



贾代平 吴丽娟 著

ORACLE

数据存储与访问技术



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

ORACLE 数据存储与访问技术

贾代平 吴丽娟 著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本专门介绍 ORACLE 系统数据存储与访问技术的学术专著，内容涉及数据存储效率与访问优化的一系列核心问题，也是深入 ORACLE 系统数据管理、性能调优、应用开发的技术基石。理解和掌握本书中阐述的系列内容及其技术细节，是读者进入精彩的 ORACLE 世界的必由之路。

作者围绕 ORACLE 系统的内部数据存储及其访问机制展开介绍，叙述角度由宏观到微观、由整体到局部、由上层到底层，多层次向读者展示海量数据存储与访问背后的各种支持技术及其应用方法，内容包括逻辑与物理存储结构、事务处理与版本控制、大表分区与索引维护、多表联接与访问、性能度量、封锁与争用、等待事件、并行处理等影响数据存储与访问效率的多个方面。本书叙述简洁，内容精炼，具有很强的针对性，特别适合 IT 领域的系统管理人员、大数据系统的研究人员阅读，同时对从事数据管理与数据处理技术的相关教师、研究生也具有很高的参考价值。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

ORACLE 数据存储与访问技术 / 贾代平, 吴丽娟著. ——北京: 电子工业出版社, 2013.9
ISBN 978-7-121-20732-7

I. ① O… II. ① 贾… ② 吴… III. ① 关系数据库系统 IV. ① TP311.138

中国版本图书馆 CIP 数据核字 (2013) 第 133071 号

责任编辑: 董亚峰

文字编辑: 薄 宇

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 720×1 000 1/16 印张: 21.75 字数: 414 千字

印 次: 2013 年 9 月第 1 次印刷

定 价: 48.00 元



凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

前 言

计算机学科是一个应用性极强的领域，其计算理论只有与软件技术相结合才能在实践中提高软件的生产力和服务效率。目前无论在学术界还是产业界，关于大数据的研究与应用方兴未艾。这里的“大”不仅指数据量大、存储量大，同时也指计算量大、访问量大。面对这些“大”的问题，在工程应用中往往伴随着存储和访问的“效率”和“性能”问题。基于现有的技术背景和应用环境，如何有效地解决在海量数据条件下的数据存储和访问的效率、性能的瓶颈问题是一项值得研究的工程应用课题。

现代数据系统的两项核心功能就是为用户提供业务数据的存储和数据访问服务。作为数据存储与服务系统的开发商，他们总是倾向于为用户提供标准的访问接口，如 ANSI/ISO SQL、RPC、OLEDB、JDBC 等，借助于这些接口，用户可以方便地访问各类数据中心及其服务器。但随着数据量的增加及海量数据处理的需要，访问的效率成为越来越多的应用系统需要关注的问题。如何有效地提高用户的访问效率？仅靠通用的访问接口是不够的，我们需要对现代数据库系统的内部机制有深入的研究，只有这样，才能在系统服务的基础上更好地利用服务器本身的存储资源、计算资源等。

作者长期从事海量数据系统的开发、管理与维护研究，近年来重点关注海量数据系统的性能调整与优化。本书是作者在大量工程实践基础上将数据存储与访问效率方面的共性问题做出的阶段性总结。为了使本书的内容具有更好的针对性，我们选择在国内外得到广泛应用的 ORACLE 系统作为讨论对象，以提高应用系统的数据访问效率为主线，重点讨论影响用户业务数据存储和访问效率的各个主要方面，内容包括 ORACLE 数据库的物理和逻辑存储结构、事务处理过程与数据的多版本控制、大表分区与索引维护、多表联接与访问、性能问题的度量、封锁与争用、等待事件、并行处理等。通过对这些问题的讨论，目的是向读者展示提高 ORACLE 系统存储和访问效率的各种技术途径。

为了实现上述目标，我们对全书内容的组织做了精心的规划与设计。虽然目前

业界实际部署的 ORACLE 系统存在多个不同版本，但面对海量数据存储和访问，这些不同版本、不同平台下的 ORACLE 系统在性能和效率方面遭遇的问题是大致相同的、相通的。本书将这些共性问题循序渐进地组织为十八章，内容从存储的逻辑架构到底层的物理结构，从访问的技术路径到访问的并发处理等。具体内容分述如下：

