对 Oracle 、SQL Server、MySQL、PostgreSQL 数据库优缺点分析

Oracle Database

Oracle Database，又名 Oracle RDBMS，或简称 Oracle。是甲骨文公司的一款关系数据库管理系统。它是在数据库领域一直处于领先地位的产品。可以说 Oracle 数据库系统是目前世界上流行的关系数据库管理系统，系统可移植性好、使用方便、功能强，适用于各类大、中、小、微机环境。它是一种高效率、可靠性好的 适应高吞吐量的数据库解决方案。

优点

1. Oracle 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选择适合解决方案对开发商全力支持。

2. Oracle 并行服务器通过使组结点共享同簇工作来扩展 windownt 能力提供高用性和高伸缩性簇解决方案 windowsNT 能满足需要用户把数据库移 UNIXOracle 并行服务器对各种 UNIX 平台集群机制都有着相当高集成度。

3. 获得最高认证级别的 ISO 标准认证。

4. Oracle 性能高 保持开放平台下 TPC-D 和 TPC-C 世界记录。

5. Oracle 多层次网络计算支持多种工业标准用 ODBC、JDBC、OCI 等网络客户连接。

6. Oracle 长时间开发经验完全向下兼容得广泛应用地风险低。

缺点

1. 对硬件的要求很高。

2. 价格比较昂贵。

3. 管理维护麻烦一些。

4. 操作比较复杂，需要技术含量较高。

SQL Server

SQL Server 是 Microsoft 公司推出的关系型数据库管理系统。具有使用方便可伸缩性好与相关软件集成程度高等优点，可跨越从运行 Microsoft Windows 98 的膝上型电脑到运行 Microsoft Windows 2012 的大型多处理器的服务器等多种平台使用。

Microsoft SQL Server 是一个全面的数据库平台，使用集成的商业智能 (BI)工具提供了企业级的数据管理。Microsoft SQL Server 数据库引擎为关系型数据和结构化数据提供了更安全可靠的存储功能，使您可以构建和管理用于业务的高可用和高性能的数据应用程序。

优点

1. 易用性、适合分布式组织的可伸缩性、用于决策支持的数据仓库功能、与许多其他服务器软件紧密关联的集成性、良好的性价比等；

2. 为数据管理与分析带来了灵活性，允许单位在快速变化的环境中从容响应，从而获得竞争优势。从数据管理和分析角度看，将原始数据转化为商业智能和充分利用 Web 带来的机会非常重要。作为一个完备的数据库和数据分析包，SQLServer 为快速开发新一代企业级商业应用程序、为企业赢得核心竞争优势打开了胜利之门。 作为重要的基准测试可伸缩性和速度奖的记录保持者，SQLServer 是一个具备完全 Web 支持的数据库产品，提供了对可扩展标记语言 (XML) 的核心支持以及在 Internet 上和防火墙外进行查询的能力；

缺点

1. SQL Server 只能 windows 上运行，没有丝毫开放性操作系统，系统稳定对数据库十分重要，WindowsX 系列产品偏重于桌面应用，NT server 只适合小型企业，而且 windows 平台可靠性、安全性和伸缩性非常有限。
2. SQL server 并行实施和共存模型并成熟难处理日益增多用户数和数据卷伸缩性有限；
3. 没有获得任何安全证书。
4. SQL Server 多用户时性能佳。

MySQL

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下公司。MySQL 在 WEB 应用方面 MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL 是一种关联数据库管理系统， 关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。其社区版的性能卓越，搭配 PHP 和 Apache 可组成良好的开发环境。

优点

1. 体积小、速度快、总体拥有成本低，开源，提供的接口支持多种语言连接操作。
2. 支持多种操作系统。
3. MySQL 的核心程序采用完全的多线程编程。线程是轻量级的进程，它可以灵活地为用户提供服务，而不过多的系统资源。用多线程和 C 语言实现的 MySQL 能很容易充分利用 CPU。
4. MySQL 有一个非常灵活而且安全的权限和口令系统。当客户与 MySQL 服务器连接时，他们之间所有的口令传送被加密，而且 MySQL 支持主机认证。
5. MySQL 能够提供很多不同的使用者界面，包括命令行客户端操作，网页浏览器，以及各式各样的程序语言界面，例如 C++，Perl，Java，PHP，以及 Python。你可以使用事先包装好的客户端，或者干脆自己写一个合适的应用程序。MySQL 可用于 Unix，Windows，以及 OS/2 等平台，因此它可以用在个人电脑或者是服务器上。

