

TURING

图灵程序设计丛书

위키북스

【韩】李成旭 著
武传海 译



深入理解 MariaDB与MySQL



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书



深入理解 MariaDB与MySQL

【韩】李成旭 著
武传海 译

人民邮电出版社
北京

图书在版编目(CIP)数据

深入理解 MariaDB 与 MySQL / (韩) 李成旭著; 武传海译. -- 北京: 人民邮电出版社, 2015.12

(图灵程序设计丛书)

ISBN 978-7-115-40908-9

I. ①深… II. ①李… ②武… III. ①关系数据库系统 IV. ①TP311.138

中国版本图书馆CIP数据核字(2015)第267480号

内 容 提 要

本书讲解了MariaDB的诞生背景及设计目的,并通过与MySQL的比较使读者轻松理解MariaDB。本书内容涉及MariaDB 5.1~10.0的各个方面,讲解重点在于MariaDB 10.0的各种实用功能。此外,MariaDB与MySQL有着紧密的联系,书中对MariaDB 10.0与MySQL 5.6的新功能进行了比较,读者可以同时学习MariaDB与MySQL的内容。

◆ 著 [韩]李成旭

译 武传海

责任编辑 傅志红

执行编辑 陈 曦

责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 28.5

字数: 622千字 2015年12月第1版

印数: 1-3 000册 2015年12月北京第1次印刷

著作权合同登记号 图字: 01-2014-6026号

定价: 89.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第0021号

版权声明

Real MariaDB

Copyright 2014 © by 이성욱 李成旭

ALL rights reserved

Simplified Chinese copyright © 2015 by POSTS & TELECOM PRESS

Simplified Chinese language edition arranged with WIKIBOOKS

through Eric Yang Agency Inc.

本书中文简体字版由 WIKIBOOKS 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

推荐语

为了提升 MariaDB 性能，MariaDB 基金会承诺引入优化器等存储引擎和各种改进手段、管理功能。

另外，我们也会持续发展 MariaDB 社区，并通过开源倡导者和许多其他人的参与，继续达成 MariaDB（其基础也在不断发展）开源的目标。

围绕着 MariaDB 基金会，许多全球性的 IT 公司如淘宝和谷歌等，也已加入 MariaDB 这个热闹的大家庭。我坚信，未来还会有很多公司期望参与 MariaDB 生态系统的建设。

我希望在韩国有更多人加入 MariaDB 社区，也有更多公司开始使用 MariaDB。

我个人感谢并祝贺李成旭在韩国出版了第一本关于 MariaDB 的书，并希望更多 DBA 从本书中体会到 MariaDB 的特性和功能之美。

MariaDB 基金会

CTO Michael “Monty” Widenius



MariaDB 首席宣传官

Colin Charles



前言

我从 2010 年左右开始写作《MySQL 权威指南》一书，与其他商用 RDBMS 相比，MySQL 服务器的用途相当明确。使用 MySQL 服务器的环境中，即使部分数据发生损坏或者服务器突然崩溃，也不会有什么大问题。因此，许多数据库管理员对 MySQL 抱着“MySQL 服务器就应该这样啊！”的看法。MySQL 服务器大量应用于无专门 DBA 的初创企业，通常由忙于业务开发的人进行管理。而且 MySQL 服务器通常运行在硬件与配置不是非常好的环境中，这就导致在此类环境中使用过 MySQL 服务器的大部分用户会对其有先入为主的偏见，认为它是一个“非常奇怪的数据库”。

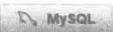
《深入理解 MariaDB 与 MySQL》原书出版于 2014 年 4 月，现在，MySQL 不再是那种不稳定的免费数据库了，也不只在小规模的初创企业或门户网站中使用。目前，连支付系统都在使用 MySQL 服务器，原来只使用 MSSQL 或 Oracle DBMS 的游戏服务也逐渐开始使用 MySQL 服务器。不仅是小规模移动游戏，甚至 MMORPG 这样的大型游戏也表现出使用 MySQL 服务器的趋势。Oracle 或 MSSQL 服务器昂贵的许可价格在客观上不断壮大 MySQL 服务器的用户队伍，但最重要的原因还在于 MySQL 服务器本身的可靠性以及超高的性价比，这是毋庸置疑的。希望本书能为 MySQL 的进一步普及尽一份绵薄之力。

