图或者数据库中的其它对象

2) DML - Data Manipulation Language，数据处理语言

对数据库中的数据进行处理，一般用于数据项（记录）的插入、删除、修改和查询。

命令 说明

SELECT 用于从一个或者多个表中检索某些记录。

INSERT 插入一条记录。

UPDATE 修改记录。

DELETE 删除记录。

3) DCL - Data Control Language，数据控制语言

控制数据的访问权限，只有被授权的用户才能进行操作。

命令 说明

GRANT 向用户分配权限。

REVOKE 收回用户权限。

2.常用数据库访问接口简介

不同的程序设计语言会有各自不同的数据库访问接口，程序语言通过这些接口，执行 SQL 语句，进行数据库管理。主要的数据库访问接口主要有 ODBC、JDBC、ADO.NET 和 PDO。

ODBC

ODBC（Open Database Connectivity，开放数据库互连）为访问不同的 SQL 数据库提供了一个共同的接口。ODBC 使用 SQL 作为访问数据的标准。这一接口提供了最大限度的互操作性。一个应用程序可以通过共同的一组代码访问不同的 SQL 数据库管理系统。

一个基于 ODBC 的应用程序对数据库的操作不依赖任何 DBMS，不直接与 DBMS 打交道，所有的数据库操作由对应的 DBMS 的 ODBC 驱动程序完成。也就是说，不论是 MySQL 还是 Oracle 数据库，均可用 ODBC API 进行访问。由此可见，ODBC 的最大优点是能以统一的方式处理所有的数据库。

JDBC

Java Data Base（JDBC，Java 数据库连接）用于 Java 应用程序连接数据库的标准方法，是一种用于执行 SQL 语句的 Java API，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。

ADO.NET

ADO.NET 是微软在 .NET 框架下开发设计的一组用于和数据源进行交互的面向对象类库。ADO.NET 提供了对关系数据、XML 和应用程序的访问，允许和不同类型的数据源以及数据库进行交互。

PDO

PDO（PHP Data Object）为 PHP 访问数据库定义了一个轻量级的、一致性的接口，它提供了一个数据访问抽象层，这样，无论使用什么数据库，都可以通过一致的函数执行查询和获取数据。PDO 是 PHP 5 新加入的一个重大功能。

3.常用数据库大汇总（附带优缺点）

现在已经存在了很多优秀的商业数据库，如甲骨文（Oracle）公司的 Oracle 数据库、IBM 公司的 DB2 数据库、微软公司的 SQL Server 数据库和 Access 数据库。同时，还有很多优秀的开源数据库，如 MySQL 数据库，PostgreSQL 数据库等。下面介绍这些常见的数据库。

1) Oracle

Oracle 是甲骨文公司的一款关系型数据库管理系统，在数据库领域一直处于领先地位的产品，是目前世界上流行的关系型数据库之一，是一种高效率、可靠性好、适应高吞吐量的数据库方案。



Oracle 图标

优点:

Oracle 可移植性好,能在所有主流平台上运行(包括 Windows),完全支持所有工业标准。采用完全开放策略,使客户可以选择最适合解决方案。以及对开发商的全力支持。

获得最高认证级别的 ISO 标准认证,安全性高。

与其它数据库相比,Oracle 性能最高。保持着开放平台下 TPC-D 和 TPC-C 世界记录。

多层次网络计算,支持多种工业标准,可以用 ODBC、JDBC、OCI 等网络客户连接。

完全向下兼容,因此被广泛应用,且风险低。

向下兼容指的是高版本支持低版本的或者说后期开发的版本支持和兼容早期开发的版本。

缺点:

对硬件的要求高

价格比较昂贵

管理维护麻烦

操作比较复杂

2) SQL Server

SQL Server 是 Microsoft (微软)公司推出的关系型数据库管理系统,主要应用于大型的管理系统中。



SQL Server 图标

优点:

与微软的 Windows 系列操作系统的兼容性很好。

高性能设计,可充分利用 WindowsNT 的优势。

系统管理先进,支持 Windows 图形化管理工具,支持本地和远程的系统管理和配置。

强壮的事务处理功能，采用各种方法保证数据的完整性。
支持对称多处理器结构、存储过程、ODBC，并具有自主的 SQL 语言。

缺点：

SQL Server 只能在 Windows 系统上运行，没有丝毫开放性。

没有获得任何安全证书。

多用户时性能不佳。

只支持 C/S 模式，SQL Server C/S 结构只支持 Windows 客户用 ADO、DAO、OLEDB、ODBC 连接。

3) MySQL

MySQL 是一种开放源代码的关系型数据库管理系统，由瑞典 MySQL AB 公司开发，属于 Oracle 旗下产品。因为其速度、可靠性和适应性而备受关注。MySQL 是流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的应用软件之一。



MySQL 图标

优点：

性能卓越服务稳定，很少出现异常宕机

开放源代码且无版权制约，自主性强、使用成本低。

历史悠久、社区及用户非常活跃，遇到问题，可以很快获取到帮助。

软件体积小，安装使用简单，并且易于维护，安装及维护成本低。

支持多种操作系统，提供多种 API 接口，支持多种开发语言。

缺点：

MySQL 最大的缺点是其安全系统，主要是复杂而非标准，只有调用 mysqladmin 来重读用户权限才会发生改变。

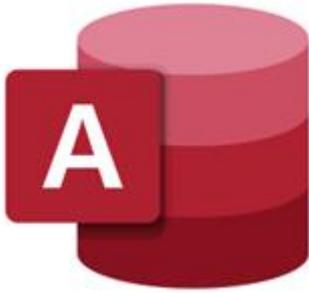
MySQL 不允许调试存储过程，开发和维护存储过程很难。

MySQL 不支持热备份。

MySQL 的价格随平台和安装方式变化。

4) Access

Access 是由 Microsoft（微软）发布的小型关系数据库管理系统，是微软把数据库引擎的图形用户界面和软件开发工具结合在一起的一个数据库管理系统。



Access 图标

优点:

存储方式简单,易于维护管理。Access 的对象有表、查询、窗体、报表、页、宏和模块,以上对象都存放在后缀为 (.mdb 或 .accdb) 的数据库文件中,便于用户的操作和管理。

Access 是一个面向对象的开发工具,这种基于面向对象的开发方式,使得开发应用程序更为简便。

界面友好、易操作。Access 是一个可视化工具,风格与 Windows 完全一样,用户想要生成对象应用,只要使用鼠标进行拖放即可,非常直观方便。系统还提供了表生成器、查询生成器、报表设计器以及数据库向导、表向导、查询向导、窗体向导、报表向导等工具,使得操作简便,容易使用和掌握。

集成环境,可以处理多种数据信息。Access 基于 Windows 操作系统下的集成开发环境,该环境集成了各种向导和生成器工具,极大地提高了开发人员的工作效率,使得建立数据库、创建表、设计用户界面、设计数据查询、报表打印等可以方便有序地进行。