第一章和第二章从整体上阐述了 ORACLE 数据库系统的逻辑存储结构及其物理存储形态，包括表空间、闪回恢复区、RAID、ASM 等重要内容。这两章是全书讨论的技术基础。

第三章探讨了 ORACLE 数据库系统的基本存储单元——数据块，这是 ORACLE 最重要的存储结构，对它的分解和认识构成了一切数据存储的微观世界。

第四章在第三章的基础上讨论了事务处理过程在数据块上的实现机制，包括对多版本数据读一致性的理解，几乎所有的数据访问都涉及到数据块级别的微观处理。

第五章和第六章的内容是第三章和第四章的延续，通过对数据块讹误的处理、数据块的导出（dump）内容的分析进一步认识 ORACLE 系统的微观存储单元。

第七章讨论了 ORACLE 数据库内部的一个重要物理量 SCN，数据存储、数据访问、数据恢复、事务处理都要依赖于该物理量。

第八章专门讨论了 ORACLE 系统各种伪列的意义及其应用。善于利用伪列是认识数据存储和提高访问效率的必备技能。

第九章专门探讨与表有关的存储结构与访问机制，包括存储参数、高水位线、行链接和行迁移的检测与消除、表分析与统计等内容，这些都是影响表访问效率的因素。

第十章讨论 ORACLE 的分区技术，包括表分区和索引分区。针对应用需求灵活运用各种分区技术是实现高效海量数据处理的重要手段。

第十一章探讨各种多表联接方法及其应用场景。确定多表联接的方法和分析多表联接的有效性是提高 SQL 语句访问效率的手段之一。

第十二章集中讨论了关于索引的话题，包括索引类别、索引参数、索引的分析与重构、索引的访问方式、索引维护、索引的价值等多个方面。高效的数据访问离不开索引的支持。

第十三章和第十四章分别描述了在有限服务器资源的情况下，数据访问性能的一般性瓶颈，以及性能问题的主要度量手段。

第十五章专门讨论了关于锁的问题，包括锁的类别、锁的争用、死锁及其解决方法等与并发访问有关的共性问题。

第十六章和第十七章集中探讨了关于等待事件及其接口问题,这是从服务器内部资源冲突的角度讨论数据访问的性能问题,其中包括介绍用户响应的时间模型、利用等待事件发现性能瓶颈等内容。

第十八章专题讨论了数据库服务器内部的并行处理技术。在多处理器、分布数据存储的条件下,适当利用并行处理机制,是有效提高数据访问效率的重要手段。

技术的价值在于应用。本书着重从应用的角度去探讨数据存储与访问方面的问题,上述十八章的内容分别从不同的侧面探索了提高 ORACLE 系统数据存储和访问效率的可能途径。在此需要说明的是,在解决工程实践中的性能和效率问题时,往往需要结合具体的应用背景灵活运用、综合实施不同的技术手段辩证地分析问题和解决问题。希望本书的出版对此类技术的推广与应用有所裨益,在此感谢电子工业出版社的大力协助。另外,研读本书需要有较好的 ORACLE 技术基础。相信读者在阅读完本书的全部内容后,对 ORACLE 系统的认识一定会得到质的升华。

本书由贾代平同志主笔,吴丽娟同志编写了其中的第七章和第八章,最后由贾代平同志统稿。限于作者的水平和个人认识上的局限性,书中存在缺陷、不足也在所难免,欢迎读者对本书的疏漏之处给予批评指正,并提供任何意见和建议,作者在此表示衷心的感谢。同时作者愿意通过邮件与读者交流与本书相关的各类技术问题,并为读者提供必要的技术支持。联系邮箱: jiadp@163169.net。

