新手机 新应用 新娱乐

PostgreSQL 9.3 培训
Day 1

digoal.zhou

2013/12/5

SKY mobi
新手机 新应用 新娱乐

个人介绍

- 
  周正中 (Digoal.zhou)
- 网名: 德哥
- 杭州斯凯网络科技有限公司(Nasdaq: MOBI)数据库技术经理
- PostgreSQL相关经历介绍
- 2008 (初次接触PostgreSQL, 负责斯凯所有数据库管理工作.)
  - 发布Oracle培训视频(冷备份与恢复,热备份与克隆, RMAN catalog的创建和使用,备份方案设计,RMAN恢复, RMAN数据迁移, RMAN远程复制, RMAN创建DATAGUARD, logminer使用,statspack使用)
  - 2010 - PostgreSQL培训视频(数据库管理) , GreenPlum培训视频(架构, 管理)
- 2010
  - 淘宝华东区数据库交流 - <PostgreSQL互联网应用>
- 2011
  - DTCC中国数据库技术大会 - <PostgreSQLInside>
  - PostgreSQL第一届全国大会 - <PostgreSQL企业应用>, <PostgreSQL 介绍>
  - PostgreSQL中国用户会成立, 在用户会负责PG数据库推广工作 以及 担任PG BBS管理板块版主
  - 组织华东区PostgreSQL数据库技术交流
  - 负责结算,营帐,认证等系统Oracle全面转PostgreSQL项目

SKY mobi
新手机 新应用 新娱乐

个人介绍

- 2012
  - 主讲 PostgreSQL 9.1 2DAY DBA免费培训
  - 华东区PostgreSQL交流 - <PostgreSQL 负载均衡>,<PostgreSQL 容灾>,<PostgreSQL 备份>,<PostgreSQL HA>,<PostgreSQL 日常维护以及调优案例>,<PostgreSQL 9.3 or Future ver upcoming Features>
  - PostgreSQL全国大会 - <PostgreSQL 经验谈>
- 2013
  - PG实时大数据统计项目(友情支持), 每日数据超过1亿的多维度实时分析系统, (仅用1台8核服务器实现超过30个维度的实时统计)
  - 筹办PG全国大会 - <PostgreSQL 9.3 新特性以及9.4+新特性预览>, <PostgreSQL内核动态跟踪>
- 其他
  - 为PostgreSQL社区做出的一些小贡献: 软件, pg_cluster, pg_arch_cloud, pg_monitor_api, pg_cloud, sar_collect, pg_nagios_plug(lock,csvlog)...
  - 文章, 851+篇(涉及PostgreSQL内核,源码,管理,开发,设计,性能优化,安全,水平扩展,复制,空间数据库等)
- 书籍(eBook) - PostgreSQL内核动态跟踪(已完成初稿), PostgreSQL管理与优化(策划中), PostgreSQL企业应用案例(策划中)
- 博客 : http://blog.163.com/digoal@126
- git : https://github.com/digoal
- QQ : 276732431

SKY mobi
新手机 新应用 新娱乐

课程内容

- Day - 1
- PostgreSQL系统概述
- 目标:
- 了解PostgreSQL的发展历程, 国内外PG社区组成和运作,
- 了解PG的特性, 与其他流行关系数据库的比对, 如何安装PostgreSQL等.

- 体系结构
- 目标:
- 了解系统表以及系统表之间的关系, 系统视图, 管理函数等
- 了解PG进程结构
- 了解PG物理结构, 数据库逻辑概貌, 物理概貌, 可靠性等.

- 使用基础
- 目标:
- 了解常用的数据库交互工具的使用
- 了解PG数据库的数据类型体系, 以及表操作

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统概述

- 发展史
- 国内外社区运作
- 特性
- 与其他流行关系数据库的比对
- 安装
- 参考资料

SKY mobi
新手机 新应用 新娱乐

PostgreSQL发展史

- PostgreSQL历史
- 全球社区介绍
- 中国社区介绍
- PostgreSQL发布历程
- PostgreSQL里程碑
- PostgreSQL数据库全球使用情况

SKY mobi
新手机 新应用 新娱乐

PostgreSQL历史

- 1973 *University INGRES* (起源于IBM System R的一系列文档, Michael Stonebraker and Eugene Wong)
- 1982 INGRES
- 1985 post-Ingres
- 1988 POSTGRES version 1 - 1993 version 4 (END)
- 1995 Postgres95 (伯克利大学学生Andrew Yu, Jolly Chen重写了SQL解释器, 替换原项目中的基于Ingres的SQL解释器)
- 1996 更为PostgreSQL, 发布第一个开源版本.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 全球贡献者

- Core Team成员
- Josh Berkus (USA, PostgreSQL Experts Inc.)
  - 主要负责PG推广, 性能测试, 优化, 文档编辑等工作.
- Peter Eisentraut (USA, MeetMe.com)
  - 主要负责了系统建设, 移植, 文档编辑, 国际化, 以及其他增强性的代码工作.
- Magnus Hagander (Sweden, redpill-linpro.se)
  - 帮助维护PostgreSQL WEB主站及基础设施, win32的移植, 以及系统认证等工作.
- Tom Lane (USA, Salesforce)
  - 遍及PostgreSQL代码的各个角落, 包括BUG评估和修复, 性能改进, 优化等.
- Bruce Momjian (USA, EnterpriseDB)
  - 负责维护TODO和FAQ列表, 代码, 发布版本补丁以及培训.
- Dave Page (United Kingdom, EnterpriseDB)
  - 负责pgadmin的开发和维护工作, 同时负责管理postgresql.org主站工程, PostgreSQL的安装程序等.

- 主要贡献者
  - http://www.postgresql.org/community/contributors/
- Committers (git@gitmaster.postgresql.org/postgresql.git)
  - 目前有20位committer. (http://wiki.postgresql.org/wiki/Committers)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 全球赞助商

- PostgreSQL全球赞助商 (最新 http://www.postgresql.org/about/sponsors/)
- 赞助商分级

Platinum	Entities who have provided at least 3 full years of significant and recurring contribution to PostgreSQL.
Gold	Entities who have provided at least 2 full years of significant and recurring contribution to PostgreSQL.
Silver	Entities who have provided at least 1 full year of significant and recurring contribution to PostgreSQL.
Bronze	Entities who have provided a significant contribution in the last 12 months to PostgreSQL.

- 赞助商列表

SKY mobi
新手机 新应用 新娱乐

PostgreSQL中国用户会介绍

- PostgreSQL中国用户会
  - 促进PostgreSQL在中国发展的非盈利组织

- PostgreSQL在中国发展简史
- 199x何伟平建立bbs, 中文网站, 展开文档以及PG网站各板块的翻译工作,
- 2011年全国大会, 广州暨南大学
  - galy牵头组织第一次全国pg大会, 组建中国PG用户会, 成员7人, 负责PG在中国的推广.
  - 成员(galy,孙鹏,泥鳅,少聪,digoal, Louis, 阿弟)
- 2012年全国大会, 北京人民大学
  - 铃木(pg-xc架构师@ntt), simon, magus, Andrew(@Instagram) 等外宾带来的一些
- 2013年全国大会, 杭州斯凯网络
  - 用户会职能工作细化. 成立几个职能小组 (商业支持,网站维护,文档翻译,内核开发,...)

- 国内近期和PostgreSQL相关的线下活动
  - 2011 DTCC
  - 2011 淘宝华东区数据库技术交流
  - 2011 第一届PG全国大会
  - 2012 上海, 广州, 深圳, 北京, 杭州, 成都等地区域PG交流活动
  - 2012 北京, 杭州 免费的2天课时PG DBA培训活动
  - 2013 各地的区域性PG交流,培训活动, ...

SKY mobi
新手机 新应用 新娱乐

PostgreSQL中文BBS统计(截至201312)

http://bbs.pgsqldb.com/client/index.php

基本概况

会员统计

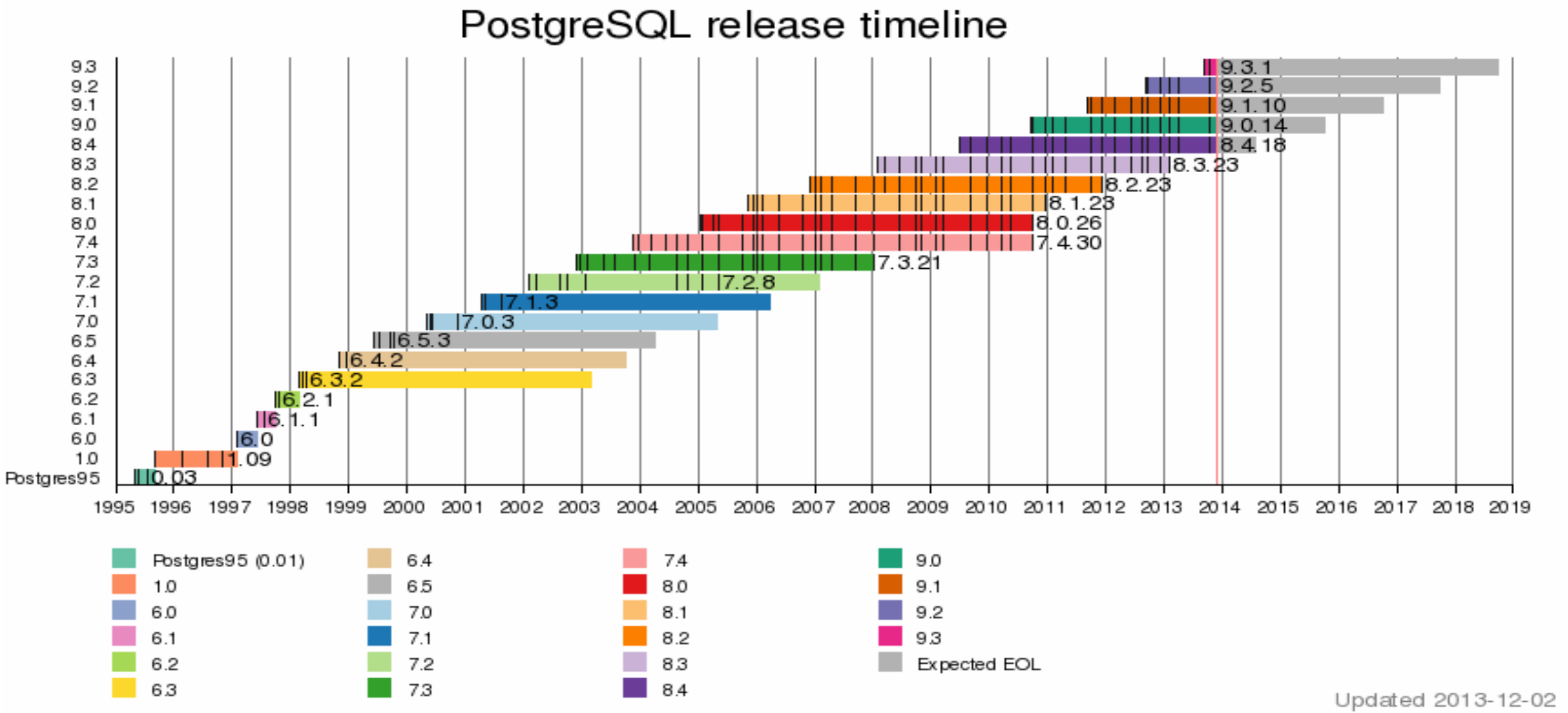
注册会员	7440	发贴会员	3332
管理成员	20	未发贴会员	4108
平均每人发帖数	7.67	发帖会员比例	44.78%
今天注册会员数	0	本月注册会员数	2

论坛统计

版块数	25	平均每日新增帖子数	14.4	最热门的版块	PostgreSQL的论坛
主题数	12697	平均每日注册会员数	1.88	主题数	7893
帖子数	57079			帖子数	40280
平均每个主题被回复次数	3.50			平均每个主题被回复次数	4.10

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 发布历程



SKY mobi
新手机 新应用 新娱乐

PostgreSQL 里程碑

Release	First release	Latest minor version	Latest release	Milestones
0.01	1995-05-01	0.03	1995-07-21	Initial release as Postgres95
1.0	1995-09-05	1.09	1996-11-04	Changed copyright to a more liberal license
6.0	1997-01-29	—		Name change from Postgres95 to PostgreSQL, unique indexes, pg_dumpall utility, ident authentication.
6.1	1997-06-08	6.1.1	1997-07-22	Multi-column indexes, sequences, money data type, GEQO (GEnetic Query Optimizer).
6.2	1997-10-02	6.2.1	1997-10-17	JDBC interface, triggers, server programming interface, constraints.
6.3	1998-03-01	6.3.2	1998-04-07	SQL92 subselect capability, PL/pgTCL
6.4	1998-10-30	6.4.2	1998-12-20	VIEWs and RULEs, PL/pgSQL
6.5	1999-06-09	6.5.3	1999-10-13	MVCC, temporary tables, more SQL statement support (CASE, INTERSECT, and EXCEPT)
7.0	2000-05-08	7.0.3	2000-11-11	Foreign keys, SQL92 syntax for joins
7.1	2001-04-13	7.1.3	2001-08-15	Write-ahead log, Outer joins
7.2	2002-02-04	7.2.8	2005-05-09	PL/Python, OIDs no longer required, internationalization of messages
7.3	2002-11-27	7.3.21	2008-01-07	Schema, Internationalization
7.4	2003-11-17	7.4.30	2010-10-04	Optimization all-round
8.0	2005-01-19	8.0.26	2010-10-04	Native server on Microsoft Windows, savepoints, tablespaces, exception handling in functions, point-in-time recovery
8.1	2005-11-08	8.1.23	2010-12-16	Performance optimization, two-phase commit, table partitioning, index bitmap scan, shared row locking, roles
8.2	2006-12-05	8.2.23	2011-09-26	Performance optimization, online index builds, advisory locks, warm standby
8.3	2008-02-04	8.3.23	2013-02-07	Heap-only tuples, full text search, SQL/XML, ENUM types, UUID types
8.4	2009-07-01	8.4.18	2013-10-10	Windowing functions, default and variadic parameters for functions, column-level permissions, parallel database restore, per-database collation, common table expressions and recursive queries

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 里程碑

Release	First release	Latest minor version	Latest release	Milestones
9.0	2010-09-20	9.0.14	2013-10-10	Built-in binary streaming replication, Hot standby, 64-bit Windows, per-column triggers and conditional trigger execution, exclusion constraints, anonymous code blocks, named parameters, password rules
9.1	2011-09-12	9.1.10	2013-10-10	Synchronous replication, per-column collations, unlogged tables, *k*-nearest-neighbor indexing, serializable snapshot isolation, writeable common table expressions, SE-Linux integration, extensions, SQL/MED attached tables (Foreign Data Wrappers), triggers on views
9.2	2012-09-10	9.2.5	2013-10-10	Cascading streaming replication, index-only scans, native JSON support, improved lock management, range types, pg_receivexlog tool, space-partitioned GiST indexes
9.3	2013-09-09	9.3.1	2013-10-10	Custom background workers, data checksums, dedicated JSON operators, LATERAL JOIN, faster pg_dump, new pg_isready server monitoring tool, trigger features, view features, writeable foreign tables, replication improvements

SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库全球使用情况

- 生物制药 {Affymetrix(基因芯片), 美国化学协会, gene(结构生物学应用案例), …}
- 电子商务 { CD BABY, etsy(与淘宝类似), whitepages, flightstats, Endpoint Corporation …}
- 学校 {加州大学伯克利分校, 哈佛大学互联网与社会中心, .LRN, 莫斯科国立大学, 悉尼大学, …}
- 金融 {Journyx, LLC, trusecommerce(类似支付宝), …}
- 游戏 {MobyGames, …}
- 政府 {美国国家气象局, 印度国家物理实验室, 联合国儿童基金, 美国疾病控制和预防中心, 美国国务院, …}
- 医疗 {calorieking, 开源电子病历项目, shannon医学中心, …}
- 制造业 {Exoteric Networks}
- 媒体 {IMDB.com, 美国华盛顿邮报国会投票数据库, MacWorld, 绿色和平组织, …}
- 开源项目 {Bricolage, Debian, FreshPorts, FLPR, PostGIS, SourceForge, OpenACS, Gforge, …}
- 零售 {ADP, CTC, Safeway, Tsutaya, Rockport, …}
- 科技 {Sony, MySpace, Yahoo, Afilias, APPLE, 富士通, Omniti, Red Hat, Sirius IT, SUN, 国际空间站, Instagram, Disqus, …}
- 通信 {Cisco, Juniper, NTT(日本电信), Optus, Skype, Tlestra(澳洲电讯), …}
- 物流 {SF}
- More : http://www.postgresql.org/about/users/

SKY mobi
新手机 新应用 新娱乐

PostgreSQL特性

- PostgreSQL 许可
- PostgreSQL 支持的SQL标准
- PostgreSQL 扩展特性介绍
- 企业特性
- PostgreSQL In BigData
- PostgreSQL In Cloud
- PostgreSQL 与其他数据库(Oracle, MySQL, Hadoop)的应用案例对比

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 许可

- 许可, http://opensource.org/licenses/postgresql
- This is a template license. The body of the license starts at the end of this paragraph. To use it, say that it is The PostgreSQL License, and then substitute the copyright year and name of the copyright holder into the body of the license. Then put the license into a prominent file ("COPYRIGHT", "LICENSE" or "COPYING" are common names for this file) in your software distribution.
- Copyright (c) $YEAR, $ORGANIZATION
- Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.
- IN NO EVENT SHALL $ORGANISATION BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF $ORGANISATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- $ORGANISATION SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND $ORGANISATION HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

SKY mobi
新手机 新应用 新娱乐

许可 PostgreSQL vs MySQL

- PostgreSQL(Copyfree) license VS MySQL(Copyleft) license
- http://www.wikivs.com/wiki/Copyfree_vs_Copyleft
- PostgreSQL Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

- MySQL's source code is available under terms of the GNU General Public License, which also fits the Free Software and Open Source definitions and conforms to the Debian Free Software Guidelines (but not to the Copyfree Standard). It is also available under a proprietary license agreement, which is typically intended for use by those who wish to release software incorporating MySQL code without having to release the source code for the entire application. In practical terms, this means that MySQL can be distributed with or without source code, as can PostgreSQL, but to distribute without source code in the case of MySQL requires paying Oracle for a MySQL Commercial License.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 遵循的SQL标准

- 标准,
  - ISO/IEC 9075:2011
  - ISO/IEC 9075-1 Framework (SQL/Framework)
  - ISO/IEC 9075-2 Foundation (SQL/Foundation)
  - ISO/IEC 9075-3 Call Level Interface (SQL/CLI)
  - ISO/IEC 9075-4 Persistent Stored Modules (SQL/PSM)
  - ISO/IEC 9075-9 Management of External Data (SQL/MED)
  - ISO/IEC 9075-10 Object Language Bindings (SQL/OLB)
  - ISO/IEC 9075-11 Information and Definition Schemas (SQL/Schemata)
  - ISO/IEC 9075-13 Routines and Types using the Java Language (SQL/JRT)
  - ISO/IEC 9075-14 XML-related specifications (SQL/XML)