支持广泛,易于扩展,弹性大。Access 是一个既可以只用来存放数据的数据库,也可以作为一个客户端开发工具来进行数据库应用系统开发。即可以开发方便易用的小型软件,也可以用来开发大型的应用系统。

缺点:

不支持并发处理。

数据库存储量小安全性不够高。

Access 是小型数据库,当数据量过大时,一般百 M 以上(纯数据,不包括窗体、报表等客户端对象)性能会变差。

虽然理论上支持 255 个并发用户,但实际上根本支持不了那么多,如果以只读方式访问大概在 100 个用户左右,而如果是并发编辑,则大概在 10-20 个用户。

单表记录数过百万时,性能就会变得较差,如果加上设计不良,这个限度还要降低。

不能编译成可执行文件 (.exe),必须要安装 Access 运行环境才能使用。

5) DB2

DB2 是美国 IBM 公司开发的一款支持多媒体、Web 的关系型数据库管理系统。主要应用于大型应用系统,具有较好的可伸缩性,可支持从大型机到单用户环境。



DB2 图标

优点:

相比较 MySQL 和 Oracle 两种数据库来说, DB2 提供了高层次的数据利用性、完整性、安全性、可恢复性, 以及小规模到大规模地应用程序执行能力, 具有与平台无关的基本功能和 SQL 命令。

DB2 采用了数据分级技术, 能够使大型数据很方便的下载到数据库服务器, 使数据库本地化和远程连接透明化。

拥有非常完备的查询优化器, 改善了查询性能, 并支持多任务并行查询。

具有很好的网络支持能力, 每个子系统可以连接十几万个分布式用户, 可同时激活上千个活动线程, 对大型分布式应用系统更加使用。

DB2 可跨平台使用。

缺点:

配置文件和参数多, 且命名不规范。

一些 DB2 产品开发不方便。

和 Oracle 相比, 命令多, 且没 Oracle 统一规范的好。

由于其设计框架的问题, 如果用户对数据库的本身优化和应用程序优化做的不足, 那么 DB2 容易出现锁等待现象。

6) PostgreSQL

PostgreSQL 是一款富有特色的自由数据库管理系统, 甚至可以说是最强大的自由软件数据库管理系统。该数据库管理系统支持了目前世界上最丰富的数据类型。是自由软件数据库管理系统中唯一支持事务、子查询、多版本并行控制系统、数据完整性检查等特性的自由软件。



PostgreSQL 图标

优点:

PostgreSQL 遵循的是 BSD 协议, 是一个完全开源、免费、同时非常强大的关系型数据库。

与 PostgreSQL 配合的有很多分布式集群软件，如 pgpool、pgcluster、slony、plpoxxy 等等，很容易做读写分离、负载均衡、数据水平拆分等方案，而这些 MySQL 则比较难实现。PostgreSQL 源代码写的很清晰，易读性比 MySQL 强，所以很多公司基本都是以 PostgreSQL 做二次开发的。

PostgreSQL 是多进程的，而 MySQL 是多线程的。并发不高时，MySQL 处理速度快，但当并发高的时候，对于现在多核的单台机器上，MySQL 的总体处理性能不如 PostgreSQL，原因是 MySQL 的线程无法充分利用 CPU 的能力。

PostgreSQL 有很强大的查询优化器，支持很复杂的查询处理。

BSD 开源协议是一个给予使用者很大自由的协议。可以自由的使用，修改源代码，也可以将修改后的代码作为开源或者专有软件再发布。

缺点：

对于简单而繁重的读取操作，相比较其它数据库来说，PostgreSQL 性能较低。

PostgreSQL 数据库扩容花费时间很长。

4.MySQL 是什么？它有什么优势？

随着时间的推移，开源数据库在中低端应用中逐渐流行起来，占据了很大的市场份额。开源数据库具有免费使用、配置简单、稳定性好、性能优良等特点，而 MySQL 数据库正是开源数据库中的杰出代表。

开源全称为“开放源代码”。很多人认为开源软件最明显的特点是免费，但实际上并不是这样的，开源软件最大的特点应该是开放，也就是任何人都可以得到软件的源代码。可以在源代码的基础上加以修改学习，甚至重新发放，当然是在版权限制范围之内。不清楚的读者可以参考《开源就等于免费吗？用事实来说话》一节。

MySQL 是最流行的数据库之一，是一个免费开源的关系型数据库管理系统，但并不意味着该数据库是完全免费的。MySQL 由瑞典 MySQL AB 公司开发，目前属于 Oracle 公司。MySQL 适合中小型软件，被个人用户以及中小企业青睐。

针对不同的用户，MySQL 分为两个版本：

MySQL Community Server（社区版）：该版本是自由下载且完全免费的，但是官方不提供技术支持。

MySQL Enterprise Server（企业版）：该版本是收费的，而且不能下载，但是该版本拥有完善的技术支持（官方提供电话技术支持）。

注意：MySQL Cluster 主要用于架设群服务器，需要在社区服务或企业版的基础上使用。

MySQL 的命名机制由 3 个数字和 1 个后缀组成，例如 mysql-5.7.20：

第 1 个数字“5”是主版本号，用于描述文件的格式，所有版本 5 的发行版都有相同的文件夹格式。

第 2 个数字“7”是发行级别，主版本号和发行级别组合在一起便构成了发行序列号。

第 3 个数字“20”是在此发行系列的版本号，随每次新发行的版本递增。通常选择已经发行的最新版本。

在 MySQL 开发过程中，同时存在多个发布系列，每个发布系列的成熟度处在不同阶段。

MySQL 5.7 是最新开发的稳定（GA）发布系列，是将执行新功能的系列，目前已经可以正常使用。

MySQL 5.6 是比较稳定的（GA）发布系列，只针对漏洞修复重新发布，不增加会影响稳定性的新功能。

MySQL 5.1 是一个稳定的（产品质量）发布系列，只针对严重漏洞修复和安全修复重新发布，不增加影响该系列稳定性的重要功能。

注意：对于 MySQL 4.1 等低于 5.0 的老版本，官方将不再提供支持。所有发布的 MySQL 版本已经经过严格标准的测试，可以保证安全可靠地使用。针对不同的操作系统，读者可以在 MySQL 官方下载页面（<http://dev.mysql.com/downloads/>）下载相应的安装文件。

MySQL 的特点、优势

MySQL 数据库管理系统具有很多的优势，下面总结了其中几种。

1) MySQL 是开放源代码的数据库

MySQL 是开放源代码的数据库，任何人都可以获取该数据库的源代码。这就使得任何人都可以修正 MySQL 的缺陷，并且任何人都能以任何目的来使用该数据库。MySQL 是一款可以自由使用的数据库。