贾代平
2013年5月

目 录

第 1 章 ORACLE 的物理存储及其管理	1
1.1 tablespace 与存储空间	1
1.2 系统表空间与数据字典	2
1.2.1 system 和 sysaux 表空间	2
1.2.2 数据字典 (data dictionary)	2
1.3 临时信息与临时表空间	4
1.4 回滚信息与回滚表空间	5
1.4.1 回滚段的分类	6
1.4.2 创建回滚段	6
1.4.3 管理回滚段	7
1.4.4 undo 空间及其自动管理	7
1.5 数据库表空间的存储结构	8
1.5.1 表空间的存储关系	8
1.5.2 表空间的存储参数设置	10
1.5.3 表空间的维护	13
1.6 数据库文件管理	15
1.6.1 控制文件的管理	15
1.6.2 数据文件的管理	18
1.6.3 联机日志的配置与管理	26
1.7 OMF 方式管理文件	32
1.8 闪回恢复区 fra	33
1.8.1 闪回恢复区的作用	34
1.8.2 配置闪回恢复区	34
1.8.3 闪回恢复区的文件保留策略	35
1.8.4 使用闪回恢复区	36

1.9	RAID 磁盘阵列存储	37
1.9.1	RAID 存储规范	37
1.9.2	ORACLE 的 RAID 存储方案建议	38
第 2 章	自动存储管理与 ASM 数据库	40
2.1	自动存储的主要概念	40
2.2	ASM 的系统结构	41
2.3	准备 ASM 存储设备	43
2.3.1	LVM 与 ASM	43
2.3.2	创建 raw device	44
2.3.3	标识 ASM 磁盘	45
2.4	ORACLE 集群同步服务 CSS	45
2.5	创建并启动 ASM 实例	46
2.6	在 ASM 实例中创建磁盘组	47
2.6.1	磁盘组的冗余级别	47
2.6.2	创建 ASM 磁盘组	48
2.6.3	在 ASM 实例中加载磁盘组	49
2.7	ASM 实例与 RDBMS 实例的关系	51
2.8	直接创建 ASM 数据库	52
2.9	将现有数据库迁移至 ASM	53
2.9.1	启用基于磁盘组的 OMF	54
2.9.2	迁移控制文件至 ASM 磁盘组	55
2.9.3	迁移数据文件至 ASM 磁盘组	56
2.9.4	切换数据库至 ASM 磁盘组上运行	57
2.9.5	迁移临时表空间	58
2.9.6	迁移联机日志文件	59
2.9.7	关于闪回日志和块跟踪文件	60
2.10	ASM 信息与 ASM 文件管理	61
2.10.1	动态性能视图	61
2.10.2	ASM 磁盘文件管理	61
2.10.3	ASM 命令行工具	63
第 3 章	数据块结构	64
3.1	data block 描述	64
3.2	data block 结构	65

3.3	行链接与行迁移	69
第 4 章	ITL 与事务处理	71
4.1	ITL 描述	71
4.2	ITL 等待	72
4.3	ITL 实验案例	73
4.4	ITL 进一步研究	75
4.5	ITL 与 CR 块	79
第 5 章	数据块损坏的修复	82
5.1	数据块损坏描述	82
5.2	数据块损坏的检测	83
5.2.1	初始化参数设置	83
5.2.2	RMAN 检查	84
5.2.3	dbverify 工具	85
5.2.4	其他工具检查	86
5.3	数据块损坏的模拟和修复	86
5.3.1	段与存储数据块	86
5.3.2	模拟数据块损坏	89
5.3.3	基于块的数据恢复	92
第 6 章	dump 数据块	95
6.1	block dump 描述	95
6.2	dump 数据块	96
6.3	dump 索引块	102
第 7 章	系统改变号 SCN	106
7.1	SCN 描述	106
7.2	SCN 与数据恢复	108
7.3	SCN 数据恢复案例	109
第 8 章	伪列 pseudo column	112
8.1	伪列描述	112
8.2	常用伪列的研究	112
8.2.1	伪列 rowid	113