不知不觉，《MySQL 权威指南》已经出版三年多了。在此期间，MySQL 市场也开始呈现多样化，其中最具代表性的是 PerconaServer 以及本书要讲解的 MariaDB。PerconaServer 从属于 Oracle 发布的 MySQL 服务器系列，但 MariaDB 在功能或设计哲学上表现出相当独立的发展方向，相信这种趋势会越来越明显。

目前，谷歌与中国最大的网上商城——“淘宝”开始为 MariaDB 贡献源代码。对这些企业巨头而言，MariaDB 的先天限制就是其优点。MariaDB 是在 Oracle MySQL 源代码基础上开发而成的，因而不能像 Oracle 一样轻易地商业化。也就是说，它不可能像 MySQL 企业版一样，将源代码封闭起来以发布商业付费版本。对谷歌和淘宝这类需要修改 MySQL 源代码并使用的企业而言，MySQL 社区版只有部分功能可用，远不如 MariaDB 更有吸引力。这类技术实力雄厚的大型企业选用 MariaDB 不仅意味着其是值得信赖的，而且他们开发并改善的各种功能可以免费提供给每个用户。

本书面向对 MariaDB 与 MySQL 感兴趣的读者，讲解时采用比较说明的方式，对 MariaDB 10.0 与 MySQL 5.6 的个性和共性进行介绍。通过阅读本书，各位应当能够理解 MariaDB 与 MySQL 的特点以及二者之间的功能差异，并能根据实际提供的服务选择合适的解决方案。

目录

第1章 MariaDB	1
1.1 MariaDB	2
1.2 MariaDB 与 MySQL	3
1.2.1 MariaDB、MySQL 和 PerconaServer	3
1.2.2 相同点	6
1.2.3 不同点	7
1.2.4 各版本间的兼容性	10
1.2.5 性能比较	11
1.2.6 MariaDB 是否会取代 MySQL	14
1.2.7 MariaDB 与 MySQL 的选择	14
第2章 安装	17
2.1 下载	18
2.2 安装	20
2.2.1 在 Windows 操作系统下安装 MariaDB	21
2.2.2 在 Linux 操作系统下安装 MariaDB	27
2.3 更新	28
2.3.1 从 MySQL 升级到 MariaDB 时的注意事项	28
2.3.2 不受版本限制的安全升级法	29
2.3.3 从 MySQL 5.0 或之前版本升级为 MariaDB 5.5	30
2.3.4 从 MySQL 5.1 升级为 MariaDB 5.5	31
2.3.5 从 MySQL 5.5 升级为 MariaDB 5.5	32
第3章 MariaDB 启动及执行查询	33
3.1 启动与终止服务器  	34
3.1.1 在 Linux 中启动与终止 MariaDB 服务器	34
3.1.2 在 Windows 系统中启动与终止 MariaDB 服务器	36

3.2 登录服务器	 	37
3.2.1 登录服务器与版本确认		37
3.2.2 mysql 客户端程序支持的选项		39
3.3 创建数据库与用户	 	41
3.3.1 MariaDB 用户账户识别与权限		41
3.3.2 MariaDB 默认用户		47
3.3.3 MariaDB 的默认数据库		48
3.3.4 新建数据库		50
3.3.5 创建用户		50
3.4 创建与修改数据表	 	51
3.4.1 创建数据表		52
3.4.2 修改数据表 (在线与离线)		53
3.4.3 删除数据表		66
3.5 数据操作	 	67
3.5.1 INSERT		67
3.5.2 SELECT		68
3.5.3 UPDATE		70
3.5.4 REPLACE		71
3.5.5 DELETE		71
第4章 执行计划分析		73
4.1 概要	 	74
4.1.1 查询执行过程		74
4.1.2 优化器种类		75
4.1.3 统计信息		76
4.1.4 直方图统计信息		82
4.1.5 连接优化器选项		87
4.2 准备示例数据		92
4.2.1 加载示例数据		92
4.2.2 收集统计信息		94
4.3 执行计划分析	 	96
4.3.1 id 列		97
4.3.2 select_type 列		98
4.3.3 table 列		106

4.3.4 type 列	108
4.3.5 possible_keys 列	117
4.3.6 key 列	117
4.3.7 key_len 列	117
4.3.8 ref 列	119
4.3.9 rows 列	120
4.3.10 Extra 列	121
4.3.11 EXPLAIN EXTENDED (Filtered 列)	149
4.3.12 EXPLAIN EXTENDED (附加优化器信息)	150
4.3.13 EXPLAIN PARTITIONS (Partitions 列)	151