缺点

1. 不支持热备份。
2. MySQL 不支持自定义数据类型
3. MySQL 最大的缺点是其安全系统，主要是复杂而非标准，另外只有到调用 mysqladmin 来重读用户权限时才发生改变。
4. MySQL 对存储过程和触发器支持不够良好。
5. 尽管 MySQL 理论上仍是开源产品，也有人抱怨它诞生之后更新缓慢。然而，应该注意到有一些基于 MySQL 并完整集成的数据库（如 MariaDB），在标准的 MySQL 基础上带来了额外价值。
6. MySQL 对 XML 支持不够良好

何时使用 ？

1. 分布式操作：

当你需要的比 SQLite 可以提供的更多时，把 MySQL 包括进你的部署栈，就像任何一个独立的数据库服务器，会带来大量的操作自由和一些先进的功能。

2. 高安全性：

MySQL 的安全功能，用一种简单的方式为数据访问（和使用）提供了可靠的保护。

3. Web 网站 和 Web 应用：

绝大多数的网站（和 Web 应用程序）可以忽视约束性地简单工作在 MySQL 上。这种灵活的和可扩展的工具是易于使用和易于管理的——这被证明非常有助于长期运行。

4. 定制解决方案：

如果你工作在一个高度量身定制的解决方案上，MySQL 能够很容易地尾随和执行你的规则，这要感谢其丰富的配置设置和操作模式。

何时不用 ？

1. SQL 服从性：

因为 MySQL 没有[想要]实现 SQL 的全部标准，所以这个工具不完全符合 SQL。如果你需要对这样的关系数据库管理系统进行整合，从 MySQL 进行切换是不容易的。

2. 并发：

即使 MySQL 和一些存储引擎能够真地很好执行读取操作，但并发读写还是有问题的。

3. 缺乏特色：

再次提及，根据数据库引擎的选择标准，MySQL 会缺乏一定的特性，如全文搜索。

PostgreSQL

PostgreSQL 是一个自由的对象-关系数据库服务器(数据库管理系统),支持大部分 SQL 标准并且提供了许多其他现代特性：复杂查询、外键、触发器、视图、事务完整性、MVCC。同样，PostgreSQL 可以用许多方法扩展，比如，通过增加新的数据类型、函数、操作符、聚集函数、索引。免费使用、修改、和分发 PostgreSQL。

优点

1. PostgreSQL 是一个开源的，免费的，同时非常强大的关系型数据管理系统。

2. PostgreSQL 背后有热忱而经验丰富的社区，可以通过知识库和问答网站获取支持，全天候免费。

3. 即使其本身功能十分强大，PostgreSQL 仍附带有许多强大的开源第三方工具来辅助系统的设计、管理和使用。

4. 可以用预先存储的流程来程序性扩展 PostgreSQL ，一个高级的关系型数据库理应如此。

5. PostgreSQL 不只是一个关系型数据库，还是一个面向对象数据库——支持嵌套，及一些其他功能。

缺点

1. 对于简单而繁重的读取操作, 超过了 PostgreSQL 的杀伤力，可能会出现比同行（如 MySQL）更低的性能。

2. 按给出的该工具的性质，从普及度来说它还缺乏足够后台支撑，尽管有大量的部署——这可能会影响能够获得支持的容易程度。

何时使用？

1. 数据完整性：

当可靠性和数据完整性是绝对必要而无需理由时，PostgreSQL 是更好的选择。

2. 复杂的自定义过程：

如果你需要你的数据库执行自定义过程，可扩展的 PostgreSQL 是更好的选择。

3. 整合:

在将来，如果可能要把整个数据库系统迁移到另一个适当的解决方案（例如 Oracle）中，PostgreSQL 对于这种切换将是最兼容和易于操作的。

4. 复杂的设计:

相比其他的开源和免费的 RDBMS（关系数据库管理系统）实现来说，对于复杂的数据库设计，PostgreSQL 提供了大部分的功能和可能性，同时并没放弃其他有价值的地方。

何时不用？

1. 速度:

如果你需要的只是快速的读取操作, PostgreSQL 不是为此而准备的工具。

2. 简化体制:

除非你需要绝对的数据完整性，原子性，一致性，隔离性，耐久性，或复杂的设计，PostgreSQL 对简化体制来说是杀手。

3. 复制:

除非你愿意花不少时间，精力和资源，否则对于那些缺乏数据库和系统管理经验的人来说，实现与 MySQL 的（主从）复制可能不容易。

摘：

为了说明 PostgreSQL 的功能，我下面简要对比一下 PostgreSQL 数据库与 MySQL 数据库之间的差异：

我们先借助 Jametong 翻译的"从 Oracle 迁移到 Mysql 之前必须知道的 50 件事"，看一看如何把 Oracle 转到 MySQL 中的困难：

50 things to know before migrating Oracle to MySQL

by Baron Schwartz,Translated by Jametong

1. 对子查询的优化表现不佳.

2. 对复杂查询的处理较弱

3. 查询优化器不够成熟

4. 性能优化工具与度量信息不足

5. 审计功能相对较弱

6. 安全功能不成熟,甚至可以说很粗糙.没有用户组与角色的概念,没有回收权限的功能(仅仅可以授予权限).当一个用户从不同的主机/网络以同样地用户名/密码登录之后,可能被当作完全不同的用户来处理.没有类似于 Oracle 的内置的加密功能.

7. 身份验证功能是完全内置的.不支持 LDAP,Active Directory 以及其它类似的外部身份验证功能.

8. Mysql Cluster 可能与你的想象有较大差异.

9. 存储过程与触发器的功能有限.

10. 垂直扩展性较弱.

11. 不支持 MPP(大规模并行处理).

12. 支持 SMP(对称多处理器),但是如果每个处理器超过 4 或 8 个核(core)时,Mysql 的扩展性表现较差.

13. 对于时间、日期、间隔等时间类型没有秒以下级别的存储类型.

14. 可用来编写存储过程、触发器、计划事件以及存储函数的语言功能较弱.

15. 没有基于回滚(roll-back)的恢复功能,只有前滚(roll-forward)的恢复功能.

16. 不支持快照功能.

17. 不支持数据库链(database link).有一种叫做 Federated 的存储引擎可以作为一个中转将查询语句传递到远程服务器的一个表上,不过,它功能很粗糙并且漏洞很多.

18. 数据完整性检查非常薄弱,即使是基本的完整性约束，也往往不能执行。

19. 优化查询语句执行计划的优化器提示非常少.

20. 只有一种表连接类型:嵌套循环连接(nested-loop),不支持排序-合并连接(sort-merge join)与散列连接(hash join).

21. 大部分查询只能使用表上的单一索引;在某些情况下,会存在使用多个索引的查询,但是查询优化器通常会低估其成本,它们常常比表扫描还要慢.

22. 不支持位图索引(bitmap index).每种存储引擎都支持不同类型的索引.大部分存储引擎都支持 B-Tree 索引.

23. 管理工具较少,功能也不够成熟.

24. 没有成熟能够令人满意的 IDE 工具与调试程序.可能不得不在文本编辑器中编写存储过程,并且通过往表(调试日志表)中插入记录的方式来做调试.

25. 每个表都可以使用一种不同的存储引擎.

26. 每个存储引擎在行为表现、特性以及功能上都可能有很大差异.

27. 大部分存储引擎都不支持外键.

28. 默认的存储引擎(MyISAM)不支持事务,并且很容易损坏.

29. 最先进最流行的存储引擎 InnoDB 由 Oracle 拥有.

30. 有些执行计划只支持特定的存储引擎.特定类型的 Count 查询,在这种存储引擎中执行很快,在另外一种存储引擎中可能会很慢.

31. 执行计划并不是全局共享的,,仅仅在连接内部是共享的.

32. 全文搜索功能有限, 只适用于非事务性存储引擎. Ditto 用于地理信息系统/空间类型和查询.

33. 没有资源控制.一个完全未经授权的用户可以毫不费力地耗尽服务器的所有内存并使其崩溃,或者可以耗尽所有 CPU 资源.

34. 没有集成商业智能(business intelligence), OLAP **数据集等软件包.

35. 没有与 Grid Control 类似的工具(http://solutions.mysql.com/go.php?id=1296&t=s)

36. 没有类似于 RAC 的功能.如果你问”如何使用 Mysql 来构造 RAC”,只能说你问错了问题.