2) MySQL 的跨平台性

MySQL 不仅可以在 Windows 系列的操作系统上运行，还可以在 UNIX、Linux 和 Mac OS 等操作系统上运行。因为很多网站都选择 UNIX、Linux 作为网站的服务器，所以 MySQL 的跨平台性保证了其在 Web 应用方面的优势。虽然微软公司的 SQL Server 数据库是一款很优秀的商业数据库，但是其只能在 Windows 系列的操作系统上运行。因此，MySQL 数据库的跨平台性是一个很大的优势。

3) 价格优势

MySQL 数据库是一个自由软件，任何人都可以从 MySQL 的官方网站上下载该软件，这些社区版本的 MySQL 都是免费试用的，即使是需要付费的附加功能，其价格也是很便宜的。相对于 Oracle、DB2 和 SQL Server 这些价格昂贵的商业软件，MySQL 具有绝对的价格优势。

4) 功能强大且使用方便

MySQL 是一个真正的多用户、多线程 SQL 数据库服务器。它能够快速、有效和安全的处理大量的数据。相对于 Oracle 等数据库来说，MySQL 的使用是非常简单的。MySQL 主要目标是快速、健壮和易用。

MySQL 与常用的主流数据库 Oracle、SQL Server 相比，主要特点就是免费，并且在任何平台上都能使用，占用的空间相对较小。但是，MySQL 也有一些不足，比如对于大型项目来说，MySQL 的容量和安全性就略逊于 Oracle 数据库。

5. MySQL 适用于哪些场景？

MySQL 是目前世界上最流行的开源关系数据库，大多应用于互联网行业。比如，在国内，大家所熟知的百度、腾讯、淘宝、京东、网易、新浪等，国外的 Google、Facebook、Twitter、GitHub 等都在使用 MySQL。社交、电商、游戏的核心存储往往也是 MySQL。

任何产品都不可能是万能的，也不可能适用于所有的应用场景。那么 MySQL 到底适用于哪些场景又不适用于哪些场景呢？

1. Web 网站系统

Web 网站开发者是 MySQL 最大的客户群，也是 MySQL 发展史上最为重要的支撑力量。

MySQL 之所以能成为 Web 网站开发者们最青睐的数据库管理系统，是因为 MySQL 数据库的安装配置都非常简单，使用过程中的维护也不像很多大型商业数据库管理系统那么复杂，而且性能出色。还有一个非常重要的原因就是 MySQL 是开放源代码的，完全可以免费使用。

2、日志记录系统

MySQL 数据库的插入和查询性能都非常的高效，如果设计的好，在使用 MyISAM 存储引擎的时候，两者可以做到互不锁定，达到很高的并发性能。所以，对需要大量的插入和查询日志记录的系统来说，MySQL 是非常不错的选择。比如处理用户的登录日志，操作日志等，都是非常适合的应用场景。

3、数据仓库系统

随着现在数据仓库数据量的飞速增长，我们需要的存储空间越来越大。数据量的不断增长，使数据的统计分析变得越来越低效，也越来越困难。下面是几个主要的解决思路。

- 1) 采用昂贵的高性能主机以提高计算性能，用高端存储设备提高 I/O 性能，效果理想，但是成本非常高；
- 2) 通过将数据复制到多台使用大容量硬盘的廉价 PC Server 上，以提高整体计算性能和 I/O 能力，效果尚可，存储空间有一定限制，成本低廉；
- 3) 通过将数据水平拆分，使用多台廉价的 PC Server 和本地磁盘来存放数据，每台机器上面都只有所有数据的一部分，解决了数据量的问题，所有 PC Server 一起并行计算，也解决了计算能力问题，通过中间代理程序调配各台机器的运算任务，既可以解决计算性能问题又可以解决 I/O 性能问题，成本也很低廉。

在上面的三个方案中，第二和第三个的实现，MySQL 都有较大的优势。通过 MySQL 的简单复制功能，可以很好的将数据从一台主机复制到另外一台，不仅仅在局域网内可以复制，在广域网同样可以。

当然，很多人可能会说，其他的数据库同样也可以做到，不是只有 MySQL 有这样的功能。确实，很多数据库同样能做到，但是 MySQL 是免费的，其他数据库大多都是按照主机数量或者 cpu 数量来收费，当我们使用大量的 PC Server 的时候，License 费用相当惊人。所以第一个方案，基本上所有数据库系统都能够实现，但是其高昂的成本不是每一个公司都能够承担的。

4、嵌入式系统

嵌入式环境对软件系统最大的限制是硬件资源非常有限，在嵌入式环境下运行的软件系统，

必须是轻量级低消耗的软件。

MySQL 在资源的使用方面的伸缩性非常大，可以在资源非常充裕的环境下运行，也可以在资源非常少的环境下正常运行。它对于嵌入式环境来说，是一种非常合适的数据库系统，而且 MySQL 有专门针对于嵌入式环境的版本。

并且，MySQL 的定位是通用数据库，各种类型的应用一般都能利用到 MySQL 存取数据的优势。业内生产实践证明，MySQL 更适合中小型企业。以目前的软硬件产品水平来看，如果数据超过几个 TB 将难以高效利用 MySQL。

MySQL 可以作为传统的关系型数据库产品使用，也可以当作一个 key-value 产品来使用。由于它具有优秀的灾难恢复功能，因此相对于目前市场上的一些 key-value 产品会更有优势。

6.学 MySQL 前，需要了解这些数据库专业术语

在正式学习 MySQL 数据库前，我们有必要先了解一下数据库中的专业术语。下面汇总了一些在学习 MySQL 过程中会遇到的专业术语。

数据库管理系统（DBMS）是位于操作系统与用户之间的一种操纵和管理数据库的软件。关系型数据库通过关系数据库管理系统（RDBMS）进行管理。

关系（Relational，即 RDBMS 里的 R）表示这是一种特殊的 DBMS，数据库中表与表之间要存在关系。

数据库（DataBase，即 RDBMS 里的 DB）是一个用来存储和管理数据的仓库。它的存储空间很大，并且有一定的数据存放规则。通过由行和列组成的二维表（类似 Excel 工作表）来管理数据。数据库中可以同时存储多个表。

管理系统（Management System，即 RDBMS 里的 MS）是一个软件，我们可以通过它来插入（insert）、查询（query）、修改（modify）或删除（delete）表中的数据。

用来管理数据的二维表在关系数据库中简称为表（Table），每个表由多个行（Row）和列（Column）组成。

表的列（垂直方向）称为字段，是具有相同数据类型的数据集合。表的行（水平方向）称为记录（Record），相当于一条数据。记录可以包含多项信息，表中的每一列都对应于其中的某一项。

