

80 | 程序员练级攻略：数据库

2018-07-05 陈皓

左耳听风

[进入课程 >](#)



讲述：柴巍

时长 11:09 大小 5.11M



对于数据库方向，重点就是两种数据库，一种是以 SQL 为代表的关系型数据库，另一种是以非 SQL 为代表的 NoSQL 数据库。关系型数据库主要有三个：Oracle、MySQL 和 Postgres。

在这里，我们只讨论越来越主流的 MySQL 数据库。首先，我们要了解数据库的一些实现原理和内存的一些细节，然后我们要知道数据的高可用和数据复制这些比较重要的话题，了解一下关系型数据库的一些实践和难点。然后，我们会进入到 NoSQL 数据库的学习。

NoSQL 数据库千奇百怪，其主要是解决了关系型数据库中的各种问题。第一个大问题就是数据的 Schema 非常多，用关系型数据库来表示不同的 Data Schema 是非常笨拙的，所以要有不同的数据库（如时序型、键值对型、搜索型、文档型、图结构型等）。另一个大问

题是，关系型数据库的 ACID 是一件很讨厌的事，这极大地影响了数据库的性能和扩展性，所以 NoSQL 在这上面做了相应的妥协以解决大规模伸缩的问题。

对于一个程序员，你可能觉得数据库的事都是 DBA 的事，然而我想告诉你你错了，这些事才真正是程序员的事。因为程序是需要和数据打交道的，所以程序员或架构师不仅需要设计数据模型，还要保证整体系统的稳定性和可用性，数据是整个系统中关键中的关键。所以，作为一个架构师或程序员，你必须了解最重要的数据存储——数据库。

关系型数据库

今天，关系型数据库最主要的两个代表是闭源的 Oracle 和开源的 MySQL。当然，还有很多了，比如微软的 SQL Server，IBM 的 DB2 等，还有开源的 PostgreSQL。关系型数据库的世界中有好多好多产品。当然，还是 Oracle 和 MySQL 是比较主流的。所以，这里主要介绍更为开放和主流的 MySQL。

如果你要玩 Oracle，我这里只推荐一本书《[Oracle Database 9i/10g/11g 编程艺术](#)》，无论是开发人员还是 DBA，它都是必读的书。这本书的作者是 Oracle 公司的技术副总裁托马斯·凯特（Thomas Kyte），他也是世界顶级的 Oracle 专家。

这本书中深入分析了 Oracle 数据库体系结构，包括文件、内存结构以及构成 Oracle 数据库和实例的底层进程，利用具体示例讨论了一些重要的数据库主题，如锁定、并发控制、事务等。同时分析了数据库中的物理结构，如表、索引和数据类型，并介绍采用哪些技术能最优地使用这些物理结构。

学习 MySQL，首先一定是要看[MySQL 官方手册](#)。

然后，官方还有几个 PPT 也要学习一下。

[How to Analyze and Tune MySQL Queries for Better Performance](#)

[MySQL Performance Tuning 101](#)

[MySQL Performance Schema & Sys Schema](#)

[MySQL Performance: Demystified Tuning & Best Practices](#)

[MySQL Security Best Practices](#)

[MySQL Cluster Deployment Best Practices](#)

[MySQL High Availability with InnoDB Cluster](#)

然后推荐《[高性能 MySQL](#)》，这本书是 MySQL 领域的经典之作，拥有广泛的影响力。不但适合数据库管理员（DBA）阅读，也适合开发人员参考学习。不管是数据库新手还是专家，都能从本书中有所收获。

如果你对 MySQL 的内部原理有兴趣的话，可以看一下这本书《[MySQL 技术内幕：InnoDB 存储引擎](#)》。当然，还有官网的[MySQL Internals Manual](#)。

数据库的索引设计和优化也是非常关键的，这里还有一本书《[数据库的索引设计与优化](#)》也是很不错的。虽然不是讲 MySQL 的，但是原理都是相通的。这也是上面推荐过的《高性能 MySQL》在其索引部分推荐的一本好书。

你千万不要觉得只有做数据库你才需要学习这种索引技术。不是的！在系统架构上，在分布式架构中，索引技术也是非常重要的。这本书对于索引性能进行了非常清楚的估算，不像其它书中只是模糊的描述，你一定会收获很多。

下面还有一些不错的和 MySQL 相关的文章。

[MySQL 索引背后的数据结构及算法原理](#)

[Some study on database storage internals](#)