37. 不支持用户自定义类型或域(domain).

38. 每个查询支持的连接的数量最大为 61.

39. MySQL 支持的 SQL 语法(ANSI SQL 标准)的很小一部分.不支持递归查询、通用表表达式（Oracle 的 with 语句）或者窗口函数（分析函数）.支持部分类似于 Merge 或者类似特性的 SQL 语法扩展,不过相对于 Oracle 来讲功能非常简单.

40. 不支持功能列(基于计算或者表达式的列,Oracle11g 开始支持计算列,以及早期版本就支持虚列(rownum,rowid)).

41. 不支持函数索引,只能在创建基于具体列的索引.

42. 不支持物化视图.

43. 不同的存储引擎之间,统计信息差别很大,并且所有的存储引擎支持的统计信息都只支持简单的基数(cardinality)与一定范围内的记录数(rows-in-a-range). 换句话说，数据分布统计信息是有限的.更新统计信息的机制也不多.

44. 没有内置的负载均衡与故障切换机制.

45. 复制(Replication)功能是异步的,并且有很大的局限性.例如,它是单线程的(single-threaded),因此一个处理能力更强的 Slave 的恢复速度也很难跟上处理能力相对较慢的 Master.

46. Cluster 并不如想象的那么完美.或许我已经提过这一点,但是这一点值得再说一遍.

47. 数据字典(INFORMATION_SCHEMA)功能很有限,并且访问速度很慢(在繁忙的系统上还很容易发生崩溃).

48. 不支持在线的 Alter Table 操作.

49. 不支持 Sequence.

50. 类似于 ALTER TABLE 或 CREATE TABLE 一类的操作都是非事务性的.它们会提交未提交的事务,并且不能回滚也不能做灾难恢复.Schame 被保存在文件系统上,这一点与它使用的存储引擎无关.

MySQL Vs PostgreSQL

特性	MySQL	PostgreSQL
实例	通过执行 MySQL 命令（mysqld）启动实例。一个实例可以管理一个或多个数据库。一台服务器可以运行多个 mysqld 实例。一个实例管理器可以监视 mysqld 的各个实例。	通过执行 Postmaster 进程（pg_ctl）启动实例。一个实例可以管理一个或多个数据库，这些数据库组成一个集群。集群是磁盘上的一个区域，这个区域在安装时初始化并由一个目 录组成，所有数据都存储在这个目录中。使用 initdb 创建第一个数据库。一台机器上可以启动多个实例。
数据库	数据库是命名的对象集合，是与实例中的其他数据库分离的实体。一个 MySQL 实例中的所有数据库共享同一个系统编目。	数据库是命名的对象集合，每个数据库是与其他数据库分离的实体。每个数据库有自己的系统编目，但是所有数据库共享 pg_databases。
数据	通过 innodb_buffer_pool_size 配置参数设置	Shared_buffers 缓存。在

缓冲区	数据缓冲区。这个参数是内存缓冲区的字节数，InnoDB 使用这个缓冲区来缓存表的数据和索引。在专用的数据库服务器上，这个参数最高可以设置为机器物理内存量的 80%。	默认情况下分配 64 个缓冲区。默认的块大小是 8K。可以通过设置 postgresql.conf 文件中的 shared_buffers 参数来更新缓冲区缓存。
身份验证	MySQL 在数据库级管理身份验证。 基本只支持密码认证。	PostgreSQL 支持丰富的认证方法：信任认证、口令认证、Kerberos 认证、基于 Ident 的认证、LDAP 认证、PAM 认证
加密	可以在表级指定密码来对数据进行加密。还可以使用 AES_ENCRYPT 和 AES_DECRYPT 函数对列数据进行加密和解密。可以通过 SSL 连接实现网络加密。	可以使用 pgcrypto 库中的函数对列进行加密/解密。可以通过 SSL 连接实现网络加密。
审计	可以对 querylog 执行 grep。	可以在表上使用 PL/pgSQL 触发器来进行审计。
查询解释	使用 EXPLAIN 命令查看查询的解释计划。	使用 EXPLAIN 命令查看查询的解释计划。
备份、恢复和日志	InnoDB 使用写前（write-ahead）日志记录。支持在线和离线完全备份以及崩溃和事务恢复。需要第三方软件才能支持热备份。	在数据目录的一个子目录中维护写前日志。支持在线和离线完全备份以及崩溃、时间点和事务恢复。可以支持热备份。
表类型	取决于存储引擎。例如，NDB 存储引擎支持分区表，内存引擎支持内存表。	支持临时表、常规表以及范围和列表类型的分区表。不支持哈希分区表。 由于 PostgreSQL 的表分区是通过表继承和规则系统完成了，所以可以实现更复杂的分区方式。
索引	取决于存储引擎。MyISAM：BTREE，InnoDB：	支持 B-树、哈希、R-树