下图是一个统计学生信息的 student 表。可以看到，每个列都包含了所有学生的某个信息，比如 name（姓名）。而每行则包含了某个学生的所有信息，即 id（编号）、name（姓名）、age（年龄）、stuno（学号）等。

id	name	age	stuno
1	zhangsan	18	23
2	lisi	19	24
3	wangwu	18	25
4	zhaoliu	18	26
5	zhangsan	18	27
6	wangwu	20	28

列名

行 (记录)

列 (字段)

SQL (Structured Query Language, 结构化查询语言) 是用来操作关系型数据库的语言, 使用 SQL 可以对数据库和表进行添加、删除、修改和查询等操作。

与其它计算机语言一样, 初次接触 SQL 的人肯定会觉得它很奇怪。例如, 在创建表时, 很多人都会把表与图表或图片联系起来, 但是 MySQL 不是这样的。在 MySQL 中创建表时, 你必须输入类似下面这样的内容。

```
CREATE TABLE `student` (
  `id` int(4),
  `name` varchar(20),
  `age` int(4),
  `stuno` int(11),
  PRIMARY KEY (`id`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

如果你对 SQL 还不太熟悉, 可能会对以上语句心生畏惧。不过请放心, 随着教程的深入学习, 你会很快的熟练使用 SQL, 并且对上述 SQL 语句的看法也会悄然地发生变化。它不再是一组怪诞地表达, 而是一种有助于信息描述的强大工具。

7.MySQL 客户端和服务端架构 (C/S 架构)

客户端-服务器 (Client/Server) 结构简称 C/S 结构, 是一种网络架构, 通常在该网络架构下的软件分为客户端和服务端。

服务器是整个应用系统资源的存储和管理中心, 多个客户端分别各自处理相应的功能, 共同实现完整的应用。在客户/服务器结构中, 客户端用户的请求被传送到数据库服务器, 数据库服务器进行处理后, 将结果返回给用户, 从而减少网络数据的传输量。

用户在使用应用程序时, 首先启动客户端, 然后通过相关命令告知服务器进行连接以完成各种操作, 而服务器则按照此请示提供相应的服务。每一个客户端软件的实例都可以向一个服

务器或应用程序服务器发出请求。

客户端和服务程序通常不在同一台计算机上运行，比如，我们平时在当当网上买书的时候，所使用的电脑和网页浏览器就被当做了一个客户端，同时，组成当当网的电脑、数据库和应用程序就被当做服务器。

数据库管理系统可分为两类：一类是基于共享文件系统的数据库管理系统，例如 Microsoft Access 和 FileMaker，主要用于桌面用途，不适合用于高端或更关键的应用；另一类是基于客户端-服务器的数据库管理系统，例如 MySQL、Oracle 和 SQL Server 等数据库。

服务器软件负责访问和处理所有数据的一个软件，这个软件运行在称为数据库服务器的计算机上，并且与数据文件打交道的只有服务器软件。

关于数据的添加、删除和更新等所有的请求都由服务器完成。这些请求来自于运行客户端的计算机。客户端用来和用户打交道。例如，如果你请求一个按字母顺序列出的产品表，则客户端会通过网络提交该请求给服务器，服务器处理这个请求，然后根据需要对数据进行过滤、丢弃和排序，最后把结果返回到客户端。

注意：客户端和服务程序可以安装在两台计算机或一台计算机上，不管它们在不在相同的计算机上，客户端都要与服务器进行通信。

以上这些过程对用户都是透明的，你不需要直接访问数据文件。为了使用 MySQL，你需要访问运行 MySQL 服务器的计算机和发布命令到 MySQL 客户端的计算机。

对于 MySQL 数据库管理系统，服务器为 MySQL DBMS。你可以在本地安装的副本上运行，也可以连接到运行在你具有访问权的远程服务器上的一个副本。

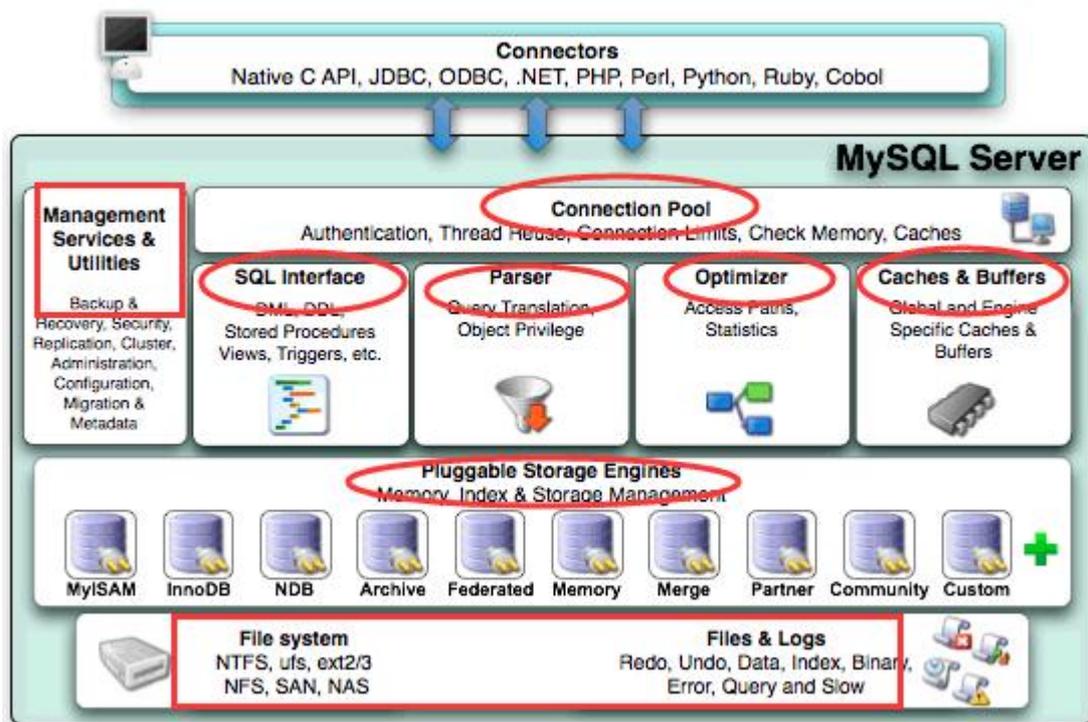
客户端可以是 MySQL 提供的工具（如 MySQL Workbench、SQLyog）、脚本语言（如 Perl）、Web 应用开发语言（如 ASP、ColdFusion、JSP 和 PHP）和程序设计语言（如 C、C++、Java）等。

8.明白了 MySQL 内部结构才能成为高手！

麻雀虽小，五脏俱全。MySQL 虽然以简单著称，但其内部结构并不简单，本节主要介绍 MySQL 的整体架构组成。

学习 MySQL 就好比盖房子，如果想把房子盖的特别高，地基一定要稳，基础一定要牢固。学习 MySQL 数据库前要先了解它的内部结构，这是学好 MySQL 数据库的前提。