- http://www.postgresql.org/docs/9.3/static/features.html
- http://www.postgresql.org/docs/9.3/static/features-sql-standard.html
- http://www.postgresql.org/docs/9.3/static/unsupported-features-sql-standard.html

- 灰色部分PostgreSQL未涉及

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 扩展特性

- 标准扩展特性矩阵

Backend

- 64-bit large objects
- Advisory locks
- ALTER object IF EXISTS
- ALTER TABLE ... ADD UNIQUE/PRIMARY KEY USING INDEX
- Autovacuum enabled by default
- ▲ Cascading streaming replication
- Changing column types (ALTER TABLE .. ALTER COLUMN TYPE)
- Checksum on data pages
- ▲ Column level permissions
- ▲ Common Table Expressions (WITH RECURSIVE)
- ▲ Concurrent GiST indexes
- ▲ COPY from/to STDIN/STDOUT
- ▲ COPY with arbitrary SELECT
- ▲ Crash-safe GiST indexes
- ▲ CREATE TABLE ... (LIKE) with foreign tables, views and composite types
- ▲ CSV support for COPY
- Custom background workers
- Default permissions
- Deferrable unique constraints
- ▲ Disk based FSM
- ▲ Dollar Quoting
- DROP object IF EXISTS
- ▲ Exclusion constraints
- ▲ EXPLAIN (BUFFERS) support
- Extension package installation
- ▲ Foreign data wrappers
- ▲ Foreign Tables
- Full code coverage generation support
- ▲ GRANT/REVOKE ON ALL TABLES/SEQUENCES/FUNCTIONS
- Holdable cursors
- ▲ Hot Standby
- ▲ Improved set of JSON functions and operators
- ▲ INSERT/UPDATE/DELETE RETURNING
- ▲ Integrated autovacuum daemon
- Join removal support
- LATERAL clause
- Loadable plugin infrastructure for monitoring the planner
- ▲ Materialized views
- Multiple autovacuum workers
- Multiple input aggregates
- Multirow VALUES
- MVCC safe CLUSTER
- Named restore points
- ON COMMIT clause for CREATE TEMPORARY TABLE
- ORDER BY NULLS FIRST/LAST
- ▲ Parallel pg_dump
- Payload support for LISTEN/NOTIFY
- ▲ Per tablespace support for GUCs
- ▲ Per user/database server configuration settings
- ▲ Pg_basebackup tool
- ▲ pg_receivexlog tool
- ▲ Point-in-Time Recovery
- Prefix support for text search synonym dictionary
- Row-wise comparison
- ▲ Savepoints
- ▲ SELECT FOR NO KEY UPDATE/SELECT FOR KEY SHARE lock modes
- ▲ Serializable Snapshot Isolation
- ▲ SP-GiST indexes for range types
- SQLDA support for ECPG
- SQL-standard information schema
- SQL standard interval handling
- ▲ Streaming-only cascading replication
- ▲ Streaming Replication
- Support for anonymous shared memory

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 扩展特性

- 标准扩展特性矩阵

- Synchronous replication
- TABLE statement
- Temporary VIEWs
- Two Phase commit
- Txid functions
- Typed tables
- UNNEST/array_agg
- Updatable views
- Updateable cursors
- Version aware psql
- Visibility map
- WAL-safe B-Tree Indexes
- Warm Standby
- Window functions
- Writable Common Table Expressions
- Writable Foreign Data Wrappers
- XML, JSON and YAML output for EXPLAIN

Performance

- Asynchronous Commit
- Automatic plan invalidation
- Background Checkpointer
- Background Writer
- Cross datatype hashing support
- Distributed checkpointing
- Foreign keys marked as NOT VALID
- Full Text Search
- GIN (Generalized Inverted Index) Indexes
- GIN indexes partial match
- GiST (Generalized Search Tree) Indexes
- Hashing support for DISTINCT/UNION/INTERSECT/EXCEPT
- Hashing support for FULL OUTER JOIN, LEFT OUTER JOIN and RIGHT OUTER JOIN
- Hashing support for NUMERIC
- Heap Only Tuples (HOT)
- Indexes on expressions
- Index-only scans
- Index support for IS NULL
- Inlining of SQL-functions
- In-memory Bitmap Indexes
- K-nearest neighbor GiST support
- Lazy XID allocation
- Multi-core scalability for read-only workloads
- Multiple temporary tablespaces
- Non-blocking CREATE INDEX
- Outer Join reordering
- Parallel restore
- Partial sort capability (top-n sorting)
- SELECT ... FOR UPDATE/SHARE NOWAIT
- Semi- and Antijoins
- Shared row level locking
- Space-Partitioned GiST Indexes
- Synchronized sequential scanning
- Table partitioning
- Tablespaces
- Unlogged tables
- WAL Buffer auto-tuning

Security

- GSSAPI support
- Large object access controls
- Native LDAP authentication
- Native RADIUS authentication
- Per user/database connection limits
- ROLES
- Search+bind mode operation for LDAP authentication
- security_barrier option on views
- Security Service Provider Interface (SSPI)
- SSL certificate validation in libpq
- SSL client certificate authentication
- SSPI authentication via GSSAPI

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 扩展特性

- 标准扩展
- 特性矩阵

Network

Full SSL support

IPv6 Support

V2 client protocol

V3 client protocol

Internationalisation

Column-level collation support

Database level Collation

EUC_JIS_2004/ SHIFT_JIS_2004 support

Multibyte encoding support, incl. UTF8

Multiple language support

Unicode string literals and identifiers

UTF8 support on Windows

Platforms

Microsoft Visual C++ Support

Native Windows Port

Spinlock support for the SuperH hardware platform

Sun Studio compiler on Linux

Windows x64 support

Datatypes

Arrays of compound types

▲ Array support

▲ ENUM data type

GUID/UUID data type

ISO 8601 interval syntax

▲ JSON data type

NULLs in Array

▲ Range types

smallserial type

Type modifier support

XML data type

Functions and triggers

ALTER TABLE ENABLE/DISABLE TRIGGER

ALTER TABLE / ENABLE REPLICA TRIGGER/RULE

▲ Column level triggers

▲ Event triggers

Function argument names

ORDER BY support within aggregates

Per function GUC settings

Per function statistics

RETURN QUERY EXECUTE

RETURNS TABLE

▲ Statement level triggers

Statement level TRUNCATE triggers

▲ Triggers on views

Variadic functions

WHEN clause for CREATE TRIGGER

Procedural Languages

▲ CASE in pl/pgsql

CONTINUE statement for PL/pgSQL

DO statement for pl/perl

▲ DO statement for pl/pgsql

▲ EXCEPTION support in PL/pgSQL

▲ EXECUTE USING in PL/pgSQL

▲ FOREACH IN ARRAY in pl/pgsql

IN/OUT/INOUT parameters for pl/pgsql and PL/SQL

Named parameters

Non-superuser language creation

Number of function arguments increased to 100

pl/pgsql installed by default

Polymorphic functions

Python 3 support for pl/python

Qualified function parameters

▲ RETURN QUERY in pl/pgsql

ROWS and COST specification for functions

Scrollable and updatable cursor support for pl/pgsql

▲ SQLERRM/SQLSTATE for pl/pgsql

Unicode object support in PL/p allows the as

User defined exceptions

Validator function for pl/perl

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 扩展特性

- 标准扩展
- 特性矩阵

SKY mobi
新手机 新应用 新娱乐

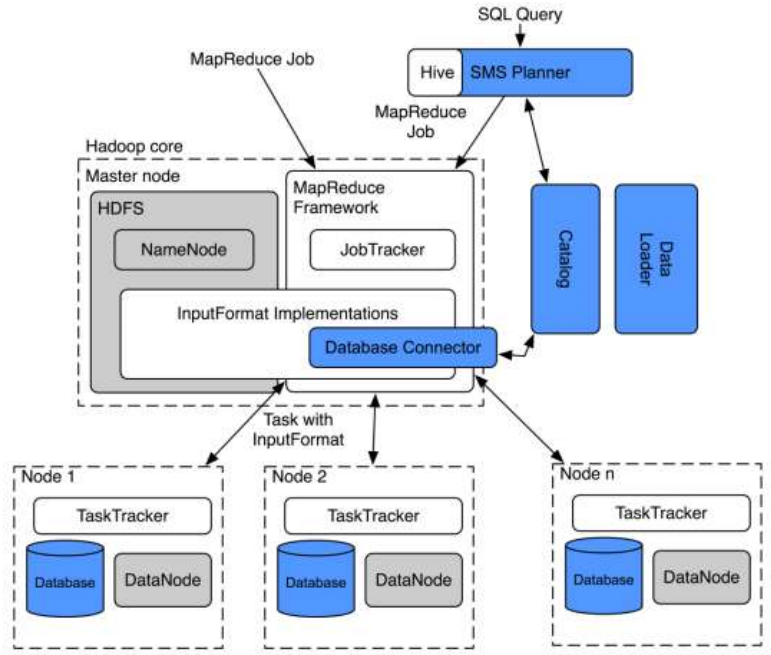
PostgreSQL 企业特性

- 安全性,
  - 认证过程加密
  - 数据传输过程支持加密
  - 基于角色的权限控制
- 稳定性,
  - 良好的代码管理机制(代码风格统一, 只有committer有权限提交代码).
  - 成熟的测试标准(Basic system testing,Regression test suite,Other run time testing(valgrind, gprof, oprofile, ...),unit testing, static analysis, model checking...)
- 可靠性,
  - WAL日志, 确保已提交事务不会因为数据库崩溃导致信息丢失
  - 在开启归档以及基础备份的前提下, 可以将数据库恢复到过去任意时间点
- 可用性(暴力破坏演示), 拔网线, 拔电源, 删数据文件, 格式化硬盘, 拔硬盘, 拔存储光纤线, 电击, 地震, 火烧, ...
  - 9.0开始支持数据库流复制, 9.1开始支持同步流复制.
  - 在配置流复制备机的前提下, 可以确保主数据库被破坏后有一份实时的备库可以使用.
  - 配合集群切换软件可以实现高可用.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL In BigData

- BigSQL (整合了pg和hadoop的一个开源项目)
  - http://www.bigsql.org/se/
- Cloudera Manager DB
  - http://www.cloudera.com/content/cloudera/en/home.html
- Hadoopdb (耶鲁大学的一个开源项目) SQL to MapReduce to SQL (SMS) Planner
  - http://hadoopdb.sourceforge.net/guide/
  - http://cs-www.cs.yale.edu/homes/dna/papers/hadoopdb-demo.pdf
- pg-xc (NTT主导的一个开源的分布式存储PostgreSQL)
  - http://sourceforge.net/apps/mediawiki/postgres-xc/index.php?title=Main_Page
- stormDB (pg-xc的一家商业支持公司)
  - http://www.stormdb.com/
- Citusdb (核心是SQL解析器以及FDW)
  - http://www.citusdata.com/
- PL/Proxy (一个非常精巧的PG代理插件)
- Greenplum(成熟高效的PG bigdata商业方案)
- Madlib(开源可扩展的分析计算库,支持pg,gp
- Matlab



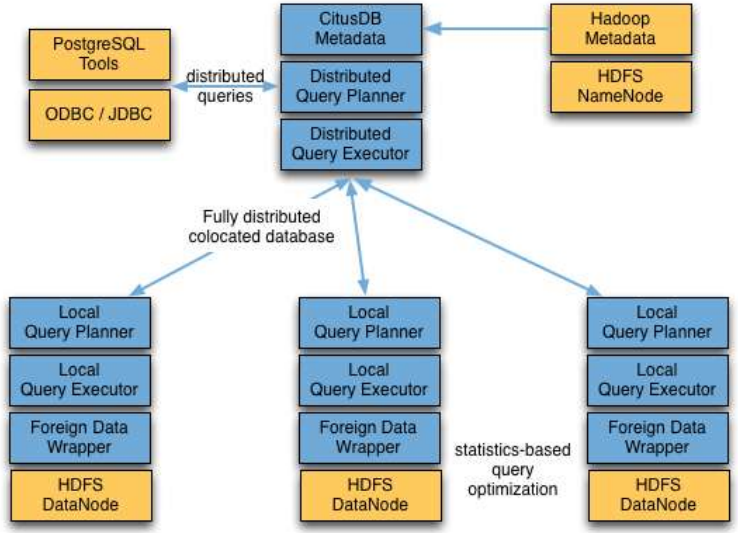
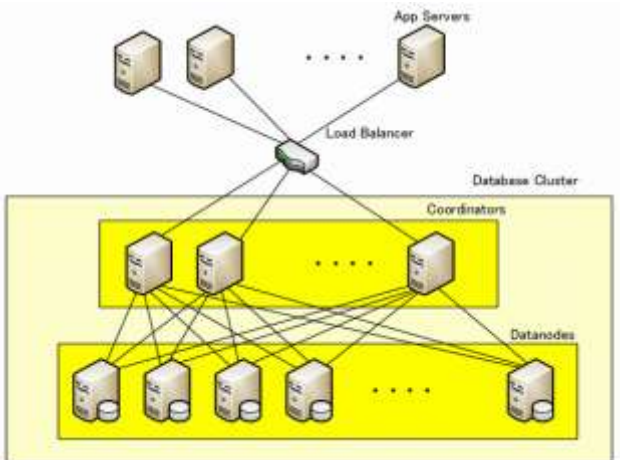


Figure 1: The HadoopDB Architecture



SKY mobi
新手机 新应用 新娱乐

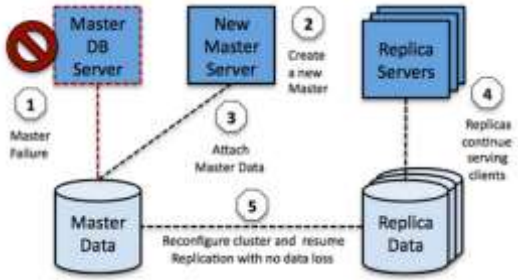
PostgreSQL In BigData

- 测试参考
- http://blog.163.com/digoal@126/blog/#m=0&t=1&c=fks_084069085087089068081083086095085080082075083081086071084
- PL/Proxy
  - http://blog.163.com/digoal@126/blog/static/16387704020104111304328/
  - http://blog.163.com/digoal@126/blog/static/16387704020104111113114315/
  - http://blog.163.com/digoal@126/blog/static/16387704020119253563089 5/
  - http://blog.163.com/digoal@126/blog/static/16387704020131022425437 65/
- Postgres-XC
  - http://blog.163.com/digoal@126/blog/static/16387704020121952051174/
  - http://blog.163.com/digoal@126/blog/static/16387704020133292915600/
  - http://blog.163.com/digoal@126/blog/static/16387704020133323359 33/
  - http://blog.163.com/digoal@126/blog/static/16387704020133564048 21/
- CitusDB
  - http://blog.163.com/digoal@126/blog/static/16387704020132190314687 6/
  - http://blog.163.com/digoal@126/blog/static/16387704020134222140958/

SKY mobi
新手机 新应用 新娱乐

PostgreSQL In Cloud (Private & Public)

- Postgres Plus Cloud Server



- Heroku Postgres

heroku

- ElephantSQL PostgreSQL as a Service

- Salesforce

salesforce

- CloudFoundry

BETA
CLOUD FOUNDRY™
DEPLOY & SCALE YOUR APPLICATIONS IN SECONDS

- Vmware vSphere

vmware

- AWS, Amazon RDS

amazon web services

SKY mobi
新手机 新应用 新娱乐

PostgreSQL与其他数据库的对比

- (建议在安装完 PostgreSQL后大家实际操作体验一下)
- 性能对比案例输出 (postgresql, oracle, mysql, hadoop), 挑选几个可以反映PostgreSQL特性的经典案例
- 数据插入性能比对. MySQL , Oracle VS PostgreSQL
  - http://www.penglixun.com/tech/database/case_about_innodb_faster_than_oracle.html
- PostgreSQL ip范围类型的例子, 该场景下pg性能是mysql性能的20+倍.
  - http://blog.163.com/digoal@126/blog/static/163877040201257010292 22/
- 大数据导入的优化案例, 性能远超同类数据库.
  - http://blog.163.com/digoal@126/blog/static/163877040201392641033482/
- 大数据实时分析的案例 (利用current_xid以及xid_snapshot做切片)
  - http://blog.163.com/digoal@126/blog/static/163877040201343111447 55/
- 数据预热的案例 (pgfincore)
- 从柱状图中快速读取需要大运算量的TOP数据
- 使用hll数据类型快速统计唯一值和新增值
- TOAST的优化案例
  - http://blog.163.com/digoal@126/blog/static/163877040201309310404 44/
- 大数据的例子, plproxy带来的线性性能提升.
- 性能优化综合案例
  - http://blog.163.com/digoal@126/blog/static/163877040201221382150858/

SKY mobi
新手机 新应用 新娱乐

PostgreSQL与其他数据库的对比

- 使用pgbench和mysql_fdw测试mysql性能的例子
- http://blog.163.com/digoal@126/blog/static/163877040201362355123969/
- 使用sysbench测试mysql和postgresql性能的例子
- http://blog.163.com/digoal@126/blog/static/163877040201341441042613/
- http://blog.163.com/digoal@126/blog/static/16387704020134145495l5/

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- 一、测试模型：
- 包含12张业务表，每个事务包含12个SQL，每个SQL向一张表做INSERT，做完12个SQL即完成一个事务。

- 用一个C API编写的程序连接MySQL，不断执行如下操作

- 开始事务：START TRANSACTION;
- 每张表插入一行：INSERT INTO xxx VALUES (val1,val2,…); #一共12次
- 提交事务：COMMIT;

- 通过一个Shell脚本来启动32个测试程序并发测试

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- 二、测试环境：
- 1. Oracle, MySQL 测试机型
- DELL R910
- CPU：Intel(R) Xeon(R) CPU E7530 @ 1.87GHz 四路48线程
- 内存：32* 4G 128G
- 存储：FusionIO 640G MLC

- 2. PostgreSQL 测试机型1
- IBM x3850 X5
- CPU : Intel(R) Xeon(R) CPU X7560 @ 2.27GHz 四路32线程
- 内存 : 8 * 8GB 64G
- 存储：OCZ RevoDrive3X2 480GB

- 3. PostgreSQL 测试机型2
- DELL R610
- CPU : Intel(R) Xeon(R) CPU E5504 @ 2.00GHz 2路8线程 (电源功率不够降频到1.6GHZ)
- 内存 : 12 * 8GB 96G
- 存储：OCZ RevoDrive3 240GB

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- 三、测试结果：
- Oracle测试结果 :
- 1. R910 Oracle单实例
- 测试人：童家旺，支付宝
- TPS：稳定值2000,峰值2600 （我没参与测试，也没有报告，无法确定详情）
- 我的补充：Oracle已经是调优的过的，请相信我们的Oracle DBA不是吃素的。我把听Oracle DBA描述的只言碎语随便写下，Oracle跑到后面TPS也是有所下降，不是能一直100%稳定，最后CPU已经吃尽了，所以基本上再怎么优化提升的幅度会比较小。