4.4 优化器提示

4.4.1 提示的使用方法	152
4.4.2 STRAIGHT_JOIN	153
4.4.3 USE INDEX/FORCE INDEX/IGNORE INDEX	156
4.4.4 SQL_CACHE/SQL_NO_CACHE	157
4.4.5 SQL_CALC_FOUND_ROWS	158
4.4.6 其他提示	160

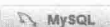
4.5 分析执行计划时的注意事项

4.5.1 Select_type 列中需要注意的项目	160
4.5.2 Type 列中需要注意的项目	161
4.5.3 Key 列中需要注意的项目	161
4.5.4 Rows 列中需要注意的项目	161
4.5.5 Extra 列中需要注意的项目	161

第5章 优化

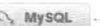
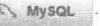
5.1 全表扫描

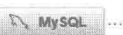
5.2 ORDER BY 处理 (Using filesort)

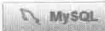
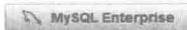
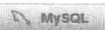
5.2.1 排序缓冲	165
5.2.2 排序算法	167
5.2.3 排序处理方式	169
5.2.4 ORDER BY..LIMIT n 优化 	177
5.2.5 与排序相关的状态变量	178

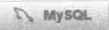
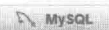
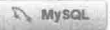
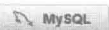
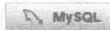
5.3 GROUP BY 处理

5.3.1 使用索引扫描处理 GROUP BY (紧凑索引扫描)	179
5.3.2 使用松散索引扫描处理 GROUP BY	179
5.3.3 使用临时表处理 GROUP BY	181

5.4	DISTINCT 处理	 	182
5.4.1	SELECT DISTINCT...		182
5.4.2	DISTINCT 用于集合函数内部		183
5.5	临时表 (Using temporary)	 	184
5.5.1	需要使用临时表的查询		185
5.5.2	在磁盘上创建临时表 (使用 Aria 存储引擎)		185
5.5.3	与临时表有关的状态变量		186
5.5.4	带索引的内部临时表		187
5.5.5	内部临时表的注意事项		187
5.6	索引条件下推	 	188
5.7	多范围读	 	191
5.7.1	基于 RowId 排序 (Rowid-orderd scan)		192
5.7.2	基于 Key 排序 (Key-ordered scan)		194
5.7.3	基于 Key 与 RowId 排序 (Key-ordered、Rowid-ordered scan)		195
5.7.4	“多范围读”优化与排序		197
5.7.5	“多范围读”优化的注意事项		198
5.8	索引合并	 	199
5.8.1	Using union		201
5.8.2	Using sort_union		202
5.8.3	Using intersect		204
5.8.4	Using sort_intersect		205
5.9	数据表连接	 	205
5.9.1	连接类型		206
5.9.2	连接算法		212
5.9.3	连接的注意事项		222
5.10	子查询	 	224
5.10.1	半连接子查询优化		224
5.10.2	非半连接的子查询优化		236
5.10.3	子查询缓存		238
第6章	存储引擎		241
6.1	Aria 存储引擎		242
6.1.1	事务		242

6.1.2	页面缓存	243
6.1.3	系统变量设置	243
6.2	XtraDB 存储引擎 	244
6.3	InnoDB 存储引擎  	247
6.3.1	MySQL 5.6 InnoDB	247
6.3.2	“脏页”刷新	262
6.3.3	缓冲池性能改善 	279
6.3.4	原子写 (针对 FusionIO SSD 的 Atomic write) 	283
6.3.5	扩展的 InnoDB 引擎状态输出 	284
6.3.6	XtraDB 重做日志存档	288
6.3.7	跟踪修改页面 	289
6.4	全文检索引擎  	290
6.4.1	添加全文检索索引	290
6.4.2	全文检索索引表空间	292
6.4.3	与全文检索索引相关的 INFORMATION_SCHEMA 信息	294
6.4.4	使用全文检索索引	295
6.4.5	注意事项	295
6.5	Memcached 插件 	298
6.5.1	架构	298
6.5.2	安装与测试	299
6.5.3	缓存策略	303
6.5.4	注册用户数据表	304
6.5.5	相关系统变量	305
6.6	Cassandra 存储引擎 	306
6.6.1	Cassandra	307
6.6.2	Cassandra 存储引擎	309
6.7	CONNECT 存储引擎 	311
6.7.1	安装 CONNECT 存储引擎	311
6.7.2	连接 Oracle RDBMS 数据表	312
6.7.3	连接 my.cnf 设置文件	313
6.7.4	连接操作系统目录	314
6.8	Sequence 存储引擎 	315
6.8.1	Sequence 存储引擎的基本用法	315
6.8.2	查找遗漏的编号	317
6.8.3	创建顺序组合的号码对	318