MySQL 由连接池、SQL 接口、解析器、优化器、缓存、存储引擎等组成，可以分为三层，即 MySQL Server 层、存储引擎层和文件系统层。MySQL Server 层又包括连接层和 SQL 层。如下是官方文档中 MySQL 的基础架构图：



MySQL 基础架构图

上图中，Connection pool 为连接层，Management Services & Utilities … Caches & Buffers 为 SQL 层，Pluggable Storage Engines 为存储引擎层，File system、Files & Logs 为文件系统层。

Connectors 不属于以上任何一层，可以将 Connectors 理解为各种客户端、应用服务，主要指的是不同语言与 SQL 的交互。

1. 连接层

应用程序通过接口（如 ODBC、JDBC）来连接 MySQL，最先连接处理的是连接层。连接层包括通信协议、线程处理、用户名密码认证 3 部分。

通信协议负责检测客户端版本是否兼容 MySQL 服务端。

线程处理是指每一个连接请求都会分配一个对应的线程，相当于一 SQL 对应一个线程，一个线程对应一个逻辑 CPU，在多个逻辑 CPU 之间进行切换。

密码认证用来验证用户创建的账号、密码，以及 host 主机授权是否可以连接到 MySQL 服务器。

Connection Pool（连接池）属于连接层。由于每次建立连接都需要消耗很多时间，连接池的作用就是将用户连接、用户名、密码、权限校验、线程处理等需要缓存的需求缓存下来，下次可以直接用已经建立好的连接，提升服务器性能。

2. SQL 层

SQL 层是 MySQL 的核心，MySQL 的核心服务都是在这层实现的。主要包含权限判断、查询缓存、解析器、预处理、查询优化器、缓存和执行计划。

权限判断可以审核用户有没有访问某个库、某个表，或者表里某行数据的权限。

查询缓存通过 Query Cache 进行操作，如果数据在 Query Cache 中，则直接返回结果给客户端，不必再进行查询解析、优化和执行等过程。

查询解析器针对 SQL 语句进行解析，判断语法是否正确。

预处理器对解析器无法解析的语义进行处理。

查询优化器对 SQL 进行改写和相应的优化，并生成最优的执行计划，就可以调用程序的 API 接口，通过存储引擎层访问数据。

Management Services & Utilities、SQL Interface、Parser、Optimizer 和 Caches & Buffers 属于 SQL 层，详细说明如下表所示。

名称	说明
Management Services & Utilities	MySQL 的系统管理和控制工具，包括备份恢复、MySQL 复制、集群等。
SQL Interface (SQL 接口)	用来接收用户的 SQL 命令，返回用户需要查询的结果。例如 SELECT FROM 就是调用 SQL Interface。
Parser (查询解析器)	在 SQL 命令传递到解析器的时候会被解析器验证和解析，以便 MySQL 优化器可以识别的数据结构或返回 SQL 语句的错误。
Optimizer (查询优化器)	SQL 语句在查询之前会使用查询优化器对查询进行优化，同时验证用户是否有权限进行查询，缓存中是否有可用的最新数据。它使用“选取-投影-连接”策略进行查询。例如 SELECT id, name FROM student WHERE gender = "女";语句中，SELECT 查询先根据 WHERE 语句进行选取，而不是将表全部查询出来以后再进行 gender 过滤。SELECT 查询先根据 id 和 name 进行属性投影，而不是将属性全部取出以后再进行过滤，将这两个查询条件连接起来生成最终查询结果。
Caches & Buffers (查询缓存)	如果查询缓存有命中的查询结果，查询语句就可以直接去查询缓存中取数据。这个缓存机制是由一系列小缓存组成的，比如表缓存、记录缓存、key 缓存、权限缓存等。

3. 存储引擎层

Pluggable Storage Engines 属于存储引擎层。存储引擎层是 MySQL 数据库区别于其他数据库最核心的一点，也是 MySQL 最具特色的一个地方。主要负责 MySQL 中数据的存储和提取。

因为在关系数据库中，数据的存储是以表的形式存储的，所以存储引擎也可以称为表类型（即存储和操作此表的类型）。

4. 文件系统层

文件系统层主要是将数据库的数据存储在操作系统的文件系统之上，并完成与存储引擎的交互。

说明：学习阅读完本节内容，了解 MySQL 体系结构即可，教程后面会详细介绍每个模块。

9. 如何学习数据库（新手必看）？

很多初学者在数据库学习的时候，由于学习难度较大，往往不知所措，容易打击学习的自信心，没有了学习的兴趣，这些都是难以避免的。下面主要介绍作为初学者应该怎么学习 MySQL。

初学者学习 MySQL 必须掌握的知识点如下：

MySQL 的下载安装。熟悉 MySQL 的配置文件，目录结构。

