

程序员练级攻略（2018）：数据库

2018-07-05 陈皓





程序员练级攻略（2018）：数据库

陈皓

- 00:01 / 11:08

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](#)，这个列表中有许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页面中有许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许多不错的文章，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](#)
- [Airbnb: How We Partitioned Airbnb's Main Database in Two Weeks](#)

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，是一篇非常不错的文章。
- [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下的一些关键点，也说了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](#)
- [Beringel: 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 1 @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（2018）》系列文章的目录（持续更新中）。

- [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）](#)
 -



Damon	2018-07-05
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	2018-07-05
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，只是为你梳理信息来源，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。你也可以把我这的东西当成一个索引来对待】。	
李文	2018-07-07
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	
杨书清	2018-07-05
感觉对普通程序员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？	
刘强	2018-07-07
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。	
yanfan_software	2018-07-05
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。	
无风	2018-07-05
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	2018-07-05
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	
高宇	2018-07-06
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	
50infivedays	2018-07-05
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
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 ~	
文盲筱晔	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	2018-07-18
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。	
KingPoker	2018-07-16
如果有精力的可以看j gray 「事务处理：概念与技术」	
Coder4	

公司是tidb付费用户，1亿数据用来存类似log的不关键离线数据，单条sql返回row多了(不是并发)db服务就挂只能重启，可用性非常差。好在不是关键数据，已经迁移到es。	2018-07-12
songgy	
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	2018-07-10
Geek_217683	
耗子哥，能不能谈谈你对编辑器选择的看法，Vim,Emacs.是否还需要？	2018-07-06
华子	
学好技术没有好的英文阅读能力是不行的	2018-07-06
1 Player	
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。	2018-07-05
阳	
请问，怎么评价tidb	2018-07-05
作者回复	
这个问题练级攻略无关。	2018-07-05
刘-阿-伟	
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！	2018-07-05
ricktian	
国内公司pingcap的tidb最近好像也很火，皓哥有研究不。有准备学习下这块	2018-07-05