类型	BTREE。	和 Gist 索引。
约束	支持主键、外键、惟一和非空约束。对检查约束进行解析，但是不强制实施。	支持主键、外键、惟一、非空和检查约束。
存储过程和用户定义函数	支 持 CREATE PROCEDURE 和 CREATE FUNCTION 语句。存储过程可以用 SQL 和 C++ 编写。用户定义函数可以用 SQL、C 和 C++ 编写。	没有单独的存储过程，都是通过函数实现的。用户定义函数可以 用 PL/pgSQL（专用的过程语言）、PL/Tcl、PL/Perl、PL/Python 、SQL 和 C 编写。
触发器	支持行前触发器、行后触发器和语句触发器，触发器语句用过程语言复合语句编写。	支持行前触发器、行后触发器和语句触发器，触发器过程用 C 编写。
系统配置文件	my.conf	Postgresql.conf
数据库配置	my.conf	Postgresql.conf
客户机连接文件	my.conf	pg_hba.conf
XML 支持	有限的 XML 支持。	有限的 XML 支持。
数据访问和管理服务器	OPTIMIZE TABLE —— 回收未使用的空间并消除数据文件的碎片 myisamchk -analyze —— 更新查询优化器所使用的统计数据（MyISAM 存储引擎） mysql —— 命令行工具 MySQL Administrator —— 客户机 GUI 工具	Vacuum —— 回收未使用的空间 Analyze —— 更新查询优化器所使用的统计数据 psql —— 命令行工具 pgAdmin —— 客户 机 GUI 工具

并发控制	支 持表级和行级锁。InnoDB 存储引擎支持 READ_COMMITTED、READ_UNCOMMITTED、REPEATABLE_READ 和 SERIALIZABLE。使用 SET TRANSACTION ISOLATION LEVEL 语句在事务级设置隔离级别。	支 持表级和行级锁。支持的 ANSI 隔离级别是 Read Committed（默认 —— 能看到查询启动时数据库的快照）和 Serialization（与 Repeatable Read 相似 —— 只能看到在事务启动之前提交的结果）。使用 SET TRANSACTION 语句在事务级设置隔离级别。使用 SET SESSION 在会话级进行设置。

MySQL 相对于 PostgreSQL 的劣势：

MySQL	PostgreSQL
最重要的引擎 InnoDB 很早就由 Oracle 公司控制。目前整个 MySQL 数据库都由 Oracle 控制。	BSD 协议，没有被大公司垄断。
对复杂查询的处理较弱，查询优化器不够成熟	很强大的查询优化器，支持很复杂的查询处理。
只有一种表连接类型:嵌套循环连接(nested-loop),不支持排序-合并连接(sort-merge join)与散列连接(hash join)。	都支持
性能优化工具与度量信息不足	提供了一些性能视图，可以方便的看到发生在一个表和索引上的 select、delete、update、insert 统计信息，也可以看到 cache 命中率。网上有一个开源的 pgstatspack 工具。
InnoDB 的表和索引都是按相同的方式存储。也就是说表都是索引组织表。这一般要求主键不能太长而且插入时的主键最好是按顺序递增，否则对性能有很大影响。	不存在这个问题。

大部分查询只能使用表上的单一索引;在某些情况下，会存在使用多个索引的查询,但是查询优化器通常会低估其成本,它们常常比表扫描还要慢。	不存在这个问题
表增加列，基本上是重建表和索引，会花很长时间。	表增加列，只是在数据字典中增加表定义，不会重建表
存储过程与触发器的功能有限。可用来编写存储过程、触发器、计划事件以及存储函数的语言功能较弱	除支持 pl/pgsql 写存储过程，还支持 perl、python、Tcl 类型的存储过程：pl/perl，pl/python，pl/tcl。 也支持用 C 语言写存储过程。
不支持 Sequence。	支持
不支持函数索引，只能在创建基于具体列的索引。 不支持物化视图。	支持函数索引，同时还支持部分数据索引，通过规则系统可以实现物化视图的功能。
执行计划并不是全局共享的, 仅仅在连接内部是共享的。	执行计划共享
MySQL 支持的 SQL 语法(ANSI SQL 标准)的很小一部分。不支持递归查询、通用表表达式（Oracle 的 with 语句）或者窗口函数（分析函数）。	都 支持