MySQL 服务器的启动，登录与退出。

MySQL 常用命令及语法规则。

MySQL 数据类型与数据表的操作。例如，数据表的增删改查、单表查询、多表查询等。

MySQL 运算符和函数，例如，日期函数，时间函数，信息函数，聚合函数，加密函数，自定义函数等。

MySQL 存储过程，存储过程的调用。

MySQL 各个存储引擎的特点，如何选择合适的存储引擎等。

MySQL 事务的概念和使用等。

MySQL 权限管理和用户管理等。

10. 如何高效的学习 MySQL

1) 培养兴趣

兴趣是最好的老师，不论学习什么知识，兴趣都可以极大地提高学习效率，当然学习数据库也不例外。

2) 及时学习新知识

正确、有效地利用学习资源，可以参考别人解决问题的思路和经验，即使获取掌握最新的知识。

3) 多练习、多操作

数据库系统具有极强的操作性，所以要想熟练的掌握数据库，就必须经常上机练习。只有实际操作使用才能发现问题。通常情况下，数据库管理员工作的时间越长，其工作经验就越丰富。很多复杂的问题，都可以根据数据库管理员的经验来很好地解决。上机练习的过程中，可以将学到的数据库理论知识理解得更加透彻。

4) 多编写 SQL 语句

SQL 语句是数据库的灵魂。数据库中的很多操作都是通过 SQL 语句来实现的。虽然现在的数据库都有易用的图形界面，可以直接在图形界面上创建数据库和表。但是，图形界面却掩盖了这些操作是如何实现的。只有经常使用 SQL 语句来操作数据库中的数据，才能更加深刻地理解数据库。

5) 通过 Java 等编程语言来操作数据库

开发的软件系统中都需要使用数据库。软件开发者学习数据库的最终目的就是在软件开发中使用数据库。因此，在学习过程中，多思考一下如何使用 Java 等程序语言来操作数据库。最好多编一些程序来操作数据库。这样，既可以加深对数据库的理解，也可以提高自己的编程能力。

6) 数据库理论知识不能丢

计算机领域的技术非常强调基础，刚开始学习可能还认识不到这一点，但是随着技术应用的深入，只有有着扎实的基础功底，才能在技术的道路上走得更快、更远。

数据库理论知识是学好数据库的基础，虽然理论知识会有点枯燥，但是这是学好数据库的前提。如果没有理论基础，学习的东西就不扎实。例如，数据库理论中会讲解 E-R 图、数据

库设计原则等知识，如果不了解这些知识，就很难独立设计一个很好的数据库及表。可以将理论和实例结合在一起学习，这样效率会更高。

11.启动 MySQL 服务的两种方式（图解）

MySQL 服务和 MySQL 数据库不同，MySQL 服务是一系列的后台进程，而 MySQL 数据库则是一系列的数据目录和数据文件。MySQL 数据库必须在 MySQL 服务启动之后才可以进行访问。本节主要介绍如何启动 MySQL 服务。

Windows 系统下启动停止 MySQL 服务的方式主要有以下两种：

通过计算机管理方式

通过命令行方式

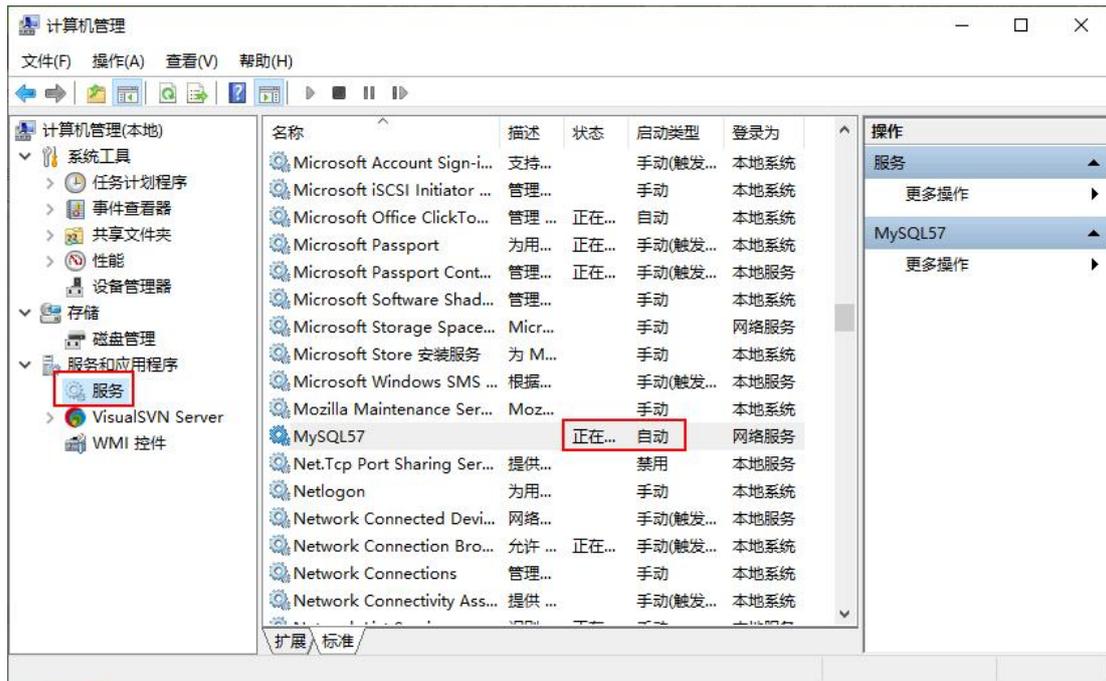
通过计算机管理方式

通过 Windows 的服务管理器查看修改，步骤如下：

步骤 1)：在桌面上右击“此电脑”→“管理”命令，如图所示。

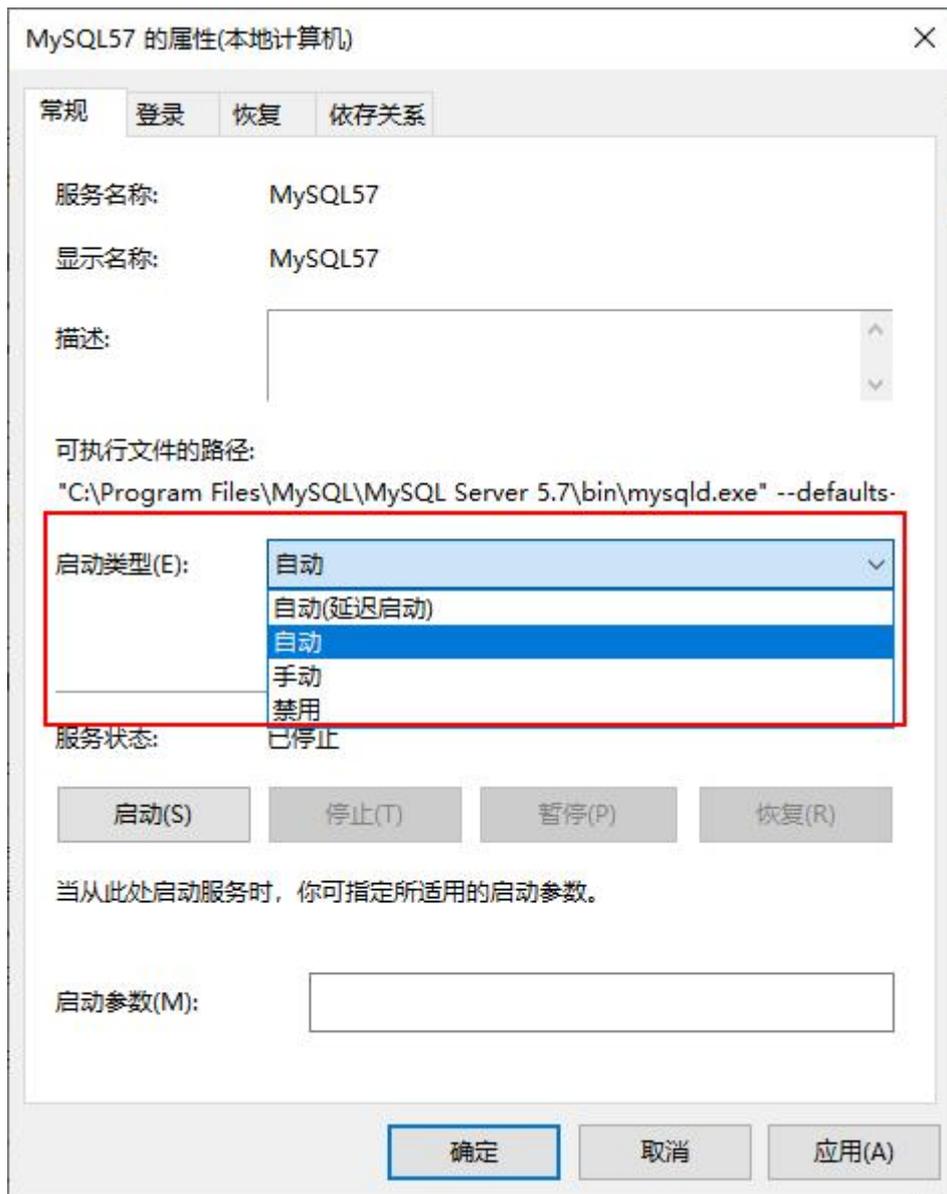


步骤 2)：弹出“计算机管理”对话框，双击“服务和应用程序”，用户可查看计算机的服务状态，MySQL 的状态为“正在运行”，表明该服务已经启动，如图所示。



在图中可以看到，服务已经启动，而且启动类型为自动。如果没有“正在运行”字样，说明MySQL 服务未启动。

可以在此处鼠标右击选择属性进入“MySQL 的属性”的界面，如图所示。



可以在 MySQL 的属性界面中设置服务状态。可以将服务状态设置为“启动”、“停止”、“暂停”和“恢复”命令。

还可以设置启动类型，在启动类型处的下拉菜单中可以选择“自动”、“手动”和“禁用”。这 3 种启动类型的说明如下：

自动：MySQL 服务是自动启动，可以手动将状态变为停止、暂停和重新启动等。

手动：MySQL 服务需要手动启动，启动后可以改变服务状态，如停止、暂停等。

已禁用：MySQL 服务不能启动，也不能改变服务状态。

如果需要经常练习 MySQL 数据库的操作，可以将 MySQL 设置为自动启动，这样可以避免每次手动启动 MySQL 服务。当然，如果使用 MySQL 数据库的频率很低，可以考虑将 MySQL 服务设置为手动启动，这样可以避免 MySQL 服务长时间占用系统资源。

通过命令行方式

可以通过 DOS 命令启动 MySQL 服务，点击“开始”菜单，在搜索框中输入“cmd”，以管理员身份运行，按回车键，弹出命令提示符界面。然后输入 `net start mysql`，按回车键，就能启动 MySQL 服务，停止 MySQL 服务的命令为 `net stop mysql`，如图所示。