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- MySQL测试结果 :
- 2. R910 MySQL多实例 Percona 5.1.60-13.1 修改版
- 测试人：彭立勋，B2B
- TPS：峰值1200*4，谷值0，均值950*4
- 重要配置：（在测试3的基础上）
- innodb_aio_pending_ios_per_thread=1024
- 测试人描述：
- 经过对测试3的分析，可以发现，InnoDB已经标记了很多Page到Flush_list，但是并没有被即时的回写，可以在INNODB_BUFFER_POOL_PAGES系统表中发现很页flush_type=2，即在Flush_list中。
- 经过review代码，发现InnoDB申请的AIO队列的长度只有256，由常量OS_AIO_N_PENDING_IOS_PER_THREAD（os0file.h）定义。将此常量修改为InnoDB的参数后，重新测试，可以使FusionIO的IOPS达到7K～18K，IO利用率得以提升，整体性能已经超越Oracle，但存在严重的低谷，大约每10s一次。

SKYY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- PostgreSQL测试结果 :
- CONFIGURE = '--prefix=/opt/pgsql9.3.1' '--with-pgport=1921' '--with-perl' '--with-python' '--with-tcl' '--with-openssl' '--with-pam' '--without-ldap' '--with-libxml' '--with-libxslt' '--enable-thread-safety' '--with-wal-blocksize=64' '--with-wal-segsize=64' '--with-blocksize=32' '--with-segsize=2'
- 3. IBM x3850 X5的测试结果(理想中应该能达到3万以上的tps, 但是没有达到, 原因是插了4路CPU, 但是只插2组内存模组)
- postgres@db-192-168-173-242-> pgbench -M prepared -n -r -f ./test.sql -c 32 -j 8 -T 60 -h $PGDATA -p 1921 -U digoal digoal
- transaction type: Custom query
- scaling factor: 1
- query mode: prepared
- number of clients: 32
- number of threads: 8
- duration: 60 s
- number of transactions actually processed: 1469319
- tps = 24487.282793 (including connections establishing)
- tps = 24502.402484 (excluding connections establishing)
- statement latencies in milliseconds:
- 1.303527 select f_test1();

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- 4. DELL R610的测试结果
- pg93@db-172-16-3-150-> pgbench -M prepared -n -r -f ./test.sql -c 16 -j 4 -T 60 -h $PGDATA -p 1921 -U postgres digoal
- transaction type: Custom query
- scaling factor: 1
- query mode: prepared
- number of clients: 16
- number of threads: 4
- duration: 60 s
- number of transactions actually processed: 909280
- tps = 15151.787574 (including connections establishing)
- tps = 15155.568739 (excluding connections establishing)
- statement latencies in milliseconds:
- 1.053459 select f_test1();

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

数据库	硬件	TPS
Oracle	DELL R910 CPU : Intel(R) Xeon(R) CPU E7530 @ 1.87GHz 四路48线程 MEM : 32* 4G 128G 存储 : FusionIO 640G MLC	稳定值2000,峰值2600
MySQL Percona 5.1.60-13.1 修改版	DELL R910 CPU : Intel(R) Xeon(R) CPU E7530 @ 1.87GHz 四路48线程 MEM : 32* 4G 128G 存储 : FusionIO 640G MLC	峰值1200*4， 谷值0， 均值950*4
PostgreSQL 9.3.1	IBM x3850 X5 CPU : Intel(R) Xeon(R) CPU X7560 @ 2.27GHz 四路32线程 内存 : 8 * 8GB 64G 存储：OCZ RevoDrive3X2 480GB	稳定值24487
PostgreSQL 9.3.1	DELL R610 CPU : Intel(R) Xeon(R) CPU E5504 @ 2.00GHz 2路8线程 (电源功率不够降频到1.6GHZ) 内存 : 12 * 8GB 96G 存储：OCZ RevoDrive3 240GB	稳定值15151

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- PostgreSQL测试脚本

```
CREATE OR REPLACE FUNCTION f_test1()   RETURNS void
LANGUAGE plpgsql
STRICT
AS $function$
declare
begin
insert into t1(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t2(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t3(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t4(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t5(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t6(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t7(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t8(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t9(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t10(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t11(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
insert into t12(c1,c2,c3,c4,c5,c6,c7) values ('test1','test2','test3','test4','test5','test6','test7');
return;
exception when others then
return;
end;
$function$;
```

SKY mobi
新手机 新应用 新娱乐

数据插入性能比对 MySQL , Oracle VS PostgreSQL

- PostgreSQL测试脚本
- create table t1 (id serial4 primary key, c1 text, c2 text, c3 text, c4 text, c5 text, c6 text, c7 text, c8 timestamp default now());
  create table t2 (like t1 including all);
  create table t3 (like t1 including all);
  create table t4 (like t1 including all);
  create table t5 (like t1 including all);
  create table t6 (like t1 including all);
  create table t7 (like t1 including all);
  create table t8 (like t1 including all);
  create table t9 (like t1 including all);
  create table t10 (like t1 including all);
  create table t11 (like t1 including all);
  create table t12 (like t1 including all);

- vi test.sql
- select f_test1();
- pgbench -M prepared -n -r -f ./test.sql -c 16 -j 4 -T 600 -h $PGDATA -p 1921 -U postgres digoal

SKY mobi
新手机 新应用 新娱乐

PostgreSQL VS MySQL 范围匹配查询

- MySQL使用场景
- CREATE TABLE ip_address_pool (
- id int(10) NOT NULL AUTO_INCREMENT COMMENT '自增主键',
- start_ip varchar(20) NOT NULL COMMENT '起始ip',
- end_ip varchar(20) NOT NULL COMMENT '截止ip',
- province varchar(128) NOT NULL COMMENT '省名',
- city varchar(128) NOT NULL COMMENT '城市',
- region_name varchar(128) NOT NULL COMMENT '地区名',
- company_name varchar(128) NOT NULL COMMENT '公司名',
- start_ip_decimal bigint(10) DEFAULT NULL,
- end_ip_decimal bigint(10) DEFAULT NULL,
- PRIMARY KEY (id),
- KEY idx_start_ip_Decimal (start_ip_decimal),
- KEY idx_end_ip_Decimal (end_ip_decimal)
-) ENGINE=InnoDB AUTO_INCREMENT=436820 DEFAULT CHARSET=utf8 COMMENT='ip地址对应表';

SKY mobi
新手机 新应用 新娱乐

PostgreSQL VS MySQL 范围匹配查询

- MySQL 中的查询, 范围匹配.
- select
- province,
- start_ip_Decimal as startIpDecimal,
- end_ip_Decimal as endIpDecimal
- from ip_address_pool
- where
- #{ip}>=start_ip_Decimal and
- #{ip}<=end_ip_Decimal;

- QPS 只能达到几千.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL VS MySQL 范围匹配查询

- PostgreSQL 场景1, 与PostgreSQL同样使用范围匹配
- CREATE TABLE ip_address_pool (
- id serial8 primary key,
- start_ip inet NOT NULL ,
- end_ip inet NOT NULL ,
- province varchar(128) NOT NULL ,
- city varchar(128) NOT NULL ,
- region_name varchar(128) NOT NULL ,
- company_name varchar(128) NOT NULL ,
- start_ip_decimal bigint ,
- end_ip_decimal bigint
-) ;
- create index idx_ip_address_pool_ip on ip_address_pool (start_ip_decimal,end_ip_decimal);

- **QPS : 3606**

SKY mobi
新手机 新应用 新娱乐

PostgreSQL VS MySQL 范围匹配查询

- PostgreSQL 场景2, 使用PostgreSQL ver9.2+的特性, 自定义iprange范围类型.
- create type iprange as range (subtype=inet);

- CREATE TABLE ip_address_pool_2 (
- id serial8 primary key,
- ip_segment iprange NOT NULL ,
- province varchar(128) NOT NULL ,
- city varchar(128) NOT NULL ,
- region_name varchar(128) NOT NULL ,
- company_name varchar(128) NOT NULL
-) ;

- CREATE INDEX ip_address_pool_2_range ON ip_address_pool_2 USING gist (ip_segment);

SKY mobi
新手机 新应用 新娱乐

PostgreSQL VS MySQL 范围匹配查询

- PostgreSQL 场景3, 使用PostgreSQL ver9.2+的特性, 系统内建的int8range范围类型.
- CREATE TABLE ip_address_pool_3 (
- id serial8 primary key,
- start_ip inet NOT NULL ,
- end_ip inet NOT NULL ,
- province varchar(128) NOT NULL ,
- city varchar(128) NOT NULL ,
- region_name varchar(128) NOT NULL ,
- company_name varchar(128) NOT NULL ,
- ip_decimal_segment int8range
-) ;
- insert into ip_address_pool_3 (id,start_ip,end_ip,province,city,region_name,company_name,ip_decimal_segment) select id,start_ip,end_ip,province,city,region_name,company_name,int8range(start_ip_decimal,end_ip_decimal+1) from ip_address_pool;
- CREATE INDEX ip_address_pool_3_range ON ip_address_pool_3 USING gist (ip_decimal_segment);

SKY mobi
新手机 新应用 新娱乐

PostgreSQL VS MySQL 范围匹配查询

- PostgreSQL 场景3, 使用PostgreSQL ver9.2+的特性, 系统内建的int8range范围类型.
- vi ip_test.sql
- \setrandom ip 0 2094967294
- select province,ip_decimal_segment from ip_address_pool_3 where ip_decimal_segment @> :ip::int8;

- pgbench -M prepared -c 8 -j 8 -f ./ip_test.sql -n -T 60 -h 127.0.0.1 -U postgres postgres
- **QPS : 80171**

SKY mobi
新手机 新应用 新娱乐

大数据导入的优化案例

- http://blog.163.com/digoal@126/blog/static/16387704020139264103348 2/
- 某运营商数据采集场景, 入库文件约300MB每个, 入库速度约100MB/s, 而直接拷贝文件的速度远远不止这个速度.
- 测试场景 :
- 8核 Intel(R) Xeon(R) CPU E5504 @ 2.00GHz (降频到1.6GHz)
- CentOS 6.4 x64, 硬盘 10K转SAS盘.
- PostgreSQL 9.3.1
- '--prefix=/home/pg93/pgsql9.3.1' '--with-pgport=1921' '--with-perl' '--with-tcl' '--with-python' '--with-openssl' '--with-pam' '--without-ldap' '--with-libxml' '--with-libxslt' '--enable-thread-safety' '--with-wal-blocksize=16' '--enable-dtrace' '--enable-debug'
- 测试表
- create table t (id int, c1 text, c2 text, c3 text, c4 text, c5 text, c6 text, c7 text, c8 text, c9 text, c10 text, c11 text, c12 text, c13 timestamp);
- 测试数据
- insert into t select generate_series(1,610000), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), md5(random()::text), clock_timestamp();
- 平均每个BLOCK存储18条记录. 平均每条记录455字节.

SKY mobi
新手机 新应用 新娱乐

大数据导入的优化案例

- 把数据拷贝到文件中, 用以模拟数据导入测试
- digoal=# copy t to '/home/pg93/t.dmp' with (header off);
- COPY 0 610000
- digoal=# \! ls -lh /home/pg93/t.dmp
- -rw-r--r-- 1 pg93 pg93 250M Oct 26 15:07 /home/pg93/t.dmp

- pgbench测试脚本
- pg93@db-172-16-3-150-> vi test.sql
- copy t from '/home/pg93/t.dmp' with (header off);

- 测试, 8个连接, 每个连接执行4次.
- pgbench -M prepared -n -r -f ./test.sql -c 8 -j 4 -t 4

- 每秒约导入91MB 或 22.3万条记录.

SKY mobi
新手机 新应用 新娱乐

大数据导入的优化案例

- 优化1
- 更改为unlogged table.
- digoal=# update pg_class set relpersistence='u' where relname='t';
- digoal=# update pg_class set relpersistence='u' where relname='idx_t_id';

- 每秒约导入106MB 或 25.8万条记录.
- 提升不明显, 原因 :
  - 扩展block的锁为排他锁. (extend . ExclusiveLock)
  - 每次扩展1个块
  - 锁介绍可参考 : http://blog.163.com/digoal@126/blog/static/163877040201391674922879/

SKY mobi
新手机 新应用 新娱乐

大数据导入的优化案例

- 优化2 (仅针对锁的一种测试优化, 实际场景中使用不合适)
- 保留测试表的最后一个数据块的最后一条记录(heap表的物理结构后面会讲到)
- 因为回收垃圾只回收尾部的全空白BLOCK, 所以这样删除可以保留表的高水位.
- delete from t where ctid<>(select max(ctid) from t);
- 更新fsm, 导入数据时不需要extend block.
- vacuum (freeze,verbose,analyze) t;
- checkpoint;

- 每秒约导入188MB 或 45.77万条记录.
- 提升较为明显.

SKY mobi
新手机 新应用 新娱乐

大数据导入的优化案例

- 优化3
- 在优化2的基础上, 将硬盘更换为OCZ RevoDrive3 240G pci-e
- 每秒约导入236MB 或 54.3万条记录.

- 优化4
- 删除索引, 纯粹的文本导入
- drop index idx_t_id;
- 每秒约导入285.5MB 或 65.7万条记录.

- 优化5, (优化2不切合实际,) 所以这里使用加大block size来减少extend block的次数.
- 使用32KB的blocksize
- 每秒约导入330MB 或 76万条记录.

- 优化6, 采用32核的机器, 提高导入并行度.
- 每秒约导入1097.3MB 或 267.6 万条记录(平均每条记录455字节).

SKY mobi
新手机 新应用 新娱乐

从柱状图中快速读取需要大运算量的TOP数据

- http://blog.163.com/digoal@126/blog/static/16387704020137101053538 62/
- 创建测试表
- digoal=# create table test_1 (id serial4 primary key, info text, appid int, crt_time timestamp);
- digoal=# insert into test_1 (info,appid,crt_time) select md5(random()::text),round(10000*random())::int,clock_timestamp() from generate_series(1,2000000);
- 1. 从explain输出的信息中评估结果条数(存在一定偏差).
- 可以省去实际执行SQL的开销.
- digoal=# select count(*) from test_1 where appid=1;
- 189
- (1 row)
- digoal=# explain select * from test_1 where appid=1;
- QUERY PLAN
- ----------
- Seq Scan on test_1 (cost=0.00..45619.00 rows=197 width=49)
- Filter: (appid = 1)
- (2 rows)

SKY mobi
新手机 新应用 新娱乐

从柱状图中快速读取需要大运算量的TOP数据

- digoal=# explain select * from test_1 where appid>1000;
- QUERY PLAN
- ---
- Seq Scan on test_1 (cost=0.00..45619.00 rows=1800419 width=49)
- Filter: (appid > 1000)
- digoal=# select count(*) from test_1 where appid>1000;
- 1800263

- digoal=# explain select * from test_1 where appid>1000 and crt_time>now();
- QUERY PLAN
- ---
- Seq Scan on test_1 (cost=0.00..55619.00 rows=180 width=49)
- Filter: ((appid > 1000) AND (crt_time > now()))
- digoal=# select * from test_1 where appid>1000 and crt_time>now();
- (0 rows)

SKY mobi
新手机 新应用 新娱乐

从柱状图中快速读取需要大运算量的TOP数据

- 从柱状图信息中评估值的排行. 例如以下, appid第一位是准确的, 后面的值因为太均匀, 所以不准确.
- digoal=# insert into test_1 (info,appid,crt_time) select 'test',1,now() from generate_series(1,100000);
- INSERT 0 100000
- digoal=# select appid,count(*) from test_1 group by appid order by count(*) desc limit 5;
- appid | count
- -------+--------
- 1 | 101189
- 9853 | 253
- 6502 | 249
- 464 | 249
- 1688 | 249
- (5 rows)
- digoal=# select most_common_vals,most_common_freqs from pg_stats where tablename='test_1' and attname='appid';
- {1,2972,94,207,1998,....}
- {0.0474333,0.000366667,0.000333333,0.000333333,0.000333333,....}

SKY mobi
新手机 新应用 新娱乐

从柱状图中快速读取需要大运算量的TOP数据

- 对于数组类型的字段, 同样适用. 可以评估出现次数最频繁的元素.
- digoal=# create table test_2(id serial primary key, appid int[], crt_time timestamp);
- CREATE TABLE

- 假设appid为0-10的程序比较火爆, 模拟100秒插入请求.
- vi test.sql
- insert into test_2(appid) select array_agg(appid) appid_agg from (select round(10*random())::int as appid from generate_series(1,20)) t;
- pg93@db-172-16-3-33-> pgbench -M prepared -n -r -f ./test.sql -c 16 -j 4 -T 100 digoal

- 假设appid为10以上的程序不火爆, 模拟10秒插入请求.
- vi test.sql
- insert into test_2(appid) select array_agg(appid) appid_agg from (select round(1000*random())::int as appid from generate_series(1,20)) t;
- pg93@db-172-16-3-33-> pgbench -M prepared -n -r -f ./test.sql -c 16 -j 4 -T 10 digoal

SKY mobi
新手机 新应用 新娱乐

从柱状图中快速读取需要大运算量的TOP数据

- digoal=# select appid,count(*) from (select unnest(appid) as appid from test_2) t group by appid order by count(*) desc limit 20;
- appid | count
- -------+--------
- 9 | 872831
- 6 | 871908
- 3 | 871867
- 7 | 871551
- 8 | 871436
- 4 | 871391
- 1 | 871051
- 5 | 870770
- 2 | 870692
- 10 | 435583
- 0 | 435342
- 387 | 831
- 69 | 824
- 665 | 823

SKY mobi
新手机 新应用 新娱乐

从柱状图中快速读取需要大运算量的TOP数据

- digoal=# select * from
- (select row_number() over(partition by r) as rn,ele from (select unnest(most_common_elems::text::int[]) ele,2 as r from pg_stats where tablename='test_2' and attname='appid') t) t1
- join
- (select row_number() over(partition by r) as rn,freq from (select unnest(most_common_elem_freqs) freq,2 as r from pg_stats where tablename='test_2' and attname='appid') t) t2
- on (t1.rn=t2.rn) order by t2.freq desc limit 20;
- rn | ele | rn | freq
- -----+-----+-----+------------
- 2 | 1 | 2 | 0.810967
- 8 | 7 | 8 | 0.8102
- 3 | 2 | 3 | 0.809233
- 4 | 3 | 4 | 0.808433
- 10 | 9 | 10 | 0.808367
- 7 | 6 | 7 | 0.808067
- 5 | 4 | 5 | 0.807467
- 6 | 5 | 6 | 0.806667
- 9 | 8 | 9 | 0.806233
- 1 | 0 | 1 | 0.590833
- 11 | 10 | 11 | 0.588033
- 474 | 939 | 474 | 0.00233333
- 169 | 348 | 169 | 0.00233333

SKY mobi
新手机 新应用 新娱乐

使用hll数据类型快速统计唯一值和新增值

- http://blog.163.com/digoal@126/blog/static/163877040201312791787 6/
- 1. 快速的检索hll中存储的唯一值.
- select count(distinct userid) from access_log where date(crt_time)='2013-02-01'; -- 非常耗时.
- create table access_date (acc_date date unique, userids hll);
- insert into access_date select date(crt_time), hll_add_agg(hll_hash_integer(user_id)) from access_log group by 1;
- select #userids from access_date where acc_date='2013-02-01'; -- 这条语句返回只要1毫秒左右. (10亿个唯一值返回也在1毫秒左右)
- 而hll仅仅需要1.2KB就可以存储1.6e+12的唯一值.