不支持用户自定义类型或域(domain)	支持。
对于时间、日期、间隔等时间类型没有秒以下级别的存储类型	可以精确到秒以下。
身份验证功能是完全内置的，不支持操作系统认证、PAM 认证,不支持 LDAP 以及其它类似的外部身份验证功能。	支持 OS 认证、Kerberos 认证 、Ident 的认证、LDAP 认证、PAM 认证
不支持 database link。有一种叫做 Federated 的存储引擎可以作为一个中转将查询语句传递到远程服务	有 dblink，同时还有一个 dbi-link 的东西，可以连接

器的一个表上,不过,它功能很粗糙并且漏洞很多	到 oracle 和 mysql 上。
Mysql Cluster 可能与你的想象有较大差异。开源的 cluster 软件较少。 复制(Replication)功能是异步的,并且有很大的局限性.例如,它是单线程的(single-threaded),因此一个处理能力更强的 Slave 的恢复速度也很难跟上处理能力相对较慢的 Master.	有丰富的开源 cluster 软件支持。
explain 看执行计划的结果简单。	explain 返回丰富的信息。
类似于 ALTER TABLE 或 CREATE TABLE 一类的操作都是非事务性的.它们会提交未提交的事务，并且不能回滚也不能做灾难恢复	DDL 也是有事务的。

PostgreSQL 主要优势：

1. PostgreSQL 完全免费，而且是 BSD 协议，如果你把 PostgreSQL 改一改，然后再拿去卖钱，也没有人管你，这一点很重要，这表明了 PostgreSQL 数据库不会被其它公司控制。oracle 数据库不用说了，是商业数据库，不开放。而 MySQL 数据库虽然是开源的，但现在随着 SUN 被 oracle 公司收购，现在基本上被 oracle 公司控制，其实在 SUN 被收购之前，MySQL 中最重要的 InnoDB 引擎也是被 oracle 公司控制 的，而在 MySQL 中很多重要的数据都是放在 InnoDB 引擎中的，反正我们公司都是这样的。所以如果 MySQL 的市场范围与 oracle 数据库的市场范 围冲突时，oracle 公司必定会牺牲 MySQL，这是毫无疑问的。

2. 与 PostgreSQl 配合的开源软件很多，有很多分布式集群软件，如 pgpool、pgcluster、slony、plploxy 等等，很容易做读写分离、负载均衡、数据水平拆分等方案，而这在 MySQL 下则比较困难。

3. PostgreSQL 源代码写的很清晰，易读性比 MySQL 强太多了，怀疑 MySQL 的源代码被混淆过。所以很多公司都是基本 PostgreSQL 做二次开发的。

4. PostgreSQL 在很多方面都比 MySQL 强，如复杂 SQL 的执行、存储过程、触发器、索引。同时 PostgreSQL 是多进程的，而 MySQL 是线 程的，虽然并发不高时，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。

目前只想到这些，以后想到再添加，欢迎大家拍砖。

PostgreSQL 与 oracle 或 InnoDB 的多版本实现的差别

PostgreSQL 与 oracle 或 InnoDB 的多版本实现最大的区别在于最新版本和历史版本是否分离存储，PostgreSQL 不分，而 oracle 和 InnoDB 分，而 innodb 也只是分离了数据,索引本身没有分开。

PostgreSQL 的主要优势在于：

1. PostgreSQL 没有回滚段，而 oracle 与 innodb 有回滚段，oracle 与 Innodb 都有回滚段。对于 oracle 与 Innodb 来说， 回滚段是非常重要的，回滚段损坏，会导致数据丢失，甚至数据库无法启动的严重问题。另由于 PostgreSQL 没有回滚段，旧数据都是记录在原先的文件 中，所以当数据库异常 crash 后，恢复时，不会象 oracle 与 Innodb 数据库那样进行那么复杂的恢复，因为 oracle 与 Innodb 恢复时同步 需要 redo 和 undo。所以 PostgreSQL 数据库在出现异常 crash 后，数据库起不来的几率要比 oracle 和 mysql 小一些。