```
管理员: 命令提示符
C:\WINDOWS\system32>net start mysql57
MySQL57 服务正在启动...
MySQL57 服务已经启动成功。

C:\WINDOWS\system32>net stop mysql57
MySQL57 服务正在停止...
MySQL57 服务已成功停止。

C:\WINDOWS\system32>net stop mysql
服务名无效。

请键入 NET HELPMSG 2185 以获得更多的帮助。

C:\WINDOWS\system32>
```

注意：`net start mysql57` 和 `net stop mysql57` 命令中的 `mysql57` 是 MySQL 服务器名称，如果你的 MySQL 服务名称是 `DB` 或其它的名字，应该输入 `net start DB` 或其它名称，否则提示服务名无效。

12. 结合实例，彻底搞懂数据库设计的三大范式

为了建立冗余较小、结构合理的数据库，设计数据库时必须遵循一定的规则。在关系型数据库中，这种规则就是范式。范式是符合某一种级别的关系模式的集合。关系型数据库中的关系必须满足一定的要求，即满足不同的范式。

目前关系型数据库有六种范式，分别为：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、第四范式（4NF）、第五范式（5NF）和第六范式（6NF）。要求最低的范式是第一范式。第二范式在第一范式的基础上又进一步的添加了要求，其余范式依次类推。

一般说来，数据库只需满足第三范式就行了，而通常我们用的最多的就是第一范式、第二范式、第三范式，也就是接下来要讲的“三大范式”。

1) 第一范式

第一范式（1NF）用来确保每列的原子性，要求每列（或者每个属性值）都是不可再分的最小数据单元（也称为最小的原子单元）。

例如，客人住宿信息表（姓名，客人编号，地址，客房号，客房描述，客房类型，客房状态，床位数，入住人数，价格）。

其中，“地址”列还可以细分为国家、省、市、区等，甚至有的程序还把“姓名”列也拆分为“姓”和“名”等。如果业务需求中不需要拆分“地址”和“姓名”列，则该数据表符合第一范式，如果需要将“地址”列拆分，则下列写法符合第一范式：

客人住宿信息表(姓名, 客人编号, 国家, 省, 市, 区, 门牌号, 客房号, 客房描述, 客房类型, 客房状态, 床位数, 入住人数, 价格)。

2) 第二范式

第二范式(2NF)在第一范式的基础上更进一层, 要求表中的每列都和主键相关, 即要求实体的唯一性。如果一个表满足第一范式, 并且除了主键以外的其他列全部都依赖于该主键, 那么该表满足第二范式。

客人住宿信息表中的数据主要用来描述客人住宿信息, 所以该表主键为(客人编号, 客房号):

“姓名”列、“地址”列 \rightarrow “客人编号”列。

“客房描述”列、“客房类型”列、“客房状态”列、“床位数”列、“入住人数”列、“价格”列 \rightarrow “客房号”列。

其中，“ \rightarrow ”符号代表依赖。以上各列没有全部依赖于主键(客人编号, 客房号), 只是部分依赖于主键, 不符合第二范式。

使用第二范式后, 客人住宿信息表可以分解成以下两个表:

客人信息表(客人编号, 姓名, 地址, 客房号, 入住时间, 结账日期, 押金, 总金额), 主键为“客人编号”列, 其他列都全部依赖于主键列。

客房信息表(客房号, 客房描述, 客房类型, 客房状态, 床位数, 入住人数, 价格), 主键为“客房号”列, 其他列都全部依赖于主键列。

3) 第三范式