- 2. 因为hll中实际上存储了键值信息, 所以还可以做类似新增用户等的统计.
- digoal=> create table access_date(acc_date date unique, userids hll);
- digoal=> insert into access_date select current_date, hll_add_agg(hll_hash_integer(user_id)) from generate_series(1,10000) t(user_id);
- digoal=> insert into access_date select current_date-1, hll_add_agg(hll_hash_integer(user_id)) from generate_series(5000,20000) t(user_id);
- digoal=> insert into access_date select current_date-2, hll_add_agg(hll_hash_integer(user_id)) from generate_series(9000,40000) t(user_id);

SKY mobi
新手机 新应用 新娱乐

使用hll数据类型快速统计唯一值和新增值

- digoal=> select *,total_users-coalesce(lag(total_users,1) over (order by rn),0) AS new_users from (
  digoal(> SELECT acc_date, row_number() over date as rn,#hll_union_agg(userids) OVER date as total_users
  digoal(> FROM access_date
  digoal(> WINDOW date AS (ORDER BY acc_date ASC ROWS UNBOUNDED PRECEDING)
  digoal(>) t;
  acc_date | rn | total_users | new_users
  ------------+----+-----------------+-----------------
  2013-02-25 | 1 | 30324.8563878223 | 30324.8563878223
  2013-02-26 | 2 | 33944.8370446358 | 3619.98065681347
  2013-02-27 | 3 | 38696.2201822711 | 4751.38313763532
  (3 rows)
  Time: 2.327 ms

SKY mobi
新手机 新应用 新娱乐

PostgreSQL参考资料

- Book
  - 《PostgreSQL Introduction and Concepts》
  - 《PostgreSQL 9 Administration Cookbook》
  - 《PostgreSQL 9.0 High Performance》
  - 《PostgreSQL Server Programming》
  - 《PostgreSQL开发必备参考手册》
  - http://www.postgresql.org/docs/
  - http://www.postgresql.org/docs/books/
- Web sites
  - 中文网站: http://bbs.pgsqldb.com http://www.postgres.cn
  - 代码树: http://doxygen.postgresql.org/
  - 代码提交集: https://commitfest.postgresql.org/
  - 项目GIT: http://git.postgresql.org
  - PostgreSQL GITHUB镜像: https://github.com/postgres/postgres
  - PostgreSQL JDBC 驱动: http://jdbc.postgresql.org
  - PostgreSQL ODBC 驱动: http://www.postgresql.org/ftp/odbc/versions/src/
  - PostgreSQL 扩展包: http://pgfoundry.org http://pgxn.org/
  - GUI工具(pgAdmin): http://www.pgadmin.org/
  - 安全漏洞: http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=postgresql
  - More: http://blog.163.com/digoal@126/blog/static/163877040201172183022203/

SKY mobi
新手机 新应用 新娱乐

PostgreSQL的安装

- PostgreSQL源码目录结构简介
- 源码安装PostgreSQL
- PostgreSQL 软件目录结构简介
- PostgreSQL 集群结构简介
- 扩展插件的安装

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码目录结构简介

- [root@db-172-16-3-150 postgresql-9.3.2]# ll
- total 2488
- -rw-r--r-- 1 1107 1107 385 Dec 3 04:57 aclocal.m4
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:10 config
- -rwxr-xr-x 1 1107 1107 888391 Dec 3 04:57 configure 源码安装的配置脚本
- -rw-r--r-- 1 1107 1107 65742 Dec 3 04:57 configure.in
- drwxrwxrwx 56 1107 1107 4096 Dec 3 05:09 contrib 已打包到PG源码中的第三方贡献的插件源码
- -rw-r--r-- 1 1107 1107 1192 Dec 3 04:57 COPYRIGHT
- drwxrwxrwx 3 1107 1107 4096 Dec 3 05:10 doc 文档
- -rw-r--r-- 1 1107 1107 3767 Dec 3 04:57 GNUmakefile.in
- -rw-r--r-- 1 1107 1107 1471819 Dec 3 05:12 HISTORY 版本变更的历史记录
- -rw-r--r-- 1 1107 1107 76689 Dec 3 05:12 INSTALL 安装说明
- -rw-r--r-- 1 1107 1107 1489 Dec 3 04:57 Makefile
- -rw-r--r-- 1 1107 1107 1284 Dec 3 04:57 README
- drwxrwxrwx 15 1107 1107 4096 Dec 3 05:12 src 源代码

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码目录结构简介

- http://www.postgresql.org/docs/9.3/static/contrib.html 第三方插件的详细介绍和使用手册
- [root@db-172-16-3-150 postgresql-9.3.2]# cd contrib/　第三方插件(大多数都非常实用)
- [root@db-172-16-3-150 contrib]# ll
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 adminpack -- 一些管理函数
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 auth_delay -- 认证失败后延迟报异常, 可以防止暴力破解
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 auto_explain -- 将超过指定执行时间的SQL的执行计划输出到日志中
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 btree_gin -- gin索引方法的btree操作符扩展. (在某些情况下"多列gin组合索引"比"多个btree单列索引"的bitmap anding更高效)
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:09 btree_gist -- gist索引方法的btree操作符扩展.(在组合索引中的某些列类型仅支持gist索引访问方法, 而另一些列的类型支持btree以及gist时btree_gist更为有效, 同时btree_gist还新增了<>用于排他约束,<->用于近邻算法)
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 chkpass -- 自动加密的字段类型(使用UNIX标准函数crypt()进行封装, 所以仅支持前8位安全) 'abcdefghijkl'::chkpass = 'abcdefgh'
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 citext -- 不区分大小写的数据类型.
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:12 cube -- 多维立方体类型, 支持多维立方体对象的相同,相交,包含等运算
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 dblink -- PostgreSQL跨库操作插件

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码目录结构简介

- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 dict_int -- 全文检索的一个字典模板, 用于控制数字被拆分的最大长度. 以控制数字在全文检索中的分词个数.(maxlen =6: 12345678 -> 123456 截断成6个, rejectlong =true则忽略这个分词)
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 dict_xsyn -- 全文检索的一个字典模块, 设置分词的同义词, 支持同义词匹配.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 dummy_seclabel -- 用于安全标签SQL的测试
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 earthdistance -- 可以使用cube或point类型计算地球表面两点之间的距离
- drwxrwxrwx 7 1107 1107 4096 Dec 3 05:09 file_fdw -- 文件外部表模块
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 fuzzystrmatch -- 单字节字符串之间的相似性判断
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:09 hstore -- hstore用于存储k-v数据类型, 同时这个插件还提供了比较多的K-V类型相关的函数和操作符. 例如提供数组,json,hstore之间的转换. k-v的存在判断,删除k-v值.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 intagg -- int类型的数组聚合函数.(内建的array_agg函数已包含这个功能)
- drwxrwxrwx 6 1107 1107 4096 Dec 3 05:09 intarray -- int类型的数组功能扩展库, 提供了一些常用的函数和操作符(数组元素个数, 元素排序, 元素下标, 取元素子集, 相交, 包含, 增加元素, 删除元素, 合并等)
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 isn -- 提供国际通用的产品标识码数据类型, 例如ISBN, ISMN...
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 lo -- 大对象的一个可选模块, lo类型以及自动unlink大对象的触发器. 方便大对象在消亡后的自动unlink, 防止大对象存储泄漏(类似内存泄漏).
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:09 ltree -- 异构数据类型以及操作函数和操作符. 例如China.Zhejiang.Hang <@ 'China'

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码目录结构简介

- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 oid2name -- oid转换成name的命令行工具, 不属于extension. 或通过系统表查询得到.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pageinspect -- 用于读取数据库PAGE裸信息的插件, 可以读main, fsm, vm FORK的页数据, 一般用于debug. (使用时请参照对应数据库版本的头文件解读信息)
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 passwordcheck -- 创建用户或者修改用户密码时, 检查密码的安全性, 如果太弱的话, 将返回错误.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_archivecleanup -- 清除归档文件的命令, 不属于extension.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pgbench -- 数据库性能测试的命令, 不属于extension.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_buffercache -- 输出当前的shared buffer的状态数据(细化到page number)
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 pgcrypto -- PostgreSQL的服务端数据加密的扩展库
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_freespacemap -- 输出对象指定page或所有page的free space map信息.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pgrowlocks -- (从行头信息中的infomask获取行锁信息), 注意输出的不是snapshot.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_standby -- 8.4以及以前的版本方便于创建warm standby的命令行工具
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_stat_statements -- 跟踪数据库的SQL, 收集SQL的统计信息.
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 pgstattuple -- 行级统计信息(dead tuples, live tuples, table_len, free_space, free_percent), 索引的统计信息.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码目录结构简介

- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_test_fsync -- 测试磁盘的fsync速率, 适用于选择最快的wal_sync_method
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_test_timing -- 测试系统定时器的开销, 开销越大, explain analyze时间结果越不准, 需要调整系统时钟源
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:09 pg_trgm -- 将字符串拆分成3个一组的多个单元, 用于测试两个字符串之间的近似度. 比分词更加暴力.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_upgrade -- 跨大版本的升级工具(例如9.0 -> 9.1)
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_upgrade_support -- pg_upgrade用到的服务端函数集
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 pg_xlogdump -- 从xlog中dump出一些易读的底层信息
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 postgres_fdw -- postgresql跨库的外部表插件
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:12 seg -- 线段类型和浮点数的区间类型. 以及相关的操作符, 索引访问方法等
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 sepgsql -- 基于SELinux安全策略的访问控制模块.
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 spi -- 一些服务端的触发器函数(例如跟踪记录的存活时间, 被哪个用户修改了, 记录的修改时间等)
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 sslinfo -- 输出ssl认证的客户端的一些认证信息
- drwxrwxrwx 3 1107 1107 4096 Dec 3 05:09 start-scripts -- 数据库启动脚本模板
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:09 tablefunc -- 一般可用于行列变换, 异构数据处理等.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码目录结构简介

- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 tcn -- 提供异步消息输出的触发器.
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 test_parser -- 全文检索中的一个自定义parser的测试插件.
- drwxrwxrwx 5 1107 1107 4096 Dec 3 05:09 tsearch2 -- 全文检索相关的插件, 在全文检索未引入PG内核前的PG版本可以使用这个插件来实现全文检索功能, >=8.3以后就不需要这个了.
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 unaccent -- 全文检索相关的插件
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 uuid-ossp -- 生成UUID的插件
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 vacuumlo -- 大对象垃圾回收的命令
- drwxrwxrwx 2 1107 1107 4096 Dec 3 05:09 worker_spi -- 9.3新增的服务端worker编程范例
- drwxrwxrwx 4 1107 1107 4096 Dec 3 05:09 xml2 -- xml相关插件

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码目录结构简介

- src目录结构 (截取部分)
- pg93@db-172-16-3-150-> ll src
- drwxrwxrwx 25 1107 1107 4.0K Dec 3 05:10 backend -- 后台进程相关的源码(包括postmaster, optimizer, replication, checkpoint, access,... 等等)
- drwxrwxrwx 12 1107 1107 4.0K Dec 3 05:10 bin -- 客户端进程的源码(例如initdb, psql, pg_dump等等)
- drwxrwxrwx 27 1107 1107 4.0K Dec 3 05:12 include -- 头文件
- drwxrwxrwx 4 1107 1107 4.0K Dec 3 05:10 interfaces -- 交互接口的源码(例如ecpg, libpq)
- drwxrwxrwx 6 1107 1107 4.0K Dec 3 05:10 pl -- 过程语言的源码(例如plperl, plpgsql, plpython, pltcl)
- drwxrwxrwx 2 1107 1107 4.0K Dec 3 05:09 template -- 不同OS平台下的编译器选项模板
- drwxrwxrwx 9 1107 1107 4.0K Dec 3 05:09 test -- 测试相关的源码
- drwxrwxrwx 4 1107 1107 4.0K Dec 3 05:10 timezone -- (从http://www.iana.org/time-zones同步的时区库)
- drwxrwxrwx 10 1107 1107 4.0K Dec 3 05:10 tools -- 编译过程中用到的一些命令行工具
- drwxrwxrwx 2 1107 1107 4.0K Dec 3 05:10 tutorial -- 包含基本的SQL教程脚本

SKY mobi
新手机 新应用 新娱乐

源码安装PostgreSQL

- 详细说明参考INSTALL文件
- 简明安装步骤
- ./configure --prefix=/opt/pgsql9.3.2 --with-pgport=1921 --with-perl --with-tcl --with-python --with-openssl --with-pam --without-ldap --with-libxml --with-libxslt --enable-thread-safety --with-wal-blocksize=16 --with-blocksize=16 --enable-dtrace --enable-debug
- gmake world
- gmake check-world -- (这个需要使用普通用户执行. 可选, 耗时较长)
- gmake install-world

- 如果遇到依赖的动态库缺失, 需要提前安装即可.
- gmake world安装包含了文档,所有的contrib.

SKY mobi
新手机 新应用 新娱乐

源码安装PostgreSQL

- 软件安装好后, 在操作系统中创建一个普通用户, 用于初始化数据库, 开启和关闭数据库.
- 首先把安装好的软件链接到一个常用的目录
- ln -s /opt/pgsql9.3.2 /opt/pgsql
- 创建用户, 并修改它的profile
- useradd postgres
- su - postgres
- vi ~/.bash_profile
- # add
- export PGPORT=1921
- export PGDATA=/pgdata/pg_root
- export LANG=en_US.utf8
- export PGHOME=/home/pg93/pgsql
- export LD_LIBRARY_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/lib64:/lib:/usr/lib:/usr/local/lib:$LD_LIBRARY_PATH

SKY mobi
新手机 新应用 新娱乐

源码安装PostgreSQL

- export DATE=`date +"%Y%m%d%H%M"`
- export PATH=$PGHOME/bin:$PATH:.
- export MANPATH=$PGHOME/share/man:$MANPATH
- export PGUSER=postgres
- export PGHOST=$PGDATA
- alias rm='rm -i'
- alias ll='ls -lh'
- export PGDATABASE=digoal

- 测试一下
- su - postgres
- $ psql -V
- psql (PostgreSQL) 9.3.2

SKY mobi
新手机 新应用 新娱乐

源码安装PostgreSQL

- 创建数据库集群
- 首先要创建数据库集群的目录
- mkdir -p /pgdata/pg_root
- chown -R postgres:postgres /pgdata/pg_root
- 初始化集群
- su - postgres
- initdb -D $PGDATA -E UTF8 --locale=C -U postgres -W

- 启动数据库前修改一下pg_hba.conf和postgresql.conf
- pg_hba.conf用于配置控制访问数据库的来源
- postgresql.conf是数据库的主配置文件, 最好也调整一下Linux内核参数.

- 启动数据库
- pg_ctl start -D $PGDATA
- 停库
- pg_ctl stop -m fast|smart|immediate -D $PGDATA

SKY mobi
新手机 新应用 新娱乐

源码安装PostgreSQL

- Linux内核相关参数调整
- vi /etc/sysctl.conf

- kernel.shmmax = 68719476736 (默认)
- kernel.shmall = 4294967296 (默认)
- kernel.shmmni = 4096
- kernel.sem = 50100 64128000 50100 1280
- fs.file-max = 7672460
- net.ipv4.ip_local_port_range = 9000 65000
- net.core.rmem_default = 1048576
- net.core.rmem_max = 4194304
- net.core.wmem_default = 262144
- net.core.wmem_max = 1048576

SKY mobi
新手机 新应用 新娱乐

源码安装PostgreSQL

- vi /etc/security/limits.conf

- * soft nofile 131072
- * hard nofile 131072
- * soft nproc 131072
- * hard nproc 131072
- * soft core unlimited
- * hard core unlimited
- * soft memlock 50000000
- * hard memlock 50000000

- vi /etc/sysconfig/iptables

- -A RH-Firewall-1-INPUT -i lo -j ACCEPT
- # 允许源IP
- -A RH-Firewall-1-INPUT -s 192.168.0.0/16 -j ACCEPT
- # 允许源IP访问目标端口
- # -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -m tcp -p tcp --dport 1921 -j ACCEPT
- # 允许任意IP访问目标端口
- # -A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m tcp --dport 5432 -j ACCEPT

SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库结构介绍

- ll $PGDATA
- drwx------. 12 pg93 pg93 4.0K Nov 22 16:41 base -- 这个目录对应pg_default表空间
- drwx------. 2 pg93 pg93 4.0K Nov 26 22:02 global --这个目录对应pg_global表空间, 存放集群中的共享对象例pg_database表, (select relkind,relname from pg_class where reltablespace = (select oid from pg_tablespace where spcname='pg_global') order by 1) , 包含控制文件等.
- drwx------. 2 pg93 pg93 4.0K Nov 21 13:50 pg_clog -- 存储事务提交状态数据
- -rw------- 1 pg93 pg93 4.6K Nov 22 17:22 pg_hba.conf -- 数据库访问控制文件
- drwx------. 2 pg93 pg93 20K Dec 7 00:00 pg_log -- 数据库日志目录(根据配置定义, 可能没有这个目录)
- drwx------. 4 pg93 pg93 4.0K Sep 27 08:10 pg_multixact -- 共享行锁的事务状态数据
- drwx------. 2 pg93 pg93 4.0K Nov 26 22:01 pg_notify -- 异步消息相关的状态数据
- drwx------. 2 pg93 pg93 4.0K Sep 27 08:10 pg_serial -- 串行隔离级别的事务状态数据
- drwx------. 2 pg93 pg93 4.0K Sep 27 08:10 pg_snapshots -- 存储执行了事务snapshot导出的状态数据
- drwx------. 2 pg93 pg93 4.0K Dec 7 15:57 pg_stat_tmp -- 统计信息的临时文件
- drwx------. 2 pg93 pg93 4.0K Nov 22 14:16 pg_subtrans -- 子事务状态数据

SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库结构介绍

- drwx------. 2 pg93 pg93 4.0K Oct 28 09:16 pg_tblspc -- 表空间的软链接目录
- drwx------. 2 pg93 pg93 4.0K Sep 27 08:10 pg_twophase -- 二阶事务的状态数据
- -rw-------. 1 pg93 pg93 4 Sep 27 08:10 PG_VERSION -- 数据库版本
- lrwxrwxrwx 1 pg93 pg93 18 Nov 15 11:17 pg_xlog -> /ssd1/pg93/pg_xlog -- 存储WAL文件
- -rw------- 1 pg93 pg93 20K Nov 26 13:50 postgresql.conf -- 配置文件
- -rw-------. 1 pg93 pg93 35 Nov 26 22:01 postmaster.opts -- 记录数据库启动时的命令行选项
- -rw------- 1 pg93 pg93 71 Nov 26 22:01 postmaster.pid -- 数据库启动的主进程信息文件(包括$PGDATA目录, 数据库启动时间, 监听端口ipc信息等)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL源码包中的插件安装

- 如果编译数据库时使用了gmake world和gmake install-world, 所有的插件都会被安装, 那么 就不需要再次安装了.
- 插件目录
- contrib

- 进入要安装的插件目录, 例如
- cd contrib/pg_freespacemap
- 把pg_config命令加入到默认路径中
- export PATH=/opt/pgsql/bin:$PATH
- gmake clean
- gmake
- gmake install -- 这一步会将生成的动态库文件拷贝到pgsql的lib目录, 同时拷贝生成的扩展文件到$PGHOME/share/extension中.
- 安装完后, 到数据库中以超级用户执行create extension
- create extension pg_freespacemap;

SKY mobi
新手机 新应用 新娱乐

PostgreSQL第三方插件安装

- 根据第三方插件提供的安装说明进行安装
- 通用的安装方法
- 1. 把第三方插件的源码目录拷贝到contrib目录中
- 2. 把pg_config加入到PATH中
- 3. gmake clean; gmake; gmake install
- 4. create extension xxx; 对于一些老版本的插件, 可能没有改成扩展插件的安装模式. 那么需要到对应的库中直接执行SQL.