[Sharding Pinterest: How we scaled our MySQL fleet](#)

[Guide to MySQL High Availability](#)

[Choosing MySQL High Availability Solutions](#)

[High availability with MariaDB TX: The definitive guide](#)

最后，还有一个 MySQL 的资源列表 [Awesome MySQL](#)，这个列表中有很多的工具和开发资源，可以帮助你做很多事。

MySQL 有两个比较有名的分支，一个是 Percona，另一个是 MariaDB，其官网上的 Resources 页面中有很多不错的资源和文档，可以经常看看。 [Percona Resources](#)、[MariaDB Resources](#)，以及它们的开发博客中也有很多不错的文章，分别为 [Percona Blog](#) 和 [MariaDB Blog](#)。

然后是关于 MySQL 的一些相关经验型的文章。

[Booking.com: Evolution of MySQL System Design](#)，Booking.com 的 MySQL 数据库使用的演化，其中有很多不错的经验分享，我相信也是很多公司会遇到的的问题。

[Tracking the Money - Scaling Financial Reporting at Airbnb](#) , Airbnb 的数据库扩展的经验分享。

[Why Uber Engineering Switched from Postgres to MySQL](#) , 无意比较两个数据库谁好谁不好, 推荐这篇 Uber 的长文, 主要是想让你从中学习到一些经验和技術细节, 这是一篇很不错的文章。

关于 MySQL 的集群复制, 下面有这些文章供你学习一下, 都是很不错的实践性比较强的文章。

[Monitoring Delayed Replication, With A Focus On MySQL](#)

[Mitigating replication lag and reducing read load with fredo](#)

另外, Booking.com 给了一系列的文章, 你可以看看:

[Better Parallel Replication for MySQL](#)

[Evaluating MySQL Parallel Replication Part 2: Slave Group Commit](#)

[Evaluating MySQL Parallel Replication Part 3: Benchmarks in Production](#)

[Evaluating MySQL Parallel Replication Part 4: More Benchmarks in Production](#)

[Evaluating MySQL Parallel Replication Part 4, Annex: Under the Hood](#)

对于 MySQL 的数据分区来说, 还有下面几篇文章你可以看看。

[StackOverflow: MySQL sharding approaches?](#)

[Why you don' t want to shard](#)

[How to Scale Big Data Applications](#)

[MySQL Sharding with ProxySQL](#)

然后, 再看看各个公司做 MySQL Sharding 的一些经验分享。

[MailChimp: Using Shards to Accommodate Millions of Users](#)

[Uber: Code Migration in Production: Rewriting the Sharding Layer of Uber' s Schemaless Datastore](#)

[Sharding & IDs at Instagram](#)

NoSQL 数据库

关于 NoSQL 数据库，其最初目的就是解决大数据的问题。然而，也有人把其直接用来替换掉关系型数据库。所以在学习这个技术之前，我们需要对这个技术的一些概念和初衷有一定的了解。下面是一些推荐资料。

Martin Fowler 在 YouTube 上分享的 NoSQL 介绍 [Introduction To NoSQL](#)，以及他参与编写的 [NoSQL Distilled - NoSQL 精粹](#)，这本书才 100 多页，是本难得的关于 NoSQL 的书，很不错，非常易读。

[NoSQL Databases: a Survey and Decision Guidance](#)，这篇文章可以带你自上而下地从 CAP 原理到开始了解 NoSQL 的种种技术，是一篇非常不错文章。

[Distribution, Data, Deployment: Software Architecture Convergence in Big Data Systems](#)，这是卡内基·梅隆大学的一篇讲分布式大数据系统的论文。其中主要讨论了在大数据时代下的软件工程中的一些关键点，也说到了 NoSQL 数据库。

[No Relation: The Mixed Blessings of Non-Relational Databases](#)，这篇论文虽然有点年代久远。但这篇论文是 HBase 的基础，你花上一点时间来读读，就可以了解到，对各种非关系型数据存储优缺点的一个很好的比较。

[NoSQL Data Modeling Techniques](#)，NoSQL 建模技术。这篇文章我曾经翻译在了 CoolShell 上，标题为 [NoSQL 数据建模技术](#)，供你参考。

[MongoDB - Data Modeling Introduction](#)，虽然这是 MongoDB 的数据建模介绍，但是其很多观点可以用于其它的 NoSQL 数据库。