6.8.4	查找倍数或公倍数	318
6.8.5	顺序生成字母表	319
6.8.6	顺序生成日期	320
6.8.7	数据复制加工	320
6.9	Mroonga 全文检索存储引擎  	321
6.9.1	索引算法	322
6.9.2	安装 Mroonga 全文检索引擎	324
6.9.3	使用 Mroonga 全文检索引擎	325
第7章	其他功能	329
7.1	性能提升  	330
7.2	管理与诊断 	335
7.2.1	SHOW EXPLAIN FOR<THREAD-ID>	335
7.2.2	向慢查询日志输出执行计划	337
7.2.3	输出结构化的执行计划	338
7.2.4	线程的内存使用量	340
7.2.5	SHUTDOWN 命令	341
7.2.6	强制终止用户或查询执行	341
7.2.7	GET DIAGNOSTICS	343
7.3	开发效率 	344
7.3.1	LIMIT ROWS EXAMINED	344
7.3.2	DELETE...RETURNING...	346
7.3.3	以微秒为单位保存时间	347
7.3.4	设置 DATETIME 类型的默认值	349
7.3.5	扩展正则表达式功能	349
7.3.6	虚拟列	352
7.3.7	动态列	354
7.4	分区  	358
7.4.1	显式指定分区	358
7.4.2	交换分区表空间	362
7.5	备份  	364
7.5.1	二进制日志远程备份	365
7.5.2	XtraBackup 介绍	368
7.5.3	XtraBackup 的功能	381

7.5.4	XtraBackup 的高级用法	386
第8章	复制	397
8.1	全局事务 ID  	398
8.1.1	全局事务 ID 的定义	398
8.1.2	全局事务 ID 的必要性	400
8.1.3	MariaDB 10.0 全局事务 ID 	403
8.1.4	MySQL 5.6 全局事务 ID 	413
8.2	多源复制 	421
8.2.1	多源复制相关命令	422
8.2.2	构建多源复制	423
8.2.3	多源复制与全局事务	426
8.3	多线程复制  	428
8.3.1	MySQL 5.6 的多线程复制	428
8.3.2	MariaDB 10.0 的多线程复制	431
8.4	Crash safe slave  	433
8.4.1	MariaDB 10.0 的崩溃安全复制	434
8.4.2	MySQL 5.6 的崩溃安全复制	435
8.5	改善基于 ROW 的复制功能  	435
8.5.1	ROW 格式的容量优化 	435
8.5.2	ROW 格式二进制日志的信息性日志事件	436
8.6	延迟复制 	439
8.7	MariaDB 与 MySQL 服务器间的复制	440
8.8	其他功能改善  	441
8.8.1	二进制日志校验和	441
8.8.2	二进制日志 API	442
8.8.3	提交二进制日志组	443

第 1 章

MariaDB

1.1 MariaDB

1.2 MariaDB与MySQL

1.1 MariaDB

MariaDB 是 Monty Program AB 公司推出的一款独立的 RDBMS，类似于 MySQL 与 Oracle 数据库管理系统。注意，此处的“独立”并不是指它可以通过插件方式使用 InnoDB 等存储引擎。事实上，很难用三言两语讲清楚 MariaDB。下面看看 MariaDB 数据库的几个主要特征，通过这些特征，各位可以大致了解 MariaDB 数据库。

1. MariaDB 诞生于 MySQL Community 代码数据库。

MariaDB 是在 MySQL（开源的 Community 版本）代码基础上创建而成的 RDBMS。MariaDB 各个版本分别基于 MySQL Community 的对应版本，后面会详细说明。但是，MariaDB 并不是原封不动地使用 MySQL Community 版本代码，Monty Program AB 公司在保证它与 MySQL Community 版本拥有相同功能的前提下，添加了新功能。由于 MariaDB 建立在 MySQL Community 版本之上，因而两者之间具有良好的兼容性，在很多方面是相通的。