SKY mobi
新手机 新应用 新娱乐

练习

- 源码安装PostgreSQL
- 内部插件的安装和使用
- 第三方插件的安装和使用
  - 可到git.postgresql.org下载第三方插件源码
- PostgreSQL 特性的测试可以参照PostgreSQL源码中提供的测试脚本进行测试
  - src/test/regress/sql

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 体系结构

- 体系结构
- 目标:
- 了解系统表以及系统表之间的关系, 系统视图, 管理函数等
- 了解PG进程结构
- 了解PG物理结构

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统表介绍

- 系统表, 系统表之间基本上都是以oid关联. 例如pg_attrdef.adrelid 关联 pg_class.oid
- select relkind,relname from pg_class where relnamespace = (select oid from pg_namespace where nspname='pg_catalog') and relkind='r' order by 1,2;
- r | pg_aggregate -- 聚合函数信息, 包括聚合函数的中间函数, 中间函数的初始值, 最终函数等.
- r | pg_am -- 系统支持的索引访问方法. (如btree, hash, gist, gin , spgist)
- r | pg_amop -- 存储每个索引访问方法操作符家族(pg_opfamily)中的详细操作符信息
- r | pg_amproc -- 存储每个索引访问方法操作符家族(pg_opfamily)支持的函数信息.
- r | pg_attrdef -- 存储数据表列的默认值(例如创建表时指定了列的default值).
- r | pg_attribute -- 存储数据表列的详细信息. 包括隐含的列(ctid, cmin, cmax, xmin, xmax)
- r | pg_auth_members -- 数据库用户的成员关系信息.
- r | pg_authid -- 存储数据库用户的详细信息(包括是否超级用户, 是否允许登陆, 密码(加密与否和创建用户时是否指定encrypted有关), 密码失效时间等)
- r | pg_cast -- 数据库的显性类型转换路径信息, 包括内建的和自定义的.
- r | pg_class -- 几乎包括了数据库的所有对象信息(r = ordinary table, i = index, S = sequence, v = view, m = materialized view, c = composite type, t = TOAST table, f = foreign table)
- r | pg_collation -- 集信息, 包括encoding, collate, ctype等.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统表介绍

- r | pg_constraint -- 存储列上定义的约束信息(例如PK, FK, UK, 排他约束, check约束, 但是不包括非空约束)
- r | pg_conversion -- 字符集之间的转换的相关信息
- r | pg_database -- 集群中的数据库信息
- r | pg_db_role_setting -- 基于角色和数据库组合的定制参数信息. (alter role set ...)
- r | pg_default_acl -- 存储新建对象的初始权限信息
- r | pg_depend -- 数据库对象之间的依赖信息.
- r | pg_description -- 数据库对象的描述信息
- r | pg_enum -- 枚举类型信息
- r | pg_event_trigger -- 事件触发器信息
- r | pg_extension -- 扩展插件信息
- r | pg_foreign_data_wrapper -- FDW信息
- r | pg_foreign_server -- 外部服务器信息
- r | pg_foreign_table -- 外部表信息.
- r | pg_index -- 索引信息

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统表介绍

- r | pg_inherits -- 继承表的继承关系信息
- r | pg_language -- 过程语言信息
- r | pg_largeobject -- 大对象的切片后的真实数据存储在这个表里
- r | pg_largeobject_metadata -- 大对象的元信息, 包括大对象的owner, 访问权限.
- r | pg_namespace -- 数据库中的schema信息(pg中称为namespace)
- r | pg_opclass -- 索引访问方法的操作符分类信息.
- r | pg_operator -- 操作符信息
- r | pg_opfamily -- 操作符家族信息
- r | pg_pltemplate -- 过程语言的模板信息
- r | pg_proc -- 数据库服务端函数信息
- r | pg_range -- 范围类型信息
- r | pg_rewrite -- 表和视图的重写规则信息

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统表介绍

- r | pg_seclabel -- 安全标签信息(SELinux)
- r | pg_shdepend -- 数据库中的对象之间或者集群中的共享对象之间的依赖关系
- r | pg_shdescription -- 共享对象的描述信息
- r | pg_shseclabel -- 共享对象的安全标签信息(SELinux)
- r | pg_statistic -- analyze生成的统计信息, 用于查询计划器计算成本.
- r | pg_tablespace -- 表空间相关的信息.
- r | pg_trigger -- 表上的触发器信息
- r | pg_ts_config -- 全文检索的配置信息
- r | pg_ts_config_map -- 全文检索配置映射信息
- r | pg_ts_dict -- 全文检索字典信息
- r | pg_ts_parser -- 全文检索解析器信息
- r | pg_ts_template -- 全文检索模板信息
- r | pg_type -- 数据库中的类型信息
- r | pg_user_mapping -- foreign server的用户配置信息.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统视图介绍

- 系统视图
- select relkind,relname from pg_class where relnamespace = (select oid from pg_namespace where nspname='pg_catalog') and relkind='v' order by 1,2;
- v | pg_available_extension_versions -- 显示当前系统已经编译的扩展插件的版本信息
- v | pg_available_extensions --显示当前系统已经编译的扩展插件信息
- v | pg_cursors -- 当前可用的游标
- v | pg_group -- 用户组信息
- v | pg_indexes -- 索引信息
- v | pg_locks -- 锁信息
- v | pg_matviews -- 物化视图信息
- v | pg_prepared_statements -- 当前会话中使用prepare语法写的预处理SQL信息.
- v | pg_prepared_xacts -- 二阶事务信息
- v | pg_roles -- 数据库角色信息
- v | pg_rules -- 数据库中使用create rule创建的规则信息.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统视图介绍

- v | pg_seclabels -- 安全标签信息
- v | pg_settings -- 当前数据库集群的参数设置信息
- v | pg_shadow -- 数据库用户信息
- v | pg_stat_activity -- 会话活动信息
- v | pg_stat_all_indexes -- 查询用户权限范围内的所有索引的统计信息
- v | pg_stat_all_tables -- 查询用户权限范围内的所有表的统计信息
- v | pg_stat_bgwriter -- bgwriter进程的统计信息
- v | pg_stat_database -- 数据库级别的统计信息
- v | pg_stat_database_conflicts -- 数据库
- v | pg_stat_replication -- 流复制相关的统计信息
- v | pg_stat_sys_indexes -- 系统表相关的索引统计信息
- v | pg_stat_sys_tables -- 系统表统计信息
- v | pg_stat_user_functions -- 用户函数统计信息
- v | pg_stat_user_indexes -- 用户表的索引相关的统计信息
- v | pg_stat_user_tables -- 用户表统计信息

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统视图介绍

- v | pg_stat_xact_all_tables -- 当前事务的表级统计信息, 显示用户可以访问的所有表
- v | pg_stat_xact_sys_tables -- 当前事务的表级统计信息, 仅显示系统表
- v | pg_stat_xact_user_functions -- 当前事务的用户函数的统计信息
- v | pg_stat_xact_user_tables -- 当前事务的用户表的统计信息
- v | pg_statio_all_indexes -- io相关的统计信息
- v | pg_statio_all_sequences
- v | pg_statio_all_tables
- v | pg_statio_sys_indexes
- v | pg_statio_sys_sequences
- v | pg_statio_sys_tables
- v | pg_statio_user_indexes

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 系统视图介绍

- v | pg_statio_user_sequences
- v | pg_statio_user_tables
- v | pg_stats -- 数据库中的统计信息, 以列为最小统计单位输出
- v | pg_tables -- 数据库中的表对象的信息
- v | pg_timezone_abbrevs -- 时区缩写信息
- v | pg_timezone_names -- 时区信息, 包含全名
- v | pg_user -- 用户信息
- v | pg_user_mappings -- 外部表的用户映射权限信息.
- v | pg_views -- 视图信息

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- http://www.postgresql.org/docs/9.3/static/functions-admin.html
- 配置函数

Name	Return Type	Description
current_setting(setting_name)	text	get current value of setting
set_config(setting_name, new_value, is_local)	text	set parameter and return new value

- 服务端信号发送函数

Name	Return Type	Description
pg_cancel_backend(pidint)	boolean	Cancel a backend's current query. You can execute this against another backend that has exactly the same role as the user calling the function. In all other cases, you must be a superuser.
pg_reload_conf()	boolean	Cause server processes to reload their configuration files
pg_rotate_logfile()	boolean	Rotate server's log file
pg_terminate_backend(pidint)	boolean	Terminate a backend. You can execute this against another backend that has exactly the same role as the user calling the function. In all other cases, you must be a superuser.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- 备份控制函数

Name	Return Type	Description
pg_create_restore_point(name text)	text	Create a named point for performing restore (restricted to superusers)
pg_current_xlog_insert_location()	text	Get current transaction log insert location
pg_current_xlog_location()	text	Get current transaction log write location
pg_start_backup(label text [, fast boolean])	text	Prepare for performing on-line backup (restricted to superusers or replication roles)
pg_stop_backup()	text	Finish performing on-line backup (restricted to superusers or replication roles)
pg_is_in_backup()	bool	True if an on-line exclusive backup is still in progress.
pg_backup_start_time()	timestamp with time zone	Get start time of an on-line exclusive backup in progress.
pg_switch_xlog()	text	Force switch to a new transaction log file (restricted to superusers)
pg_xlogfile_name(location text)	text	Convert transaction log location string to file name
pg_xlogfile_name_offset(location text)	text, integer	Convert transaction log location string to file name and decimal byte offset within file
pg_xlog_location_diff(location text, location text)	numeric	Calculate the difference between two transaction log locations

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- 恢复控制函数

Name	Return Type	Description
pg_is_in_recovery()	bool	True if recovery is still in progress.
pg_last_xlog_receive_location()	text	Get last transaction log location received and synced to disk by streaming replication. While streaming replication is in progress this will increase monotonically. If recovery has completed this will remain static at the value of the last WAL record received and synced to disk during recovery. If streaming replication is disabled, or if it has not yet started, the function returns NULL.
pg_last_xlog_replay_location()	text	Get last transaction log location replayed during recovery. If recovery is still in progress this will increase monotonically. If recovery has completed then this value will remain static at the value of the last WAL record applied during that recovery. When the server has been started normally without recovery the function returns NULL.
pg_last_xact_replay_timestamp()	timestamp with time zone	Get time stamp of last transaction replayed during recovery. This is the time at which the commit or abort WAL record for that transaction was generated on the primary. If no transactions have been replayed during recovery, this function returns NULL. Otherwise, if recovery is still in progress this will increase monotonically. If recovery has completed then this value will remain static at the value of the last transaction applied during that recovery. When the server has been started normally without recovery the function returns NULL.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- 恢复控制函数

Name	Return Type	Description
pg_is_xlog_replay_paused()	bool	True if recovery is paused.
pg_xlog_replay_pause()	void	Pauses recovery immediately.
pg_xlog_replay_resume()	void	Restarts recovery if it was paused.

- 事务镜像导出函数

Name	Return Type	Description
pg_export_snapshot()	text	Save the current snapshot and return its identifier

例如用于并行逻辑备份, (举例讲解)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- 数据库对象管理函数

Name	Return Type	Description
pg_column_size(any)	int	Number of bytes used to store a particular value (possibly compressed)
pg_database_size(oid)	bigint	Disk space used by the database with the specified OID
pg_database_size(name)	bigint	Disk space used by the database with the specified name
pg_indexes_size(regclass)	bigint	Total disk space used by indexes attached to the specified table
pg_relation_size(relation regclass, fork text)	bigint	Disk space used by the specified fork ('main', 'fsm' or 'vm') of the specified table or index
pg_relation_size(relation regclass)	bigint	Shorthand for pg_relation_size(..., 'main')
pg_size_pretty(bigint)	text	Converts a size in bytes expressed as a 64-bit integer into a human-readable format with size units
pg_size_pretty(numeric)	text	Converts a size in bytes expressed as a numeric value into a human-readable format with size units
pg_table_size(regclass)	bigint	Disk space used by the specified table, excluding indexes (but including TOAST, free space map, and visibility map)
pg_tablespace_size(oid)	bigint	Disk space used by the tablespace with the specified OID
pg_tablespace_size(name)	bigint	Disk space used by the tablespace with the specified name
pg_total_relation_size(regclass)	bigint	Total disk space used by the specified table, including all indexes and TOAST data

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- 数据库对象存储位置管理函数

Name	Return Type	Description
pg_relation_filenode(relation regclass)	oid	Filenode number of the specified relation
pg_relation_filepath(relation regclass)	text	File path name of the specified relation

- 文件访问函数

Name	Return Type	Description
pg_ls_dir(dirname text)	setof text	List the contents of a directory
pg_read_file(filename text [, offset bigint, length bigint])	text	Return the contents of a text file
pg_read_binary_file(filename text [, offset bigint, length bigint])	bytea	Return the contents of a file
pg_stat_file(filename text)	record	Return information about a file

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- 应用锁函数, 对于长时间持锁的应用非常有效. 因为长时间的数据库重量锁会带来垃圾回收的问题. (举例讲解)

Name	Return Type	Description
pg_advisory_lock(key bigint)	void	Obtain exclusive session level advisory lock
pg_advisory_lock(key1 int, key2 int)	void	Obtain exclusive session level advisory lock
pg_advisory_lock_shared(key bigint)	void	Obtain shared session level advisory lock
pg_advisory_lock_shared(key1 int, key2 int)	void	Obtain shared session level advisory lock
pg_advisory_unlock(key bigint)	boolean	Release an exclusive session level advisory lock
pg_advisory_unlock(key1 int, key2 int)	boolean	Release an exclusive session level advisory lock
pg_advisory_unlock_all()	void	Release all session level advisory locks held by the current session
pg_advisory_unlock_shared(key bigint)	boolean	Release a shared session level advisory lock
pg_advisory_unlock_shared(key1 int, key2 int)	boolean	Release a shared session level advisory lock
pg_advisory_xact_lock(key bigint)	void	Obtain exclusive transaction level advisory lock
pg_advisory_xact_lock(key1 int, key2 int)	void	Obtain exclusive transaction level advisory lock
pg_advisory_xact_lock_shared(key bigint)	void	Obtain shared transaction level advisory lock

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 管理函数

- 应用锁函数

Name	Return Type	Description
pg_advisory_xact_lock_shared(key1 int, key2 int)	void	Obtain shared transaction level advisory lock
pg_try_advisory_lock(key bigint)	boolean	Obtain exclusive session level advisory lock if available
pg_try_advisory_lock(key1 int, key2 int)	boolean	Obtain exclusive session level advisory lock if available
pg_try_advisory_lock_shared(key bigint)	boolean	Obtain shared session level advisory lock if available
pg_try_advisory_lock_shared(key1 int, key2 int)	boolean	Obtain shared session level advisory lock if available
pg_try_advisory_xact_lock(key bigint)	boolean	Obtain exclusive transaction level advisory lock if available
pg_try_advisory_xact_lock(key1 int, key2 int)	boolean	Obtain exclusive transaction level advisory lock if available
pg_try_advisory_xact_lock_shared(key bigint)	boolean	Obtain shared transaction level advisory lock if available
pg_try_advisory_xact_lock_shared(key1 int, key2 int)	boolean	Obtain shared transaction level advisory lock if available

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 进程结构

- 进程源码大部分在 : src/backend/postmaster

- postmaster -- 所有数据库进程的主进程(负责监听和fork子进程)
- startup -- 主要用于数据库恢复的进程
- syslogger -- 记录系统日志
- pgstat -- 收集统计信息
- pgarch -- 如果开启了归档, 那么postmaster会fork一个归档进程.
- checkpointer -- 负责检查点的进程
- bgwriter -- 负责把shared buffer中的脏数据写入磁盘的进程
- autovacuum lanucher -- 负责回收垃圾数据的进程, 如果开启了autovacuum的话,那么postmaster会fork这个进程.
- autovacuum worker -- 负责回收垃圾数据的worker进程, 是lanucher进程fork出来的.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

- 主要讲一下数据存储结构
- 第一个问题 : 对象对应的物理文件在哪里?
- digoal=# select pg_relation_filepath('pg_class'::regclass);
- pg_relation_filepath
- ----------
- pg_tblspc/66422/PG_9.3_201306121/16384/12658
- (1 row)
- 分解 :
- 1. pg_tblspc/66422/PG_9.3_201306121/16384/12658
  - 代表$PGDATA中的相对路径
- 2. 66422
  - 这个对应表空间oid
  - digoal=# select spcname from pg_tablespace where oid=66422;
  - spcname
  - ----------
  - tbs_digoal
  - (1 row)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