[Firebase - Structure Your Database](#)，Google 的 Firebase 数据库使用 JSON 建模的一些最佳实践。

因为 CAP 原理，所以当你需要选择一个 NoSQL 数据库的时候，你应该看看这篇文档 [Visual Guide to NoSQL Systems](#)。

选 SQL 还是 NoSQL，这里有两篇文章，值得你看看。

[SQL vs. NoSQL Databases: What's the Difference?](#)

[Salesforce: SQL or NoSQL](#)

各种 NoSQL 数据库

学习使用 NoSQL 数据库其实并不是一件很难的事，只要你把官方的文档仔细地读一下，是很容易上手的，而且大多数 NoSQL 数据库都是开源的，所以，也可以通过代码自己解决问题。下面我主要给出一些典型的 NoSQL 数据库的一些经验型的文章，供你参考。

列数据库 Column Database

Cassandra 相关

沃尔玛实验室有两篇文章值得一读。

[Avoid Pitfalls in Scaling Cassandra Cluster at Walmart](#)

[Storing Images in Cassandra at Walmart](#)

[Yelp: How We Scaled Our Ad Analytics with Apache Cassandra](#)，Yelp 的这篇博客也有一些相关的经验和教训。

[Discord: How Discord Stores Billions of Messages](#)，Discord 公司分享的一个如何存储十亿级消息的技术文章。

[Cassandra at Instagram](#)，Instagram 的一个 PPT，其中介绍了 Instagram 中是怎么使用 Cassandra 的。

[Netflix: Benchmarking Cassandra Scalability on AWS - Over a million writes per second](#)，Netflix 公司在 AWS 上给 Cassandra 做的一个 Benchmark。

HBase 相关

[Imgur Notification: From MySQL to HBASE](#)

[Pinterest: Improving HBase Backup Efficiency](#)

[IBM : Tuning HBase performance](#)

[HBase File Locality in HDFS](#)

[Apache Hadoop Goes Realtime at Facebook](#)

[Storage Infrastructure Behind Facebook Messages: Using HBase at Scale](#)

[GitHub: Awesome HBase](#)

针对于 HBase 有两本书你可以考虑一下。

首先，先推荐两本书，一本是偏实践的《[HBase 实战](#)》，另一本是偏大而全的手册型的《[HBase 权威指南](#)》。

当然，你也可以看看官方的 [The Apache HBase™ Reference Guide](#)

另外两个列数据库：

[ClickHouse - Open Source Distributed Column Database at Yandex](#)

[Scaling Redshift without Scaling Costs at GIPHY](#)

文档数据库 Document Database - MongoDB, SimpleDB, CouchDB

[Data Points - What the Heck Are Document Databases?](#)

[eBay: Building Mission-Critical Multi-Data Center Applications with MongoDB](#)

[The AWS and MongoDB Infrastructure of Parse: Lessons Learned](#)

[Migrating Mountains of Mongo Data](#)

[Couchbase Ecosystem at LinkedIn](#)

[SimpleDB at Zendesk](#)

[Github: Awesome MongoDB](#)

数据结构数据库 Data structure Database - Redis

[Learn Redis the hard way \(in production\) at Trivago](#)

[Twitter: How Twitter Uses Redis To Scale - 105TB RAM, 39MM QPS, 10,000+ Instances](#)

[Slack: Scaling Slack's Job Queue - Robustly Handling Billions of Tasks in Milliseconds Using Kafka and Redis](#)

[GitHub: Moving persistent data out of Redis at GitHub](#)

[Instagram: Storing Hundreds of Millions of Simple Key-Value Pairs in Redis](#)

[Redis in Chat Architecture of Twitch \(from 27:22\)](#)

[Deliveroo: Optimizing Session Key Storage in Redis](#)

[Deliveroo: Optimizing Redis Storage](#)

[GitHub: Awesome Redis](#)

时序数据库 Time-Series Database

[What is Time-Series Data & Why We Need a Time-Series Database](#)

[Time Series Data: Why and How to Use a Relational Database instead of NoSQL](#)

[Beringei: High-performance Time Series Storage Engine @Facebook](#)

[Introducing Atlas: Netflix' s Primary Telemetry Platform @Netflix](#)

[Building a Scalable Time Series Database on PostgreSQL](#)

[Scaling Time Series Data Storage - Part I @Netflix](#)

[Design of a Cost Efficient Time Series Store for Big Data](#)

[GitHub: Awesome Time-Series Database](#)