2. 由于旧的数据是直接记录在数据文件中，而不是回滚段中，所以不会象 oracle 那样经常报 ora-01555 错误。

3. 回滚可以很快完成，因为回滚并不删除数据，而 oracle 与 Innodb，回滚时很复杂，在事务回滚时必须清理该事务所进行的修改，插入的记录要删除，更 新的记录要更新回来(见 row_undo 函数)，同时回滚的过程也会再次产生大量的 redo 日志。

4. WAL 日志要比 oracle 和 Innodb 简单，对于 oracle 不仅需要记录数据文件的变化，还要记录回滚段的变化。

PostgreSQL 的多版本的主要劣势在于：

1、最新版本和历史版本不分离存储，导致清理老旧版本需要作更多的扫描，代价比较大，但一般的数据库都有高峰期，如果我们合理安排 VACUUM，这也不是很大的问题，而且在 PostgreSQL9.0 中 VACUUM 进一步被加强了。

2、由于索引中完全没有版本信息，不能实现 Coverage index scan，即查询只扫描索引，直接从索引中返回所需的属性，还需要访问表。而 oracle 与 Innodb 则可以;

进程模式与线程模式的对比

PostgreSQL 和 oracle 是进程模式，MySQL 是线程模式。

进程模式对多 CPU 利用率比较高。

进程模式共享数据需要用到共享内存，而线程模式数据本身就是在进程空间内都是共享的，不同线程访问只需要控制好线程之间的同步。

线程模式对资源消耗比较少。

所以 MySQL 能支持远比 oracle 多的更多的连接。

对于 PostgreSQL 的来说，如果不使用连接池软件，也存在这个问题，但 PostgreSQL 中有优秀的连接池软件软件，如 pgbouncer 和 pgpool，所以通过连接池也可以支持很多的连接。

堆表与索引组织表的的对比

Oracle 支持堆表，也支持索引组织表

PostgreSQL 只支持堆表，不支持索引组织表

Innodb 只支持索引组织表

索引组织表的优势：

表内的数据就是按索引的方式组织，数据是有序的，如果数据都是按主键来访问，那么访问数据比较快。而堆表，按主键访问数据时，是需要先按主键索引找到数据的物理位置。

索引组织表的劣势：

索引组织表中上再加其它的索引时，其它的索引记录的数据位置不再是物理位置，而是主键值，所以对于索引组织表来说，主键的值不能太大，否则占用的空间比较大。

对于索引组织表来说，如果每次在中间插入数据，可能会导致索引分裂，索引分裂会大大降低插入的性能。所以对于使用 innodb 来说，我们一般最好让主键是一个无意义的序列，这样插入每次都发生在最后，以避免这个问题。

由于索引组织表是按一个索引树，一般它访问数据块必须按数据块之间的关系进行访问，而不是按物理块的访问数据的，所以当做全表扫描时要比堆表慢很多，这可能在 OLTP 中不明显，但在数据仓库的应用中可能是一个问题。

PostgreSQL9.0 中的特色功能：

PostgreSQL 中的 Hot Standby 功能

也就是 standby 在应用日志同步时，还可以提供只读服务，这对做读写分离很有用。这个功能是 oracle11g 才有的功能。

PostgreSQL 异步提交（Asynchronous Commit）的功能：

这个功能 oracle 中也是到 oracle11g R2 才有的功能。因为在很多应用场景中，当宕机时是允许丢失少量数据的，这个功能在这样的场景中就特别合适。在 PostgreSQL9.0 中把 synchronous_commit 设置为 false 就打开了这个功能。需要注意的是，虽然设置为了异步提交，当主机宕机时，PostgreSQL 只会 丢失少量数据，异步提交并不会导致数据损坏而数据库起不来的情况。MySQL 中没有听说过有这个功能。

PostgreSQL 中索引的特色功能：

PostgreSQL 中可以有部分索引，也就是只能表中的部分数据做索引，create index 可以带 where 条件。同时 PostgreSQL 中的索引可以反向扫描，所以在 PostgreSQL 中可以不必建专门的降序索引了。