8.2.2	伪列 rownum	114
8.3	伪列 ora_rowscn 研究	116
8.4	伪列 level 与层次查询	121
第 9 章	表存储结构与访问优化	126
9.1	存储结构中的块	126
9.1.1	块结构及其控制参数	127
9.1.2	行迁移与行链接	128
9.2	正确设置参数 pctfree	128
9.3	避免行迁移与行链接	131
9.3.1	行迁移与行链接的检测	131
9.3.2	行迁移与行链接的消除方法	132
9.4	消除迁移行和链接行实例	133
9.5	高水位线 HWM	135
9.6	表存储统计实验	136
9.6.1	验证表结构	136
9.6.2	发现行迁移	137
9.6.3	收集统计信息	137
9.6.4	表分析实验	137
第 10 章	分区 Partition 技术	145
10.1	分区 Partition 概述	145
10.2	表分区的基本类别	146
10.2.1	范围分区 (by range)	146
10.2.2	列表分区 (by list)	148
10.2.3	散列分区 (by hash)	148
10.2.4	间隔分区 (by interval)	149
10.3	基本分区的复合	150
10.4	分区索引技术	152
10.4.1	索引分区与表	152
10.4.2	本地分区索引	153
10.4.3	全局分区索引	155
10.5	分区表与索引的维护	157
10.6	分区交换及其应用	164
10.7	联机分区处理	167

第 11 章 数据访问之多表联接	173
11.1 多表联接概述	173
11.2 联接条件和类型	174
11.2.1 笛卡尔乘积	174
11.2.2 多表的条件联接	175
11.2.3 避免使用子查询	177
11.3 多表联接的方法	178
11.3.1 嵌套循环	178
11.3.2 融合联接	182
11.3.3 哈希联接	184
11.3.4 三种联接方式比较	185
第 12 章 数据访问之 ORACLE 索引	187
12.1 ORACLE 索引概述	187
12.2 b 树索引和 bitmap 索引	188
12.2.1 b 树索引结构	188
12.2.2 bitmap 索引结构	189
12.2.3 对 null 值的索引	190
12.3 聚簇因子	192
12.3.1 计算聚簇因子	192
12.3.2 对访问性能的影响	194
12.4 索引分析与重建	199
12.4.1 索引分析与统计	199
12.4.2 查询索引状态及使用信息	200
12.4.3 索引重建	204
12.5 与索引有关的参数	205
12.6 访问索引的方式	206
12.6.1 索引扫描方式	206
12.6.2 两类数据块扫描	210
12.7 b 树索引的维护机制	211
12.7.1 insert 操作的 b 树维护	211
12.7.2 delete 操作的 b 树维护	221
12.7.3 update 操作的 b 树维护	227
12.8 复合索引的使用	228

12.8.1	两个使用原则	228
12.8.2	复合索引和 order by	232
12.9	关于索引使用的建议	232
第 13 章	数据库访问性能的几个基本问题	234
13.1	内存问题	234
13.2	CPU 利用率	235
13.3	I/O 问题	235
13.4	高资源消耗的 SQL	236
13.5	引发性能瓶颈的应用问题	238
13.6	OLTP 与 OLAP	240
第 14 章	性能问题的度量手段	242
14.1	性能调优的度量概述	242
14.2	explain 解释 SQL	242
14.2.1	配置 explain	243
14.2.2	获得执行计划	243
14.3	语句级跟踪 autotrace	245
14.4	会话级跟踪 sql_trace	247
14.4.1	设置 SQL 跟踪	247
14.4.2	tkprof 格式化跟踪文件	249
14.5	扩展的 SQL 跟踪	251
第 15 章	ORACLE 锁机制与锁争用问题	254
15.1	ORACLE 数据库的锁类型	254
15.2	数据访问过程中的加锁	256
15.3	与锁有关的字典参数与指令	262
15.3.1	有关锁的数据字典视图	262
15.3.2	有关锁的初始化参数	263
15.3.3	事务的隔离级别	264
15.4	锁争用与死锁	265
15.4.1	量测锁争用	266
15.4.2	处理死锁	267
15.5	锁存器 latch	271
15.5.1	锁存器机制	271