图数据库 - Graph Platform

首先是 IBM Developerworks 上的两个简介性的 PPT。

[Intro to graph databases, Part 1, Graph databases and the CRUD operations](#)

[Intro to graph databases, Part 2, Building a recommendation engine with a graph database](#)

然后是一本免费的电子书《[Graph Database](#)》。

接下来是一些图数据库的介绍文章。

[Handling Billions of Edges in a Graph Database](#)

[Neo4j case studies with Walmart, eBay, AirBnB, NASA, etc](#)

[FlockDB: Distributed Graph Database for Storing Adjacency Lists at Twitter](#)

[JanusGraph: Scalable Graph Database backed by Google, IBM and Hortonworks](#)

[Amazon Neptune](#)

搜索数据库 - ElasticSearch

[Elasticsearch: The Definitive Guide](#) 这是官网方的 ElasticSearch 的学习资料，基本上来说，看这个就够了。

接下来是 4 篇和性能调优相关的工程实践。

[Elasticsearch Performance Tuning Practice at eBay](#)

[Elasticsearch at Kickstarter](#)

[9 tips on ElasticSearch configuration for high performance](#)

[Elasticsearch In Production - Deployment Best Practices](#)

最后是 GitHub 上的资源列表 [GitHub: Awesome ElasticSearch](#)。

小结

好了，总结一下今天分享的内容。虽然有人会认为数据库与程序员无关，是 DBA 的事儿。但我坚信，数据库才真正是程序员的事儿。因为程序是需要和数据打交道的，所以程序员或架构师不仅需要设计数据模型，还要保证整体系统的稳定性和可用性，数据是整个系统中关键中的关键。

对于数据库方向，重点就是两种数据库，一种是以 SQL 为代表的关系型数据库，另一种是以非 SQL 为代表的 NoSQL 数据库。因而，在这篇文章中，我给出了 MySQL 和各种开源 NoSQL 的一些相关的有价值的文章和导读，主要是让你对这些数据库的内在有一定的了解，但又不会太深。同时给出了一些知名企业使用数据库的工程实践，这对于了解各种数据库的优劣非常有帮助，值得认真读读。

从下篇文章开始，我们将进入分布式系统架构方面的内容，里面不仅涵盖了大量的理论知识，更有丰富的入门指导和大量的工程实践。敬请期待。

下面是《程序员练级攻略》系列文章的目录。

[开篇词](#)

入门篇

[零基础启蒙](#)

[正式入门](#)

修养篇

[程序员修养](#)

专业基础篇

[编程语言](#)

[理论学科](#)

[系统知识](#)

软件设计篇

[软件设计](#)

高手成长篇

[Linux 系统、内存和网络（系统底层知识）](#)

[异步 I/O 模型和 Lock-Free 编程（系统底层知识）](#)

[Java 底层知识](#)

[数据库](#)

[分布式架构入门（分布式架构）](#)

[分布式架构经典图书和论文（分布式架构）](#)

[分布式架构工程设计（分布式架构）](#)

[微服务](#)

[容器化和自动化运维](#)

[机器学习和人工智能](#)

[前端基础和底层原理（前端方向）](#)

[前端性能优化和框架（前端方向）](#)

[UI/UX 设计（前端方向）](#)

[技术资源集散地](#)

左耳朵耗子

全年独家专栏《左耳听风》

20000 名程序员的练级攻略

陈皓 资深技术专家
骨灰级程序员



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

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

上一篇 79 | 程序员练级攻略：Java底层知识

下一篇 81 | 程序员练级攻略：分布式架构入门

精选留言 (26)

 写留言



Damon

2018-07-05

 13

链接的质量很高，但读者更想看耗子的理解和总结吧，希望多点描述性文字。

作者回复: 在高手篇开篇描述中，我有如下的描述：

我假设你在前面已经打下了非常扎实的基础，但是要成为一个高手，基础知识只是一个地基，你还需要很多更为具体的技术。对我来说，就是看各种各样的文章、手册、论文、分享..... 其实，学习到一定程度，就是要从书本中走出去，到社区里和大家一起学习，而且还需要自己找食吃了。

【所以，对于这里面的文章，有很多都是在罗列各种文章和资源，只是为你梳理信息源，而不是喂你吃饭】。

老实说，我已经为你梳理并过滤掉了很多的信息，这里只留下了 30% 我觉得最经济也最有价值的信息。虽然对于不同定位和不同需求的人还可以再对这些信息进行删减，但是觉得我这么一做就会对其它人不公平了。【所以，这也是我觉得最小数量集的信息和资源吧。你也可以把我这里的東西当成一个索引来对待】。



少年姜太公

2018-07-07

👍 9

耗子叔，除了阅读书籍、博客和论文，有没有开源代码阅读经验分享。我的体会是阅读优秀开源代码就好比阅读经典文学作品，从中学习编程风格，设计思想以提高自身代码编写质量，偶尔还能把某些开源项目的代码片段、编写思路应用到自己代码中。求耗子叔分享一下阅读开源项目和提高编程水平的经验

展开 ∨



yanfan_so...