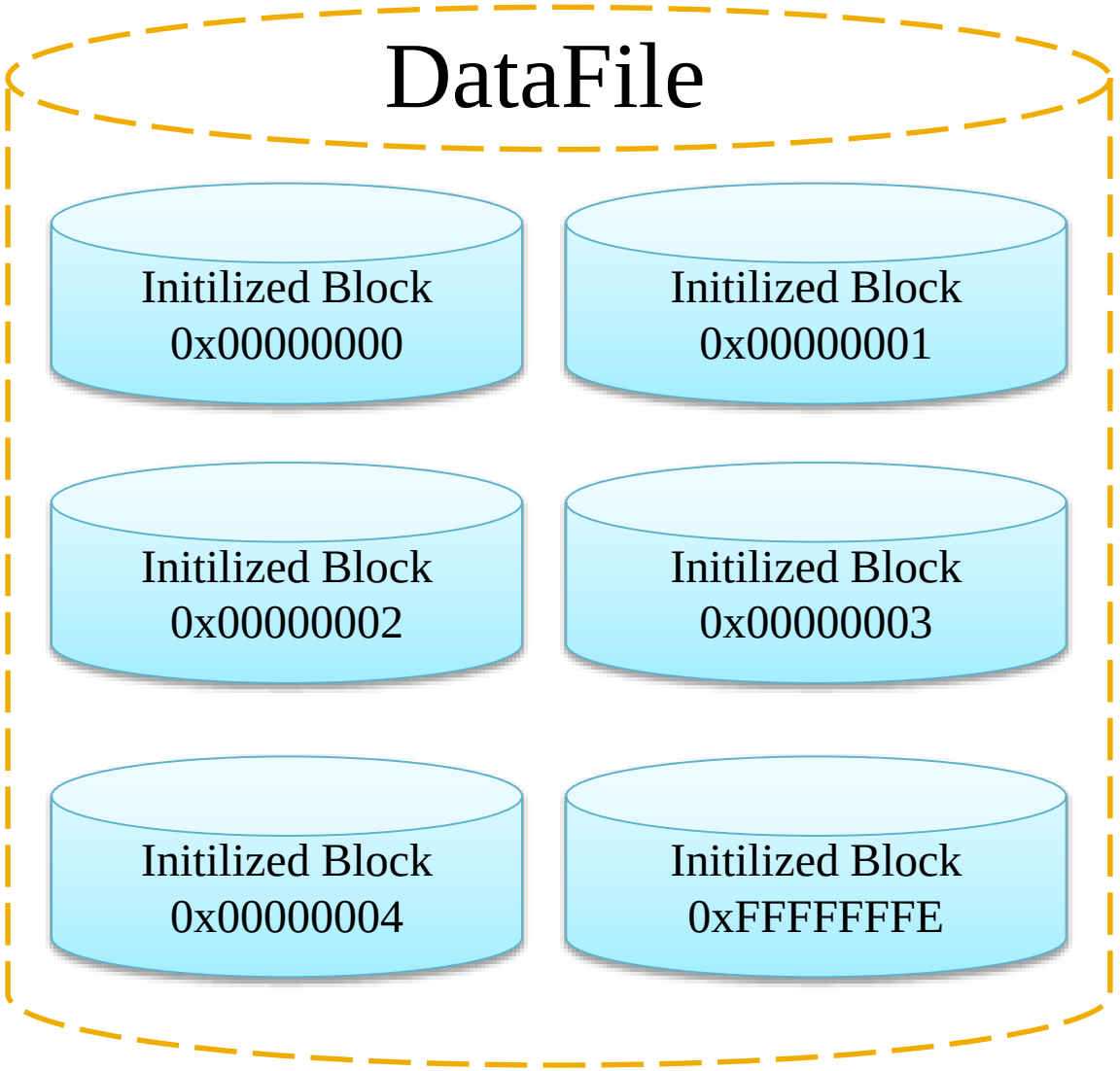
- 3. 16384
  - 对应数据库的oid
  - digoal=# select oid from pg_database where datname=current_database();
  - oid
  - -------
  - 16384
  - (1 row)
- 4. 12658
  - 对应表的main fork文件名, 其他fork后面加后缀fsm, vm, init等. 对于超过1GB(系统编译时指定), 文件名后面加 .x

- pg93@db-172-16-3-150-> ll $PGDATA/pg_tblspc/66422/PG_9.3_201306121/16384/12658*
- -rw------- 1 pg93 pg93 80K Dec 7 21:20 /ssd2/pg93/pg_root/pg_tblspc/66422/PG_9.3_201306121/16384/12658
- -rw------- 1 pg93 pg93 24K Nov 27 16:11 /ssd2/pg93/pg_root/pg_tblspc/66422/PG_9.3_201306121/16384/12658_fsm
- -rw------- 1 pg93 pg93 8.0K Nov 21 08:19 /ssd2/pg93/pg_root/pg_tblspc/66422/PG_9.3_201306121/16384/12658_vm

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

- 数据文件的结构



One DataFile(s) Per Table or Index .
BlockID :
sequentially, 0 to 0xFFFFFFFE

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

单个BLOCK的结构

PageHeaderData(24 Bytes)

ItemIdData(Array of (offset,length) pairs
pointing to the actual items. 4 bytes per item)

Free space(The unallocated space.
New item pointers are allocated from the start of this area,
new items from the end.)

Items (The actual items themselves.)

Special space (Index access method specific data.
Different methods store different data. Empty
in ordinary tables.)(an access method should always
initialize its pages with PageInit
and then set its own opaque fields.)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

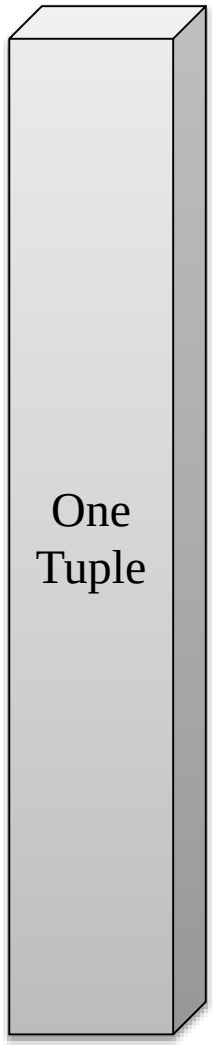
- BLOCK头数据结构

Field	Type	Length	Description
pd_lsn	XLogRecPtr	8 bytes	LSN: next byte after last byte of xlog record for last change to this page
pd_tli	uint16	2 bytes	TimeLineID of last change (only its lowest 16 bits)
pd_flags	uint16	2 bytes	Flag bits
pd_lower	LocationIndex	2 bytes	Offset to start of free space
pd_upper	LocationIndex	2 bytes	Offset to end of free space
pd_special	LocationIndex	2 bytes	Offset to start of special space
pd_pagesize_version	uint16	2 bytes	Page size and layout version number information
pd_prune_xid	TransactionId	4 bytes	Oldest unpruned XMAX on page, or zero if none

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

- TUPLE数据结构



HeapTupleHeaderData(23 Bytes on most machine)

null bitmap(Optional,The null bitmap is only present
if the *HEAP_HASNULL* bit is set in t_infomask.
occupies enough bytes to have one bit per data column.
a 1 bit indicates not-null, a 0 bit is a null)

Padding(Optional, Any padding needed to make t_hoff a MAXALIGN
multiple will appear between the null bitmap and the object ID)

object ID(Optional, only present if the
HEAP_HASOID bit is set in t_infomask)

ColumnData

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

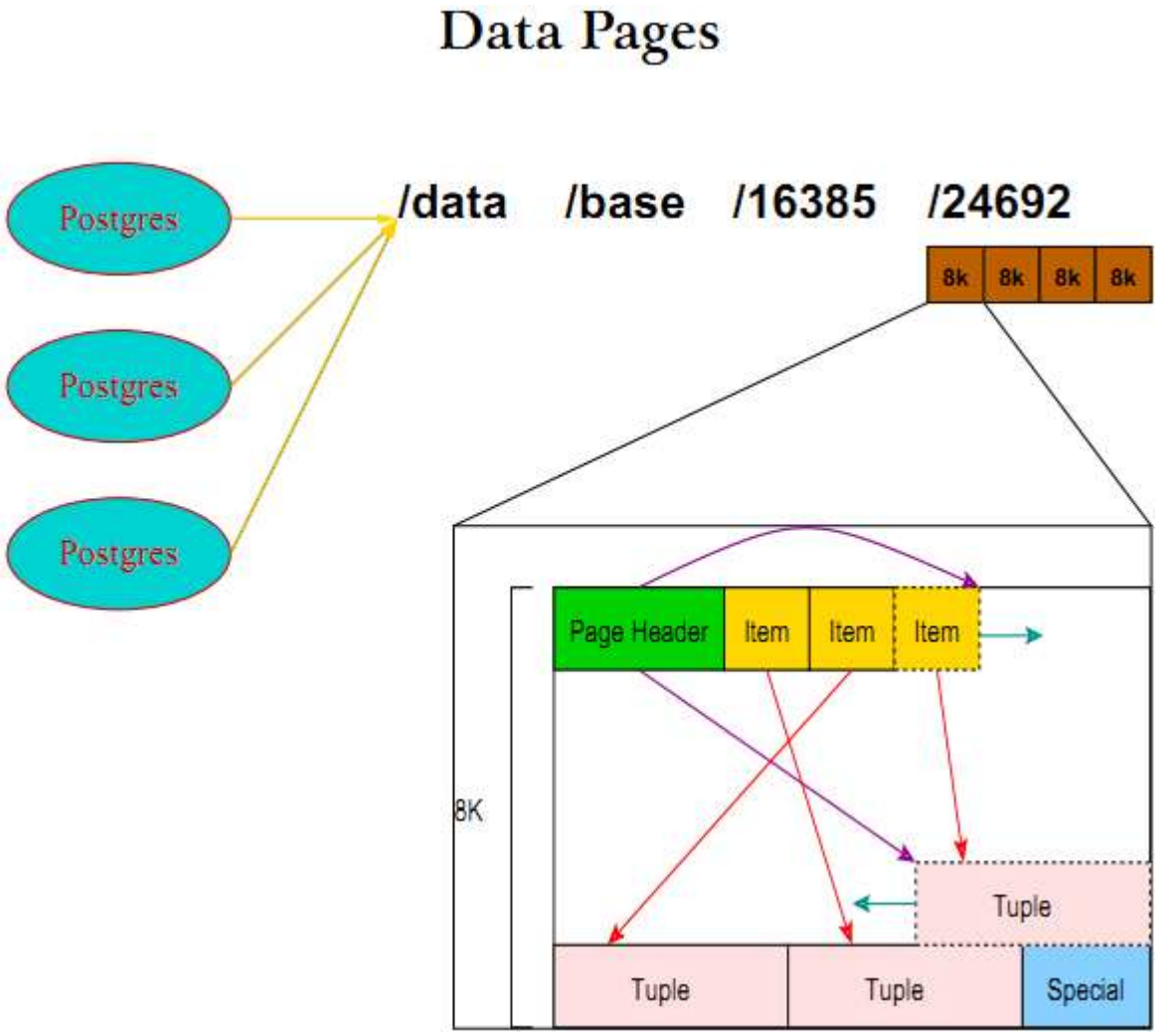
- TUPLE头部数据结构

Field	Type	Length	Description
t_xmin	TransactionId	4 bytes	insert XID stamp
t_xmax	TransactionId	4 bytes	delete XID stamp
t_cid	CommandId	4 bytes	insert and/or delete CID stamp (overlays with t_xvac)
t_xvac	TransactionId	4 bytes	XID for VACUUM operation moving a row version
t_ctid	ItemPointerData	6 bytes	current TID of this or newer row version
t_infomask2	int16	2 bytes	number of attributes, plus various flag bits
t_infomask	uint16	2 bytes	various flag bits
t_hoff	uint8	1 byte	offset to user data

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 物理结构

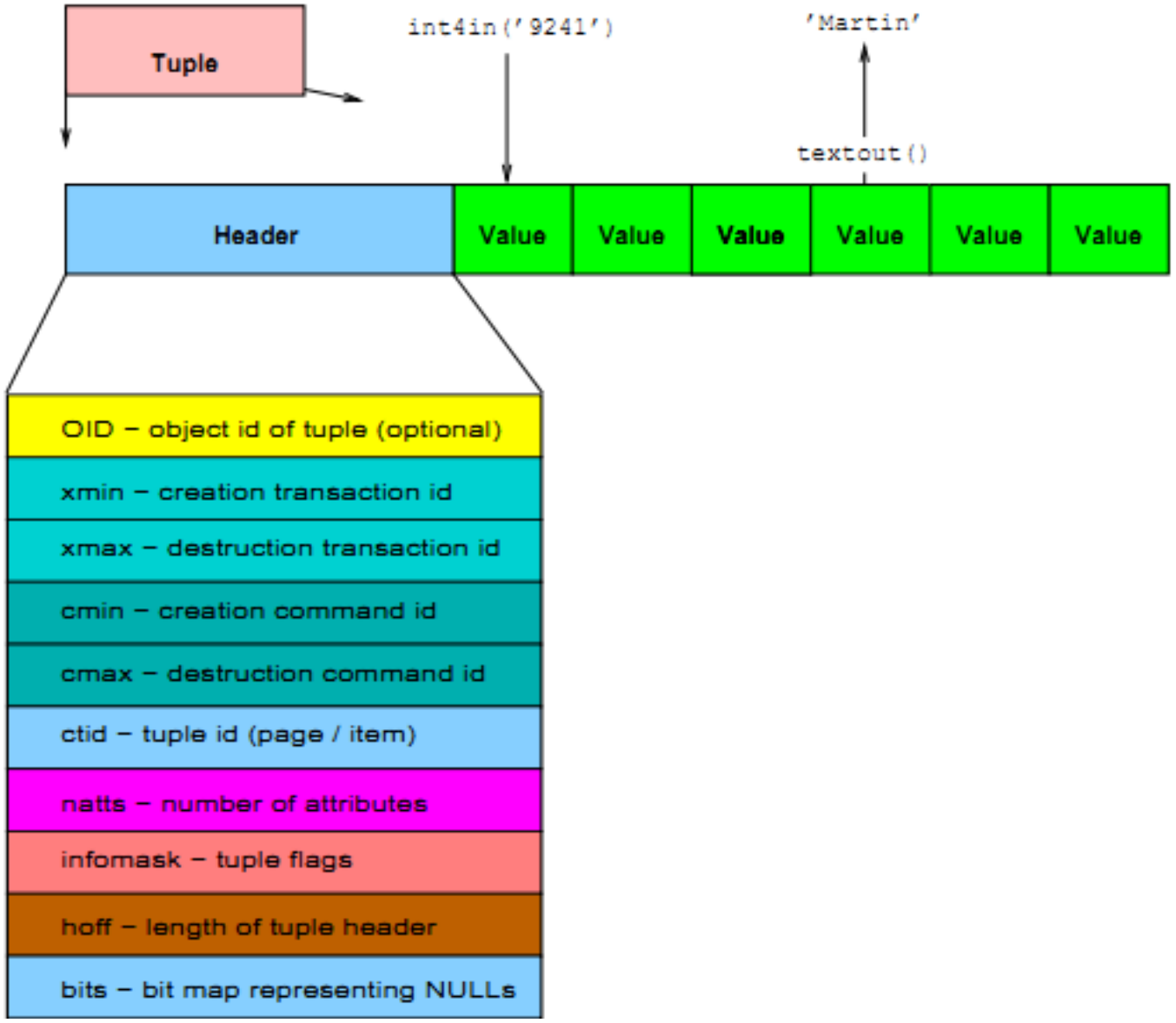
- 图例
- 使用pg_pageinspect插件可以观察这些数据



SKY mobi
新手机 新应用 新娱乐

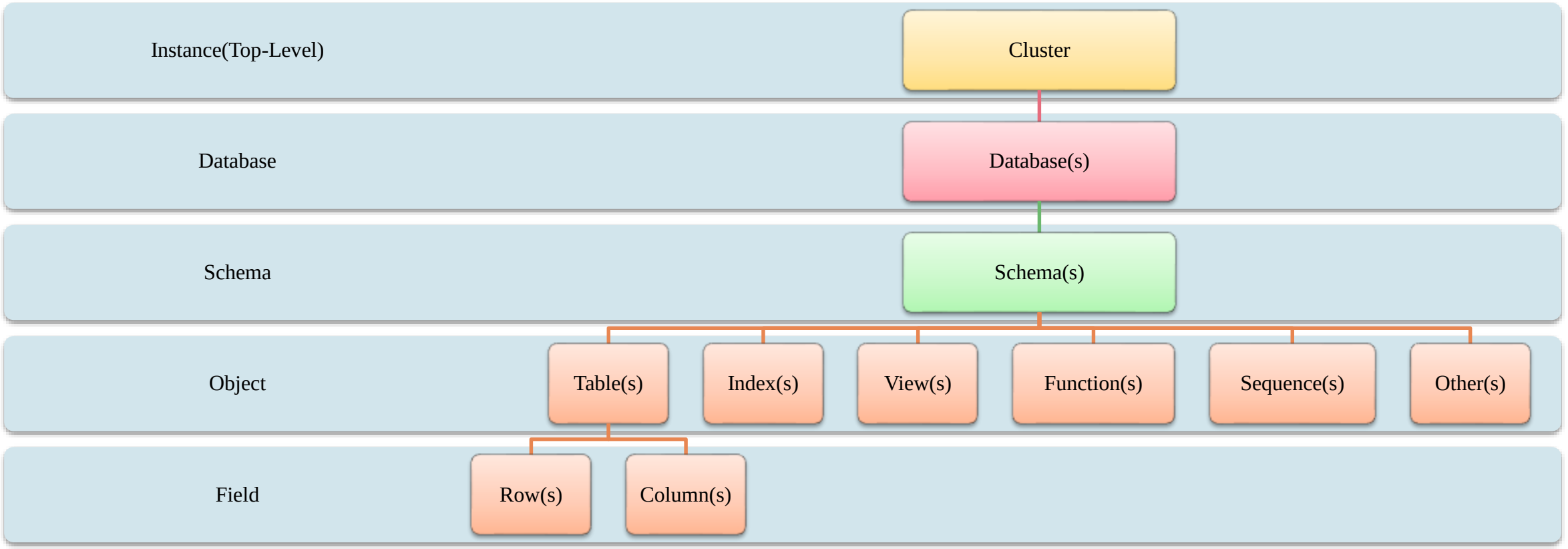
PostgreSQL 物理结构

- 图例



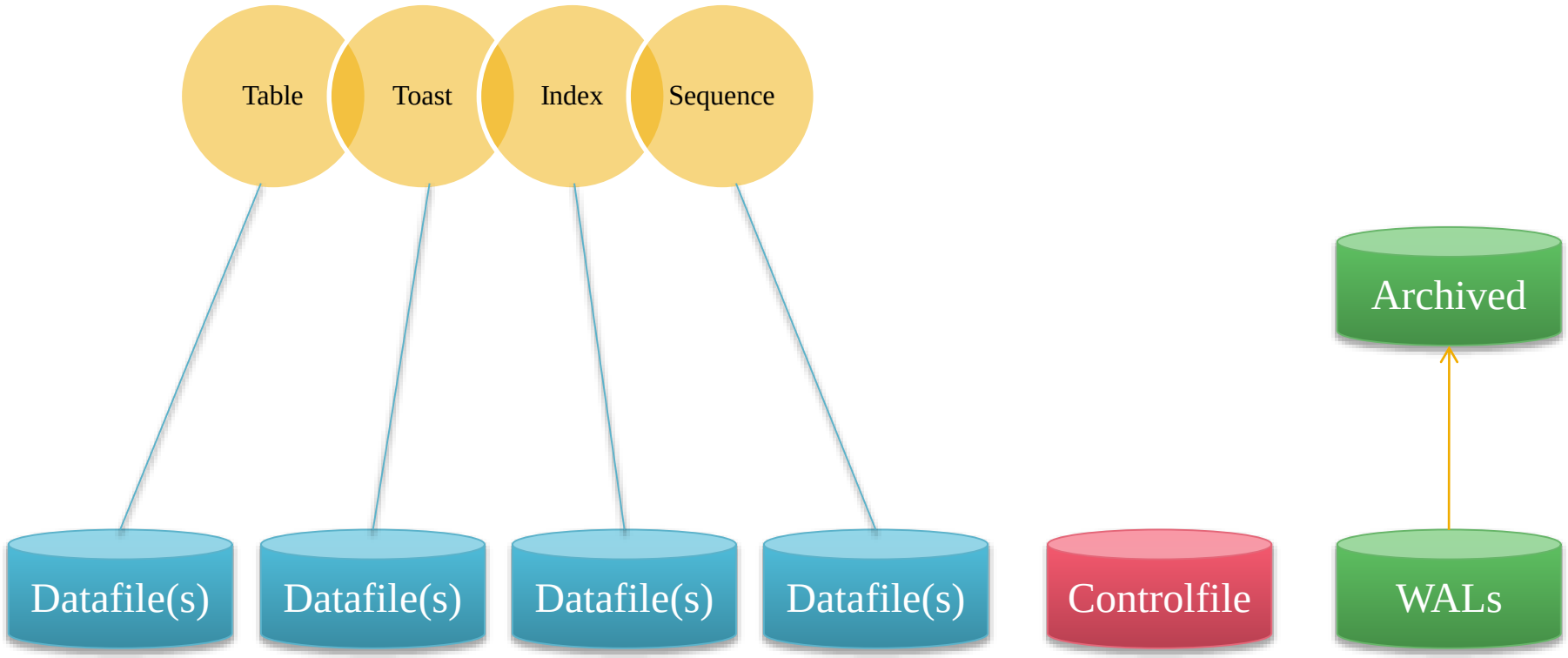
SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库逻辑概貌