15.5.2	检查 latch 争用	271
第 16 章	wait 与 time 响应问题	274
16.1	基于 wait 的性能问题描述	274
16.2	用户响应的时间模型	276
16.2.1	CPU 服务时间	276
16.2.2	等待事件及等待时间	277
16.3	统计项与等待事件	278
16.4	db time 和 db CPU	280
16.5	top SQL 说明	281
16.6	等待事件直方图	282
第 17 章	等待事件接口与性能调整	284
17.1	性能与等待事件	284
17.2	常见的等待事件及其描述	285
17.3	等待事件不能反映的信息	286
17.4	收集等待事件信息	287
17.5	利用等待事件发现性能瓶颈	289
17.5.1	一个慢速查询的处理	289
17.5.2	耗时的调度批处理	292
17.5.3	客户服务器应用中的等待事件	295
17.5.4	疲于应付的数据库服务器	297
17.6	等待事件接口小结	299
第 18 章	并行处理与访问技术	300
18.1	并行处理技术概述	300
18.2	SQL 语句的并行处理	300
18.2.1	串行处理与并行处理	301
18.2.2	并行处理的主要概念	302
18.3	并行处理的性能提升	303
18.4	并行处理的适应性	304
18.4.1	多 CPU 主机系统	304
18.4.2	分布式存储	305
18.4.3	资源密集型 SQL	305
18.4.4	批量数据扫描	306

18.5	控制并行处理	306
18.5.1	确定并行度	306
18.5.2	使用并行提示 hint	308
18.5.3	调整与并行处理有关的参数	309
18.6	并行处理的执行计划	310
18.7	实时的并行处理信息	312
18.8	并行处理的跟踪	313
18.9	并行处理实例	314
18.9.1	并行数据更新	314
18.9.2	并行数据添加	315
18.9.3	DDL 的并行处理	318
18.9.4	并行索引访问	319
18.10	并行处理的优化	321
18.10.1	并行处理的一般性原则	321
18.10.2	部分并行化与完全并行化	322
18.10.3	监控实际运行中的并行度	325
18.10.4	并行处理进程的负荷分配	326
18.10.5	RAC 环境下的并行处理	329
	参考文献	330
	后记	332

第 1 章

ORACLE 的物理存储及其管理

ORACLE 数据库虽然是一个庞大而复杂的系统，但数据库中的所有数据（系统数据、用户数据）都物理地落实到磁盘存储上。逻辑上，数据库空间被划分为若干个表空间（tablespace）；物理上，数据库的存储空间都落实到磁盘的数据文件上。

1.1 tablespace 与存储空间

表空间是数据库的逻辑划分。每个数据库可以含有一个或多个表空间（至少得含有一个系统表空间 system，从 ORACLE 10g 开始增加了一个 sysaux 表空间），并且每个表空间只能属于一个数据库。数据库中的所有数据逻辑上都存储在数据库的表空间中。表空间在物理上由一个或多个数据文件构成。

由于表空间起逻辑划分和组织模式对象及其数据的作用，因此在实际的应用中，典型的数据库都拥有多个表空间。多表空间的数据库在管理和性能方面都优于单表空间数据库。多表空间数据库有以下优点：①多个表空间可以把系统数据（数据字典中的数据）和用户数据分离开；②通过多个表空间可以将一个应用系统的数据与其他应用系统的数据分离；③把不同表空间的数据文件分别放在不同的磁盘上以减少 I/O 竞争；④多个表空间可以实现对不同类型的存储段（segment）的分离，将不同类型的数据存储到不同的表空间，以提高效率；⑤数据库运行期间可以根据需要将某些表空间暂时置于脱机状态、备份状态等，方便存储空间的管理；⑥可以为特定类型数据库应用保留表空间，例如只读活动、临时段存储、

回滚段存储等。

ORACLE 数据库在初始创建时只有少数的几个表空间（取决于 ORACLE 版本、`create database` 语句等），为了实现上述多表空间的优点，需要数据库管理员根据应用需要创建后续的表空间：为数据库分配更多的磁盘存储空间，以扩大数据库的容量；为了使某类数据与其他数据分开，需要为其创建一个单独的表空间。

表空间按用途不同可以分为这样几大类，分别是：系统表空间、临时表空间、回滚表空间和用户表空间，其中前三类都是用于 ORACLE 数据库自身内部使用，用户的业务数据应该存储于一个或多个用户表空间上。下面分别讲述这几类表空间的创建与使用。

1.2 系统表空间与数据字典

1.2.1 system 和 sysaux 表空间

ORACLE 数据库需要的第一个表空间就是 `system` 表空间，它是在创建数据库时自动创建的，即，一个 ORACLE 数据库至少要具有一个 `system` 表空间，且该表空间在数据库打开期间必须处于联机状态。