2018-07-05

👍 9

会用数据库并不代表就能设计好数据库，设计数据库与业务关联，需要了解用户需求和业务联系，这篇文章主要是介绍各个数据库的而非设计，但也没提太多底层，数据库优化，希望耗子叔可以抽空给讲讲这些。



杨书清

2018-07-05

👍 6

感觉对普通程序猿来说数据库的建模和设计也很重要，有这方面的资料推荐么？



刘強

2018-07-07

👍 5

资料看多了，我对高手有了不同的理解。我觉得真正的高手是有自己的想法，自己的个性，独特的思维，解决问题的方式。看书再多，都理解了，都掌握了，顶多也不过是嚼别人嚼过的东西。这样的人貌似比大多数人强，其实他只不过是比别人看的多而已。有自己的思考，有自己独特的思想，有自己的输出，进而影响到别人，这才叫高手。我认为黑客才是真正的高手。这里的黑客是黑客与画家里面的定义。

展开 ∨



songyy

👍 4



2018-07-10

我最近在读 《designing data intensive application》其中深入浅出地比较了数据库的基本算法，解决数据方面问题的模式 感觉内容很棒

展开 ∨



1 Player

2018-07-05

👍 4

皓叔，有些东西学习过后，一段时间不用就会忘了。怎么解决这种问题呢。



无风

2018-07-05

👍 4

这一系列看下，真的很难想象皓哥是如何做到如此广面的知识积累

作者回复: 我至少看了3倍于这些资料的量，才给出这些东西。这主要是兴趣和坚持



刘-阿-伟

2018-07-05

👍 3

我报一下坐标，深圳南山四年Java，老师程序员练级攻略我95%都不会，是不是我太菜了！



高宇

2018-07-06

👍 2

读者交流群申请表提交了没有回复，也不知道具体什么情况



木易

2018-08-11

👍 1

大数据方面的会写一篇吗，hadoop，spark等

展开 ∨



50infiveda...

2018-07-05

👍 1

怎么加读者群呀 我关注公众号也回复了公众号 还是不行



zzz

2018-07-05

👍 1

耗子叔，请教一个问题啊，go 能开发网站和写爬虫吗？我个人初步判断是，可以做网站，因为有web 框架，也可以写爬虫，因为能解析HTML。求解~

展开 ∨



weilai

2019-03-18

👍

<https://downloads.mysql.com/docs/refman-8.0-en.pdf>



godtrue

2019-01-09

👍

优质知识集散地

展开 ∨



caohuan

2018-10-26

👍

回看 自己做的项目，都是应用层上的 增删改查 以及 优化的功能，一旦涉及到 底层一点的知识，完全懵逼，所以也一直徘徊在 功能开发的码农层次，看到 耗子哥 列出的一大串资源，知道自己应该 要再次入门，甩掉身上的赘肉，从用到的MySQL开始。

展开 ∨



张飞洪

2018-09-29

👍

关系型数据库主要有三个：Oracle、MySQL 和 Pg？请问耗子叔，那微软的SQL Server为什么没有罗列在内？SQL Server性能比Mysql不出名还是比较差？多指教，谢谢！



luwei

2018-09-08

👍

不知道耗子叔以后会不会出一篇有关数据库和内存数据同步问题的文章？比如某个积分变

换，先更新的内存数据，然后更新数据库的值，但更新数据库的时候写失败了怎么办？



penn

2018-09-06



请问皓哥，在高手成长篇里，哪些是和现在火热的所谓大数据比较相关的？



文盲筱烨

2018-07-17



陈老师，请教下，pg数据库未来的发展 数据库有点群雄四起

作者回复: 我对pg不熟悉，我只知道pg还没到主流数据库的程度。未来是否有大发展还看是否有大公司支持，是否有杀手级应用，目前还看不到。