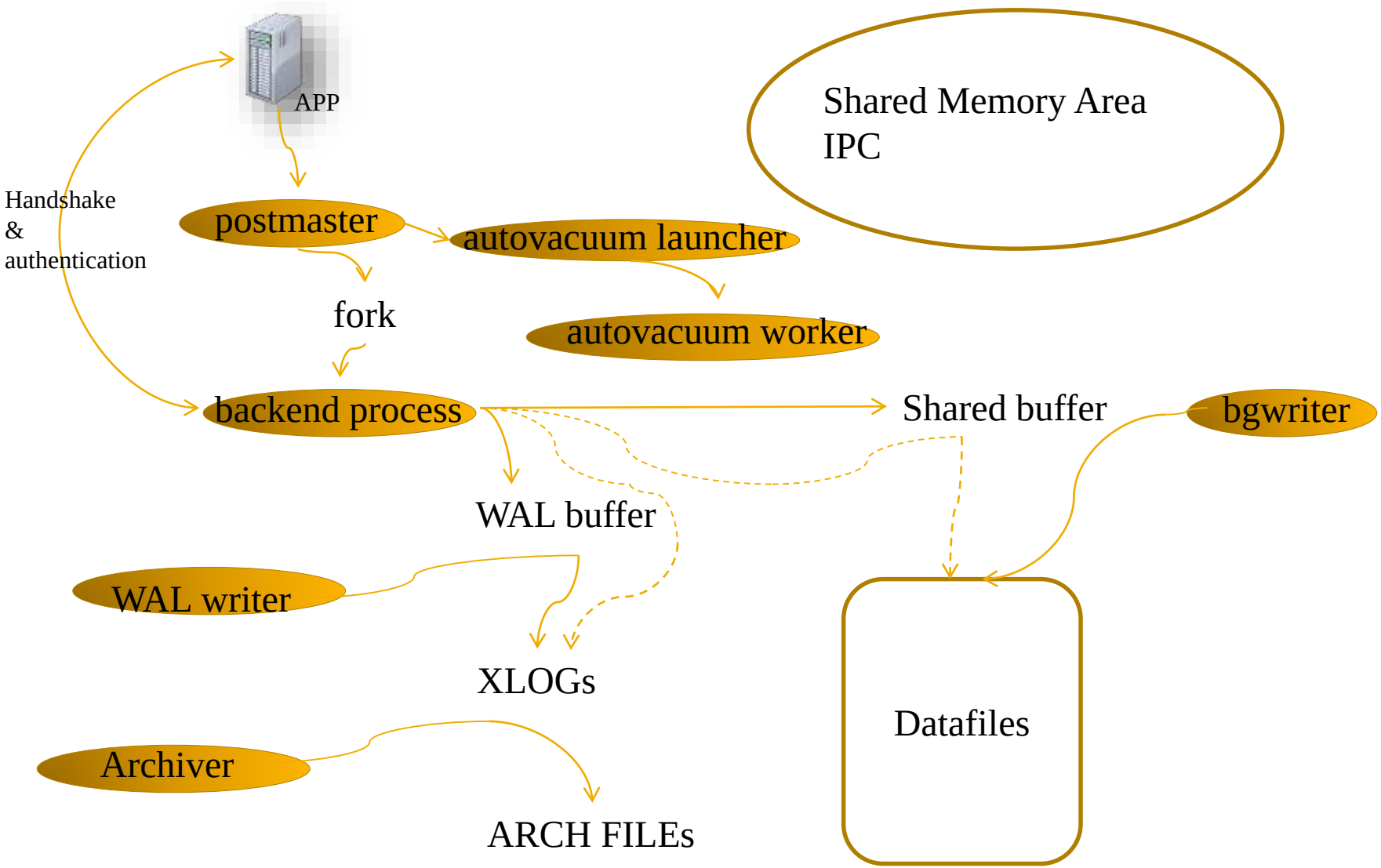
SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库物理存储概貌



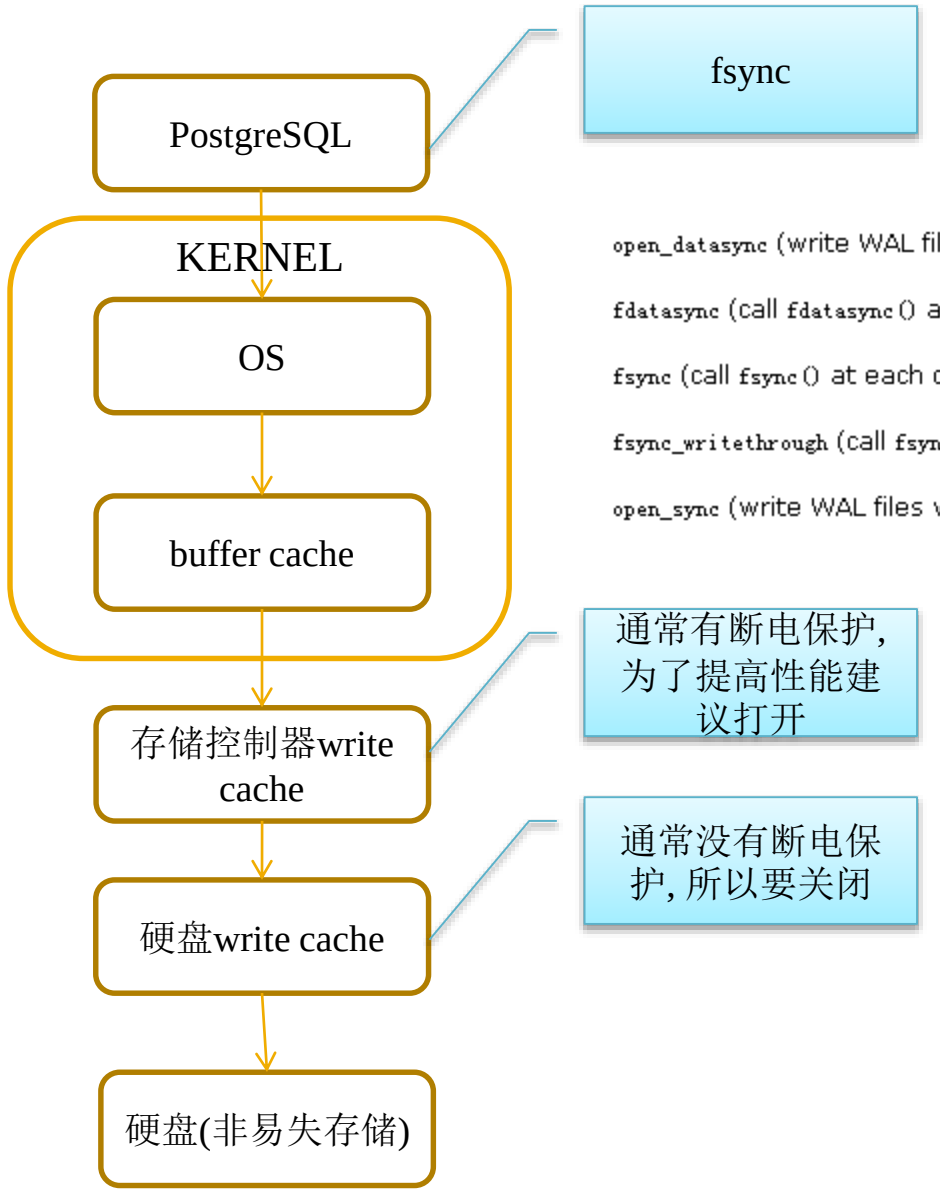
SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库进程结构概貌



SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库可靠性



PostgreSQL调用OS的sync WRITE函数.
wal_sync_method=?

open_datasync (write WAL files with open() option O_DSYNC)

fdatasync (call fdatasync() at each commit)

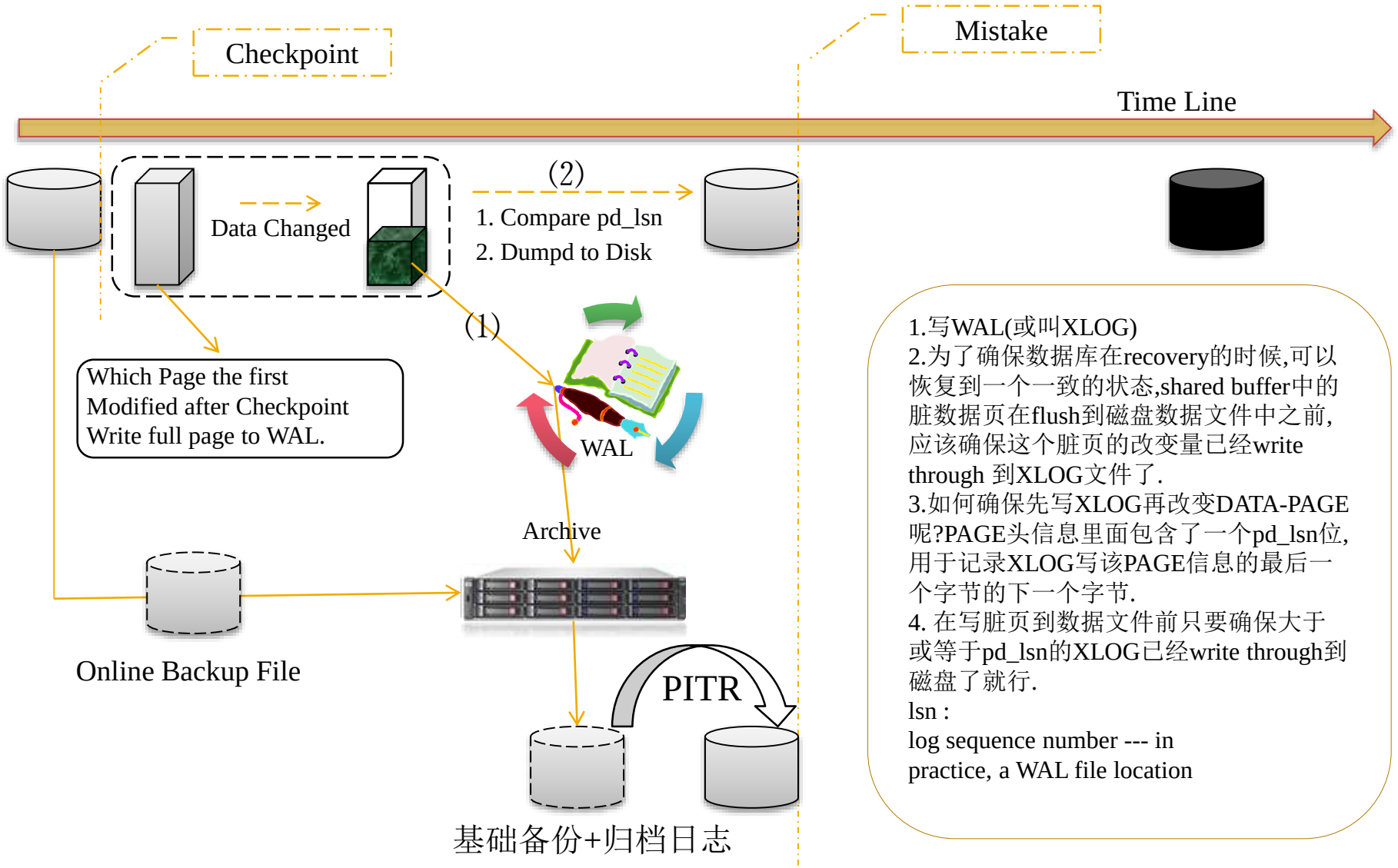
fsync (call fsync() at each commit)

fsync_writethrough (call fsync() at each commit, forcing write-through of any disk write cache)

open_sync (write WAL files with open() option O_SYNC)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库可靠性



1.写WAL(或叫XLOG)
2.为了确保数据库在recovery的时候,可以恢复到一个一致的状态,shared buffer中的脏数据页在flush到磁盘数据文件中之前,应该确保这个脏页的改变量已经write through 到XLOG文件了.
3.如何确保先写XLOG再改变DATA-PAGE呢?PAGE头信息里面包含了一个pd_lsn位,用于记录XLOG写该PAGE信息的最后一个字节的下一个字节.
4. 在写脏页到数据文件前只要确保大于或等于pd_lsn的XLOG已经write through到磁盘了就行.
lsn :
log sequence number --- in practice, a WAL file location

SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库可靠性注意事项

- 让数据库可靠的注意事项
  - 事务提交后确保这个事务未来可恢复吗?
    - 事务返回成功前, 事务日志(xlog)写入磁盘, synchronous_commit = on
  - 备份可恢复吗?恢复后确保数据一致吗?
    - -- fsync = on . full_page_writes = on
  - 必须写入非易失存储的数据已经写入到非易失存储了吗?
    - write - through , write - back
    - 关闭磁盘的write cache
    - 只允许有断电保护的write cache.
  - 主机异常DOWN机后重启数据库能不能起到一个一致的状态?
    - PostgreSQL periodically writes full page images to permanent WAL storage **before** modifying the actual page on disk. -- full_page_writes = on
  - 数据库异常DOWN机后重启数据库能不能起到一个一致的状态?
    - PostgreSQL periodically writes full page images to permanent WAL storage **before** modifying the actual page on disk. -- full_page_writes = on

SKY mobi
新手机 新应用 新娱乐

PostgreSQL数据库可靠性注意事项

- 让数据库可靠的注意事项
  - 事务日志可以用于恢复到任意时间点吗?
    - 开启归档, 并且有良好的备份策略.
    - wal_level = archive 或 hot_standby
  - 如果存储挂了怎么办?
    - 开启归档, 并且有良好的备份策略.
    - wal_level = archive 或 hot_standby
    - archive_mode = on
    - archive_command = 'cp %p /backup/%f'
  - 如果IDC挂了怎么办?
    - 开启归档, 并且有良好的备份策略.
    - wal_level = archive 或 hot_standby
    - 异地容灾, 如流复制.

SKY mobi
新手机 新应用 新娱乐

练习

- 系统表直接的关联关系的熟悉
- 系统视图的使用
- 系统管理函数的使用
- pageinspect插件观察数据块,tuple,数据

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 使用基础

- 使用基础
- 目标:
- 了解常用的数据库交互工具的使用
- 了解PG数据库的基本概念(类型, 语法等)和SQL操作

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 交互工具的使用

- psql 工具
- 详细的帮助参考man psql

- 两个比较有用的帮助, 在psql shell中输入 :
- \?　　-- 可以得到psql的一些快捷命令
- \h　　-- 可以得到SQL的语法帮助

- 常用的快捷命令
- \dt　-- 输出当前搜索路径下的表
- \set VERBOSITY verbose
  - 设置详细的打印输出, 例如可以报出问题的代码.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据库基本操作

- PostgreSQL数据类型介绍
- 表的操作(创建, 插入, 更新, 删除, 截断, 删除, 重命名, 修改表的属性...)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

Table 45-49. `typcategory` Codes

Code	Category
A	Array types
B	Boolean types
C	Composite types
D	Date/time types
E	Enum types
G	Geometric types
I	Network address types
N	Numeric types
P	Pseudo-types
S	String types
T	Timespan types
U	User-defined types
V	Bit-string types
X	unknown type

- p:
- e:
- m:
- x:

```
        Table "pg_catalog.pg_type"
     Column      |     Type     | Modifiers
-----------------+--------------+-----------
 typname         | name         | not null
 typnamespace    | oid          | not null
 typowner        | oid          | not null
 typlen          | smallint     | not null
 typbyval        | boolean      | not null
 typtype         | "char"       | not null
 typcategory     | "char"       | not null
 typispreferred  | boolean      | not null
 typisdefined    | boolean      | not null
 typdelim        | "char"       | not null
 typrelid        | oid          | not null
 typelem         | oid          | not null
 typarray        | oid          | not null
 typinput        | regproc      | not null
 typoutput       | regproc      | not null
 typreceive      | regproc      | not null
 typsend         | regproc      | not null
 typmodin        | regproc      | not null
 typmodout       | regproc      | not null
 typanalyze      | regproc      | not null
 typalign        | "char"       | not null
 typstorage      | "char"       | not null
 typnotnull      | boolean      | not null
 typbasetype     | oid          | not null
 typtypmod       | integer      | not null
 typndims        | integer      | not null
 typcollation    | oid          | not null
 typdefaultbin   | pg_node_tree |
 typdefault      | text         |
```

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 数字

Name	Storage Size	Description	Range
smallint	2 bytes	small-range integer	-32768 to +32767
integer	4 bytes	typical choice for integer	-2147483648 to +2147483647
bigint	8 bytes	large-range integer	-9223372036854775808 to 9223372036854775807
decimal / numeric	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
real	4 bytes	variable-precision, inexact	6 decimal digits precision
double precision	8 bytes	variable-precision, inexact	15 decimal digits precision
serial	4 bytes	autoincrementing integer	1 to 2147483647
bigserial	8 bytes	large autoincrementing integer	1 to 9223372036854775807

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 字符

Name	Storage Size	Description
character varying(n), varchar(n)	variable(can store n chars)	variable-length with limit
character(n), char(n)	n chars	fixed-length, blank padded
text	variable	variable unlimited length
"char"	1 byte	single-byte internal type
name	64 bytes	internal type for object names