2. MariaDB 是由 Monty Program AB 维护的 MySQL。

Monty Program AB 是 MySQL 创始人 Michael “Monty” Widenius（下称“Monty”）2009 年创建的公司。MySQL 先由 Sun Microsystems 收购，后来又落入 Oracle 之手。MySQL 的许可证政策也随之发生变化，开发人员遭到大量替换。据说，许多 MySQL 的核心开发人员转投到 Monty Program AB 公司工作。Oracle 发布 MySQL Community 新版本时，Monty Program AB 的 MySQL 专家都会改善既有代码，添加额外功能，发布 MariaDB 新版本。因此，可以将 MariaDB 视为 MySQL Community 相应版本的增强版本。

3. MariaDB 是开源数据库。

与以前一样，Oracle MySQL Community 版本仍然是开源的。许多用户都有这样的疑问：“既然 MySQL Community 版本是开源的，那为什么还要推出另外一个开源的 RDBMS 呢？”事实上，Oracle MySQL Community 版本是其开源商业模式的一部分^①。开源商业模式中，并非一切都是免费的，部分需要付费，类似于部分付费游戏。换言之，在部分付费游戏中玩游戏是免费的，但是要想使用更多装备、技能就需要支付一定的费用。MySQL 与此类似，基本功能是免费的，而要使用更多高级功能就要支付费用，这种版本称为 MySQL 企业版。但是 MariaDB 没有这种区别^②，用户不必支付额外费用以获得更多软件功能，所需功能可以随时下载并免费使用。具有一定能力的用户也可以下载源代码，通过源码分析了解各项功能实现的细节。当然，不包含软件之外的技术支持等服务。

MySQL 创始人 Monty 推出 MariaDB 开源数据库的原因如下：

- 希望 MySQL Code Base（以 MariaDB 的名义）继续保持开源；

① 当然，我们并不是对 Oracle 商业模式持否定态度，使用更好、更强大的功能当然需要付费。

② 现在，MariaDB 也推出了相应的企业版，使用时也需要支付一定费用。——译者注

- 希望为 MySQL 开发者搭建一个更好的平台，保证开发人员继续以开源方式开发 MySQL。

Monty Program AB 公司未来可能也会被其他公司收购，并因此取消开源。但是，从上面两点看，我认为：只要 Monty 还在，MariaDB 数据库就将继续保持开源，用户必需的功能也可以继续免费使用。此外，MariaDB 是在 MySQL 社区版基础上开发而成的，这将为 MariaDB 商业版的推出带来明显的法律风险。因为 Monty Program AB 公司并未从 Oracle 公司重新购回 MySQL。

本节对 MariaDB 进行了简单的介绍，但这对了解整个 MariaDB 还是远远不够的。由于本书并非主要阐述 MariaDB 与开源哲学，所以各位大致了解上面这些内容就可以了。

提示

“MariaDB”这个名称究竟从何而来？听着好像有点宗教意味，而作为 DBMS 的软件名称也有点不自然。要想弄清“MariaDB”源自何处，首先要了解 MySQL 是如何创建的，还要对 Monty 的家庭有一些了解，虽然这对我们并不重要。

很多人可能会对“MySQL”中的“My”一词产生误解，以为它是“I”（我）的所有格形式，即“我的”，于是将“MySQL”理解为“我的 SQL”。其实不然，“My”一词来自 Monty 的第一个女儿的名字。Monty 曾用二儿子名 Max 创建过名为 MaxDB 的 DBMS。介绍到这里，相信很多读者已经猜到了，MariaDB 源自 Monty 的第三个女儿 Maria。对于命名，Monty 似乎很有章法。Monty 生于 1962 年，到 2013 年已有 51 岁，应该不太可能推出第四个 DBMS 了。

1.2 MariaDB 与 MySQL

下面从计算机工程师（而非哲学）角度讲解 MariaDB。如前所述，MariaDB 始于 MySQL，通过比较二者异同，各位将进一步了解 MariaDB。

注意

本书所用的“MySQL”一词有时仅指 Oracle 的 MySQL，有时也指 MariaDB、PerconaServer（Percona 出品）、Oracle MySQL 的统称。因为对于篇幅及说明的简洁性而言，每次都使用“Oracle MySQL”或“MySQL 分支版本（Fork）”将会使表述变得冗长。这一规则也同样适用于 InnoDB 与 XtraDB 存储引擎。“InnoDB”有时仅指 Oracle MySQL 开发的 InnoDB 存储引擎，有时还包括 XtraDB。