系统表空间主要用来存储数据字典（`data dictionary`）信息。虽然 ORACLE 并不禁止用户向系统表空间存放其他的数据库对象，但为了数据库的有效运行，强烈建议不要这样做。数据字典中的所有对象都属于 `sys` 用户。默认情况下，`sys` 用户和 `system` 用户下的所有对象都存放在系统表空间内，换句话说，系统表空间尽量留给 `sys` 用户和 `system` 用户使用。在包括 ORACLE 较早的版本中，系统表空间还存储有 ORACLE 管理工具所使用的数据库对象，在 ORACLE 10g 中，这些辅助性的数据库对象已经移植到另外一个系统表空间 `sysaux` 中，也就是说，ORACLE 10g 中有两个系统表空间，它们都是 ORACLE 自身所使用的数据库空间。

1.2.2 数据字典（`data dictionary`）

数据字典是 ORACLE 数据库的“信息中心”，它是由一系列 `sys` 用户所拥有的数据库对象（包括一系列基础表和动态性能视图等）构成，保存了关于数据库自身，以及其他所有数据库对象的信息（这类信息有时又称为 `metadata`），在 ORACLE 数据库运行期间，需要随时访问这类信息，并且会自动维护它们，不需要 DBA 去人为地更改它们。

具体地，数据字典中保存有如下主要信息：

- (1) 数据库的结构信息，包括内存结构、物理结构、进程结构、会话信息等；
- (2) 各种数据库对象（表、索引、视图、同义词、序列、存储过程、触发器、程序包等）的定义信息，它们由 DDL 语句生成；
- (3) 数据库存储空间的配置和使用信息，包括表空间及其内部使用情况；
- (4) 数据库安全及其权限控制信息，如用户、角色、权限分配等信息；
- (5) 数据库运行期间的各类统计信息，这些信息可以让 DBA 随时监测数据库性能；
- (6) 其他关于数据库本身的各种必要信息，这些一般与数据库版本有关。

数据字典中的库表信息主要分为三类：基础表、静态视图和动态性能视图。基础表中的信息一旦形成，在运行期间一般不会改变（除非 DDL 语句的作用），静态视图是在基础表上形成的一系列视图，是数据字典的对外接口。动态性能视图是依赖数据库运行时的状态，反映数据库运行的一些内在信息，所以在访问这类数据字典时往往不是一成不变的。以下分别就这两类数据字典来论述。

数据字典主要是由表和视图组成，ORACLE 建议通过视图访问数据字典。数据字典中的静态视图主要分为三类，它们分别由三个前缀构成：`user_*`、`all_*`和 `dba_*`。

(1) `user_*`：该视图存储了关于当前用户所拥有的对象的信息（即所有在该用户模式下的对象）。

(2) `all_*`：该视图存储了当前用户能够访问的对象的信息（与 `user_*`相比，`all_*`并不需要拥有该对象，只需要具有访问该对象的权限即可）。

(3) `dba_*`：该视图存储了数据库中所有对象的信息（前提是当前用户具有访问这些数据库的权限，一般来说必须具有管理员 DBA 权限）。

从上面的描述可以看出，三者存储的数据肯定会有重叠，其实它们除了访问范围不同以外（因为权限不同，所以访问对象的范围不同），其他均具有一致性。具体来说，由于数据字典视图是由 `sys`（系统用户）所拥有的，所以在默认情况下，只有 `sys` 和拥有 DBA 系统权限的用户可以看到所有的视图。没有 DBA 权限的用户只能看到 `user_*`和 `all_*`视图。如果没有被授予相关的 `select` 权限，他们是不能看到 `dba_*`视图的。

动态性能视图：ORACLE 包含了一些潜在的由系统管理员如 `sys` 维护的表和视图，由于当数据库运行时它们会不断进行更新，所以称它们为动态数据字典（或者是动态性能视图）。这些视图提供了关于内存和磁盘的运行情况，所以我们只能对其进行只读访问而不能修改它们。ORACLE 中这些动态性能视图都是以 `v$`开头的视图，比如 `v$session`、`v$sgainfo` 等。