第三范式(3NF)在第二范式的基础上更进一层, 第三范式是确保每列都和主键列直接相关, 而不是间接相关, 即限制列的冗余性。如果一个关系满足第二范式, 并且除了主键以外的其他列都依赖于主键列, 列和列之间不存在相互依赖关系, 则满足第三范式。

为了更好的理解第三范式, 这里我们需要了解传递依赖。假设 A、B 和 C 是关系 R 的三个属性, 如果 $A \rightarrow B$ 且 $B \rightarrow C$, 则从这些函数依赖中, 可以得出 $A \rightarrow C$ 。如上所述, 依赖 $A \rightarrow C$ 称之为传递依赖。

以第二范式中的客房信息表为例, 初看该表时没有问题, 满足第三范式, 每列都和主键列“客房号”相关, 再细看会发现:

“床位数”列、“价格”列 \rightarrow “客房类型”列。

“客房类型”列 \rightarrow “客房号”列。

“床位数”列、“价格”列 \rightarrow “客房号”列

为了满足第三范式, 应该去掉“床位数”列, “价格”列和“客房类型”列, 将客房信息表分解为如下两个表。

客房表（客房号，客房描述，客房类型编号，客房状态，入住人数）

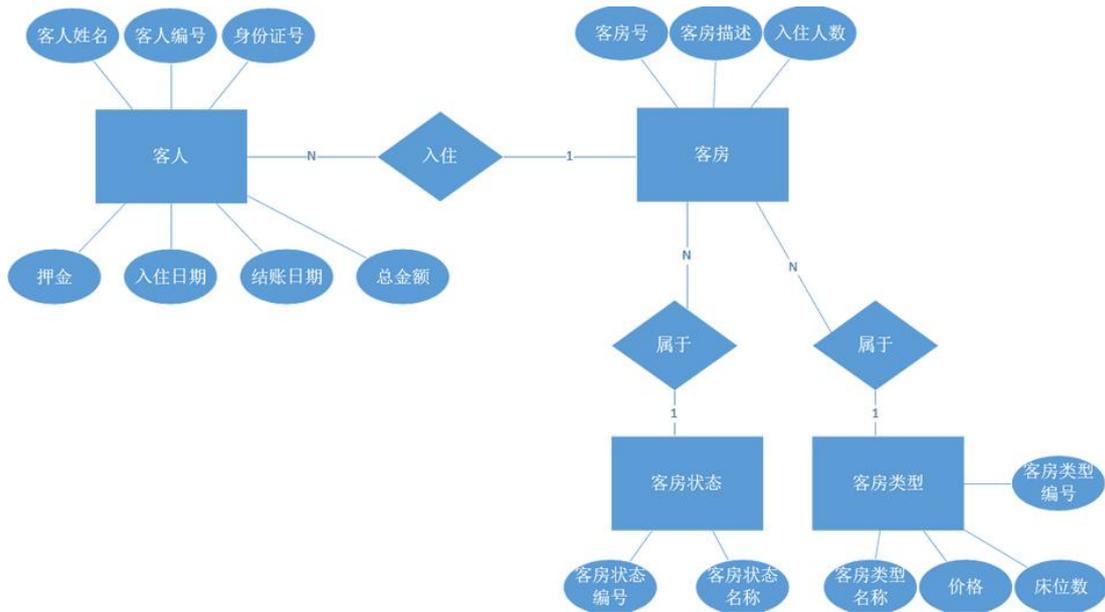
客房类型表（客房类型编号，客房类型名称，床位数，价格）

主键与外键在多表中的重复出现不属于数据冗余，非键字段的重复出现才是数据冗余。在客房表中客房状态存在冗余，需要进行规范化，规范化以后的表如下：

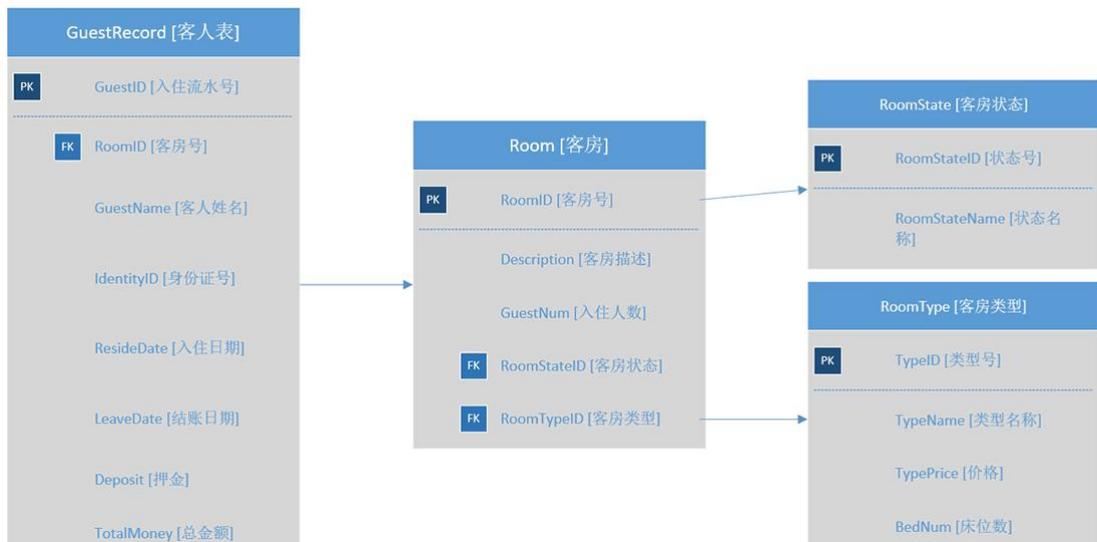
客房表（客房号，客房描述，客房类型编号，客房状态编号，入住人数）。

客房状态表（客房状态编号，客房状态名称）

最后，满足三大范式的 E-R 图如下所示：



满足三大范式的数据库模型图如下所示：



4) 反范式化

不满足范式的数据库设计，就是反范式化。

我们需要知道对于项目的最终用户来说，用户关心的是方便，清晰的数据结果。所以在设计数据库时，设计人员和客户在数据库的设计规范化和性能之间会有一些矛盾。

上面我们通过三大范式将客房表分解出两个表，为了满足客户的需求，最终可能需要通过三个或四个表之间的连接查询，来得到客户需要的数据结果，插入数据同样如此，对于客户输入的数据，我们需要分开插入到三个或四个不同的表中。

由此可以看出，为了满足三大范式，我们的数据操作性能会受到相应的影响。

所以，在实际的数据库设计中，既要考虑三大范式，避免数据的冗余和各种数据操作异常，又要考虑数据访问性能。为了减少表连接，提高数据库的访问性能，也可以允许适当的数据冗余列，这也许就是最合适的数据库设计方案。

比如，有一张存放商品的基本表，数据表中包括“单价”、“数量”“金额”等字段。“金额”这个字段就说明该表的设计不满足第三范式，因为“金额”可以由“单价”乘以“数量”得到，说明“金额”是冗余字段。

与第三范式中介绍的冗余相比，前面介绍的冗余属于低级冗余，我们反对低级冗余，但这里的冗余为高级冗余，目的是提高数据的处理速度，增加“金额”列后，可以提高查询统计的速度，这是以空间换取时间的做法。

注意：不要轻易违反数据库设计的规范化原则，如果处理不好，可能会适得其反，使应用程序运行速度更慢。

优缺点

最后我们来总结一下范式和反范式的优缺点。

1) 范式化

优点如下：

减少数据冗余

范式化后的表中只有很少的重复数据，更新时只需要更新较少的数据，所以范式化的更新操作比反范式化更快

范式化的表通常比反范式化更小

缺点如下：

范式化的表在查询时经常需要很多的关联，这会导致性能降低

增加了索引优化的难度

2) 反范式化

优点如下：

可以减少表的关联

可以更好的进行索引优化

缺点如下：

数据表存在数据冗余及数据维护异常

对数据的修改需要更多的成本