我在叙述相关内容时会尽量使用合适的术语将之区分，但仍有一些内容无法通过用词的改变进行划分。各位阅读这些内容时，请根据上下文进行判断。

1.2.1 MariaDB、MySQL 和 PerconaServer

了解 MariaDB 之前，先要学习 MySQL 与 PerconaServer。相信各位已经掌握了大量 MySQL 服务器的相关知识。PerconaServer 是 Percona 公司（<http://www.percona.com/>）以 MySQL 服务器源代码为基础创建的 MySQL 服务器的另一个分支版本。

与 Oracle、Monty Program AB 不同，Percona 是一家以提供 MySQL 服务器咨询服务为主的公司。Percona 公司利用提供技术支持时积累的经验，对 MySQL 服务器存在的不足逐一进行修改调整，并在此基础上推出了 PerconaServer，大大提升了 MySQL 服务器的整体性能。

PerconaServer 更注重数据库运营维护的便利与整体性能的提升。与 MariaDB 不同，每当 MySQL 发布新版本时，Percona 会将自己扩展的功能插入 MySQL 服务器源代码，形成新的 PerconaServer 版本并发布。也就是说，Percona 推出 PerconaServer 的周期以及版本号与 MySQL 几乎完全一致。

MySQL、MariaDB、PerconaServer 三者之间的关系如图 1-1 所示。

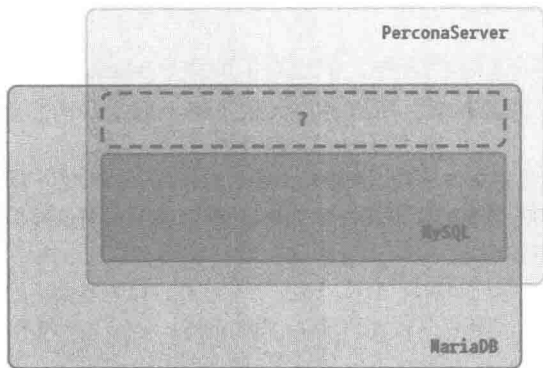


图 1-1 MariaDB、PerconaServer 之间的关系

从图 1-1 可以看到，MariaDB 与 PerconaServer 都在 MySQL 服务器代码基础上推出新版本。现在，虽然 MariaDB 与 MySQL 源代码正逐渐向不同方向发展，但每当 MySQL 推出新版本时，PerconaServer 几乎同时就会推出对应的新版本并添加新功能，对性能进行改善和提升。与 PerconaServer 类似，MariaDB 也会在 MySQL 基础上添加一些扩展功能再推出。有意思的是，PerconaServer 不会使用 MariaDB 中实现的新功能，但是 MariaDB 会把 Percona 中实现的功能（图 1-1 中的“？”部分）一起添加到新的发布版本，其中最具代表性的就是 Percona 开发的 XtraDB 存储引擎。

图 1-2 描述了 MariaDB、MySQL、PerconaServer 三者之间的版本关系。图中实线表示“基于”的关系，箭头所指的数据库是在箭尾数据库的基础上创建而成的；图中虚线表示“部分基于”的关系，即采用部分功能的源代码，并且这些代码一般都要经过修改完善。图中从 PerconaServer 的 XtraDB 到 MariaDB 的虚线表示 MariaDB 直接使用 XtraDB 存储引擎。

InnoDB 是 Oracle MySQL 服务器默认包含的存储引擎，XtraDB 是在 InnoDB 的源码基础上改良而成的新存储引擎。XtraDB 与 InnoDB 的所有数据文件保持 100% 兼容。因此，实际应用中完全可以使用 XtraDB 取代 InnoDB 存储引擎。PerconaServer 与 MariaDB 5.5 中，XtraDB 取代了 InnoDB，成为默认的存储引擎。到 MariaDB 10.0.6^① 版本为止，尚未包含 Percona 的 XtraDB 存储引擎，但是当 Percona 的 XtraDB 存储引擎成为 MariaDB 10.0 必需功能后（MariaDB 10.0.7），它即作为默认存储引擎随 MariaDB 一起发布。因此，要先掌握并记住现在所用的

① 本书所讲内容均以 MariaDB 10.0.6 为基础，部分功能在 MariaDB 10.0.7（开始与 XtraDB 一起发布）中测试。