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 时间

Name	Storage Size	Description	Low Value	High Value	Resolution
timestamp [(p)] [without time zone]	8 bytes	both date and time (no time zone)	4713 BC	294276 AD	1 microsecond / 14 digits
timestamp [(p)] with time zone	8 bytes	both date and time, with time zone	4713 BC	294276 AD	1 microsecond / 14 digits
date	4 bytes	date (no time of day)	4713 BC	5874897 AD	1 day
time [(p)] [without time zone]	8 bytes	time of day (no date)	00:00:00	24:00:00	1 microsecond / 14 digits
time [(p)] with time zone	12 bytes	times of day only, with time zone	00:00:00+1459	24:00:00-1459	1 microsecond / 14 digits
interval [fields] [(p)]	12 bytes	time interval	-178000000 years	178000000 years	1 microsecond / 14 digits

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 时间
- 特殊日期/时间输入

Input String	Valid Types	Description
epoch	date, timestamp	1970-01-01 00:00:00+00 (Unix system time zero)
infinity	date, timestamp	later than all other time stamps
-infinity	date, timestamp	earlier than all other time stamps
now	date, time, timestamp	current transaction's start time
today	date, timestamp	midnight today
tomorrow	date, timestamp	midnight tomorrow
yesterday	date, timestamp	midnight yesterday
allballs	time	00:00:00.00 UTC

- postgres=# select timestamp 'epoch',date 'infinity',time 'now',date 'today',time 'allballs';
- timestamp | date | time | date | time
- ---+---+---+---+---
- 1970-01-01 00:00:00 | infinity | 15:14:13.461166 | 2012-04-27 | 00:00:00

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 时间
- 时间输入输出格式

Style Specification	Description	Example
ISO	ISO 8601/SQL standard	1997-12-17 07:37:16-08
SQL	traditional style	12/17/1997 07:37:16.00 PST
POSTGRES	original style	Wed Dec 17 07:37:16 1997 PST
German	regional style	17.12.1997 07:37:16.00 PST

datestyle Setting	Input Ordering	Example Output
SQL, DMY	day/month/year	17/12/1997 15:37:16.00 CET
SQL, MDY	month/day/year	12/17/1997 07:37:16.00 PST
Postgres, DMY	day/month/year	Wed 17 Dec 07:37:16 1997 PST

- postgres=# set datestyle='SQL,DMY';
- postgres=# select now();
- 27/04/2012 15:49:51.373789 CST
- postgres=# set datestyle='SQL,MDY';
- postgres=# select now();
- 04/27/2012 15:50:07.882063 CST

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 时间
- 时间间隔interval 格式
- [@] quantity unit [quantity unit...] [direction]
- P quantity unit [quantity unit ...] [T [quantity unit ...]]
- P [years-months-days] [T hours:minutes:seconds]
- IntervalStyle样式

Abbreviation	Meaning
Y	Years
M	Months (in the date part)
W	Weeks
D	Days
H	Hours
M	Minutes (in the time part)
S	Seconds

Style Specification	Year-Month Interval	Day-Time Interval	Mixed Interval
sql_standard	1-2	3 4:05:06	-1-2 +3 -4:05:06
postgres	1 year 2 mons	3 days 04:05:06	-1 year -2 mons +3 days -04:05:06
postgres_verbose	@ 1 year 2 mons	@ 3 days 4 hours 5 mins 6 secs	@ 1 year 2 mons -3 days 4 hours 5 mins 6 secs ago
iso_8601	P1Y2M	P3DT4H5M6S	P-1Y-2M3DT-4H-5M-6S

- postgres=# show IntervalStyle ;
- postgres
- postgres=# select interval 'P-1Y-2M3DT-4H-5M-6S';
- -1 years -2 mons +3 days -04:05:06
- postgres=# select interval '1 day ago';
- -1 days
- postgres=# set IntervalStyle ='sql_standard';
- postgres=# select interval 'P-1Y-2M3DT-4H-5M-6S';
- -1-2 +3 -4:05:06

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 布尔逻辑

Name	Storage Size	Description
boolean	1 byte	state of true or false

- 真
- TRUE 't' 'true' 'y' 'yes' 'on' '1'

- 假
- FALSE 'f' 'false' 'n' 'no' 'off' '0'

- unknown (三价逻辑运算)
- NULL

- http://blog.163.com/digoal@126/blog/static/16387704020130242244617 5/

A AND *B*	True	Unknown	False
True	True	Unknown	False
Unknown	Unknown	Unknown	False
False	False	False	False

A OR *B*	True	Unknown	False
True	True	True	True
Unknown	True	Unknown	Unknown
False	True	Unknown	False

A	NOT *A*
True	False
Unknown	Unknown
False	True

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 常用数据类型, 枚举
- CREATE TYPE mood AS ENUM ('sad', 'ok', 'happy');
- CREATE TABLE person (name text,current_mood mood);
- INSERT INTO person VALUES ('Moe', 'happy');
- SELECT * FROM person WHERE current_mood = 'happy';
- name | current_mood
- Moe | happy
- (1 row)
- -- 输入一个不存在的枚举值, 将报错
- postgres=# SELECT * FROM person WHERE current_mood = 'happ';
- ERROR: invalid input value for enum mood: "happ"
- -- 避免报错的方法, 把枚举转换成text
- postgres=# SELECT * FROM person WHERE current_mood::text = 'happ';
- name | current_mood
- ------+--------------
- (0 rows)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 枚举值每一个在行中占用4 bytes :
- postgres=# select current_mood,pg_column_size(current_mood) from person;
- current_mood | pg_column_size
- --------------+----------------
- happy | 4

- 枚举的标签在定义中最大限制由NAMEDATALEN决定, 默认是64-1. 前面已经讲过.
- 查找枚举的数据结构 :
- postgres=# select oid,typname from pg_type where typname='mood';
- oid | typname
- ---------+---------
- 3952969 | mood
- postgres=# select * from pg_enum where enumtypid=3952969;
- enumtypid | enumsortorder | enumlabel
- -----------+---------------+-----------
- 3952969 | 1 | sad
- 3952969 | 2 | ok
- 3952969 | 3 | happy

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 枚举类型变更
- ALTER TYPE name ADD VALUE new_enum_value [{ BEFORE | AFTER } existing_enum_value]
- This form adds a new value to an enum type. If the new value's place in the enum's ordering is not specified using BEFORE or AFTER, then the new item is placed at the end of the list of values.

- 注意事项, 添加枚举元素时尽量不要改动原来的元素的位置, 即尽量新增值插到最后.
- 否则可能会带来性能问题.
- ALTER TYPE ... ADD VALUE (the form that adds a new value to an enum type) cannot be executed inside a transaction block.
- Comparisons involving an added enum value will sometimes be slower than comparisons involving only original members of the enum type. This will usually only occur if BEFORE or AFTER is used to set the new value's sort position somewhere other than at the end of the list. However, sometimes it will happen even though the new value is added at the end (this occurs if the OID counter "wrapped around" since the original creation of the enum type). The slowdown is usually insignificant; but if it matters, optimal performance can be regained by dropping and recreating the enum type, or by dumping and reloading the database.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- money类型
- 显示和客户端参数lc_monetary有关

Name	Storage Size	Description	Range
money	8 bytes	currency amount	-92233720368547758.08 to +92233720368547758.07

- postgres=# show lc_monetary;
- C
- postgres=# SELECT '12.345'::money;
- $12.35
- postgres=# set lc_monetary='zh_CN';
- postgres=# SELECT '12.345'::money;
- ￥12.35

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- bytea类型

Name	Storage Size	Description
bytea	1 or 4 bytes plus the actual binary string	variable-length binary string

- The bytea data type allows storage of binary strings
- A binary string is a sequence of octets (or bytes)
- bytea与字符类型的区别
  - binary strings specifically allow storing octets of value zero and other "non-printable" octets.
  - Character strings disallow zero octets, and also disallow any other octet values and sequences of octet values that are invalid according to the database's selected character set encoding.
  - Second, operations on binary strings process the actual bytes, whereas the processing of character strings depends on locale settings. In short, binary strings are appropriate for storing data that the programmer thinks of as "raw bytes", whereas character strings are appropriate for storing text.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- bytea类型
- 同时支持两种格式输入
  - escape
    - select E'\\336\\255\\276\\357'::bytea;
  - hex, 每两个16进制数字为一组, 表示一个"raw byte"
    - SELECT E'\\x DE AD BE EF'::bytea;
- 支持两种格式输出, 需配置
  - 9.0引入hex输出(通过配置bytea_output)
  - 9.0以前为escape输出
  - 如果有从老版本数据库迁移到9.0及以后版本的情况, 需要注意, 可能再次与程序不兼容, 只需要将默认值调整为escape即可.
- 推荐使用hex格式输入输出

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 几何类型

Name	Storage Size	Representation	Description
point	16 bytes	Point on a plane	(x,y)
line	32 bytes	Infinite line (not fully implemented)	((x1,y1),(x2,y2))
lseg	32 bytes	Finite line segment	((x1,y1),(x2,y2))
box	32 bytes	Rectangular box	((x1,y1),(x2,y2))
path	16+16n bytes	Closed path (similar to polygon)	((x1,y1),...)
path	16+16n bytes	Open path	[(x1,y1),...]
polygon	40+16n bytes	Polygon (similar to closed path)	((x1,y1),...)
circle	24 bytes	Circle	<(x,y),r> (center point and radius)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 网络地址类型

Name	Storage Size	Description
cidr	7 or 19 bytes	IPv4 and IPv6 networks
inet	7 or 19 bytes	IPv4 and IPv6 hosts and networks
macaddr	6 bytes	MAC addresses

cidr Input	cidr Output	abbrev(cidr)
192.168.100.128/25	192.168.100.128/25	192.168.100.128/25
192.168/24	192.168.0.0/24	192.168.0/24
192.168/25	192.168.0.0/25	192.168.0.0/25
192.168.1	192.168.1.0/24	192.168.1/24
192.168	192.168.0.0/24	192.168.0/24
128.1	128.1.0.0/16	128.1/16
128	128.0.0.0/16	128.0/16
128.1.2	128.1.2.0/24	128.1.2/24
10.1.2	10.1.2.0/24	10.1.2/24
10.1	10.1.0.0/16	10.1/16
10	10.0.0.0/8	10/8
10.1.2.3/32	10.1.2.3/32	10.1.2.3/32
2001:4f8:3:ba::/64	2001:4f8:3:ba::/64	2001:4f8:3:ba::/64
2001:4f8:3:ba:2e0:81ff:fe22:d1f1/128	2001:4f8:3:ba:2e0:81ff:fe22:d1f1/128	2001:4f8:3:ba:2e0:81ff:fe22:d1f1
::ffff:1.2.3.0/120	::ffff:1.2.3.0/120	::ffff:1.2.3/120
::ffff:1.2.3.0/128	::ffff:1.2.3.0/128	::ffff:1.2.3.0/128

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 网段填充例子：
- Table "digoal.tbl_ip_info"
- Column | Type | Modifiers
- ----------+----------------------+-----------
- id | integer |
- province | character varying(10) | 省份
- start_ip | inet | 开始IP
- end_ip | inet | 结束IP

- digoal=> insert into tbl_ip_info values (1,'浙江','192.168.1.254','192.168.2.5');
- digoal=> insert into tbl_ip_info values (2,'广东','192.168.2.254','192.168.3.5');
- digoal=> insert into tbl_ip_info values (3,'湖南','192.168.3.254','192.168.4.5');

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

```
digoal=> select id,generate_series(0,end_ip-start_ip)+start_ip from tbl_ip_info ;

id |   ?column?
----+--------------
  1 | 192.168.1.254
  1 | 192.168.1.255
  1 | 192.168.2.0
  1 | 192.168.2.1
  1 | 192.168.2.2
  1 | 192.168.2.3
  1 | 192.168.2.4
  1 | 192.168.2.5
  2 | 192.168.2.254
  2 | 192.168.2.255
  2 | 192.168.3.0
  2 | 192.168.3.1
  2 | 192.168.3.2
  2 | 192.168.3.3
  2 | 192.168.3.4
  2 | 192.168.3.5
  3 | 192.168.3.254
  3 | 192.168.3.255
  3 | 192.168.4.0
  3 | 192.168.4.1
  3 | 192.168.4.2
  3 | 192.168.4.3
  3 | 192.168.4.4
  3 | 192.168.4.5
(24 rows)
```

SKYY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 比特类型, 支持变长和定长
- Bit strings are strings of 1's and 0's. They can be used to store or visualize bit masks. There are two SQL bit types: bit(n) and bit varying(n), where n is a positive integer.

- CREATE TABLE test (a BIT(3), b BIT VARYING(5));
- INSERT INTO test VALUES (B'101', B'00');
- INSERT INTO test VALUES (B'10', B'101');
- ERROR: bit string length 2 does not match type bit(3)

- INSERT INTO test VALUES (B'10'::bit(3), B'101');
- SELECT * FROM test;
- a | b
- -----+-----
- 101 | 00
- 100 | 101

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 全文检索类型
- tsvector
  - 去除重复分词后按分词顺序存储
  - 可以存储位置信息和权重信息
- tsquery
  - 存储查询的分词，可存储权重信息

```
postgres=# select * from pg_ts_dict ;
    dictname     | dictnamespace | dictowner | dicttemplate |                    dictinitoption
-----------------+---------------+-----------+--------------+-----------------------------------------------------
 simple          |            11 |        10 |         3727 |
 danish_stem     |            11 |        10 |        12144 | language = 'danish', stopwords = 'danish'
 dutch_stem      |            11 |        10 |        12144 | language = 'dutch', stopwords = 'dutch'
 english_stem    |            11 |        10 |        12144 | language = 'english', stopwords = 'english'
 finnish_stem    |            11 |        10 |        12144 | language = 'finnish', stopwords = 'finnish'
 french_stem     |            11 |        10 |        12144 | language = 'french', stopwords = 'french'
 german_stem     |            11 |        10 |        12144 | language = 'german', stopwords = 'german'
 hungarian_stem  |            11 |        10 |        12144 | language = 'hungarian', stopwords = 'hungarian'
 italian_stem    |            11 |        10 |        12144 | language = 'italian', stopwords = 'italian'
 norwegian_stem  |            11 |        10 |        12144 | language = 'norwegian', stopwords = 'norwegian'
 portuguese_stem |            11 |        10 |        12144 | language = 'portuguese', stopwords = 'portuguese'
 romanian_stem   |            11 |        10 |        12144 | language = 'romanian'
 russian_stem    |            11 |        10 |        12144 | language = 'russian', stopwords = 'russian'
 spanish_stem    |            11 |        10 |        12144 | language = 'spanish', stopwords = 'spanish'
 swedish_stem    |            11 |        10 |        12144 | language = 'swedish', stopwords = 'swedish'
 turkish_stem    |            11 |        10 |        12144 | language = 'turkish', stopwords = 'turkish'
```

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 全文检索类型

```
SELECT $$the lexeme 'Joe''s' contains a quote$$::tsvector;
                    tsvector
-----------------------------------------------
 'Joe''s' 'a' 'contains' 'lexeme' 'quote' 'the'
```

```
SELECT 'a:1 fat:2 cat:3 sat:4 on:5 a:6 mat:7 and:8 ate:9 a:10 fat:11 rat:12'::tsvector;
                                  tsvector
-------------------------------------------------------------------------------
 'a':1,6,10 'and':8 'ate':9 'cat':3 'fat':2,11 'mat':7 'on':5 'rat':12 'sat':4
```

```
SELECT 'a:1A fat:2B,4C cat:5D'::tsvector;
          tsvector
----------------------------
 'a':1A 'cat':5 'fat':2B,4C
```

```
SELECT to_tsvector('english', 'The Fat Rats');
   to_tsvector
-----------------
 'fat':2 'rat':3
```

```
SELECT to_tsvector( 'postgraduate' ) @@ to_tsquery( 'postgres:*' );
 ?column?
----------
 t
(1 row)
```

```
SELECT 'fat & rat'::tsquery;
    tsquery
---------------
 'fat' & 'rat'

SELECT 'fat & (rat | cat)'::tsquery;
          tsquery
---------------------------
 'fat' & ( 'rat' | 'cat' )

SELECT 'fat & rat & ! cat'::tsquery;
        tsquery
------------------------
 'fat' & 'rat' & !'cat'
```

```
SELECT 'fat:ab & cat'::tsquery;
     tsquery
------------------
 'fat':AB & 'cat'
```

```
SELECT to_tsquery('postgres:*');
 to_tsquery
------------
 'postgr':*
(1 row)
```

label * specify prefix matching

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- uuid类型
- UUIDs could be generated by client applications or other libraries invoked through a server-side function.
- specifically a group of 8 digits followed by three groups of 4 digits followed by a group of 12 digits, for a total of 32 digits representing the 128 bits.

输出格式:

```
a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11
```

输入格式:

```
A0EEBC99-9C0B-4EF8-BB6D-6BB9BD380A11
{a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11}
a0eebc999c0b4ef8bb6d6bb9bd380a11
a0ee-bc99-9c0b-4ef8-bb6d-6bb9-bd38-0a11
{a0eebc99-9c0b4ef8-bb6d6bb9-bd380a11}
```

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- xml
- Use of this data type requires the installation to have been built with configure --with-libxml
- 构造xml类型的语法
- SQL标准写法
- XMLPARSE ({ DOCUMENT | CONTENT } value)
- 例如
- XMLPARSE (DOCUMENT '<?xml version="1.0"?><book><title>Manual</title><chapter>...</chapter></book>')
- XMLPARSE (CONTENT 'abc<foo>bar</foo><bar>foo</bar>')
- PostgreSQL写法
- xml '<foo>bar</foo>'
- '<foo>bar</foo>'::xml
- 从xml到字符串的转换
- XMLSERIALIZE ({ DOCUMENT | CONTENT } value AS type)
- 例如
- XMLSERIALIZE (CONTENT '<foo>bar</foo>'::xml AS text)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 数组类型
- 不限长度
  - 目前PostgreSQL未对长度强限定, 如int[]和int[10]都不会限定元素个数.
  - array_length(ARRAY[[1,2,3,4,5],[6,7,8,9,10]], 1)
- 不限维度
  - 目前PostgreSQL未对维度强限定,如int[]和int[][], 效果是一样的, 都可以存储任意维度的数组.
- 矩阵强制
  - 多维数组中, 同一个维度的元素个数必须相同.
  - 正确
  - array[[1,2,3,4],[5,6,7,8]]
  - 不正确
  - array[[1,2,3,4],[5,6,7]]
- 元素强制
  - 元素类型必须一致
  - 正确
  - array[1,2,3]
  - 不正确
  - array[1,2,'abc']

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 扩展
  - 一维数组支持prepend, append, cat操作
    - array_append(ARRAY['digoal','francs'],'david')
    - array_prepend('david',ARRAY['digoal','francs'])
  - 二维数组仅支持cat操作
    - array_cat(ARRAY[['digoal','zhou'],['francs','tan']], ARRAY['david','guo'])
- subscript
  - 元素脚本默认从1开始, 也可以指定.
  - array_lower(ARRAY[[1,2,3,4,5],[6,7,8,9,10]], 2)
  - array_lower('[-3:-2]={1,2}'::int[], 1)
  - select array_upper('[-3:-2]={1,2}'::int[], 1)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 切片
  - array_dims(ARRAY[[1,2,3,4,5],[6,7,8,9,10]])
- a[1:2][1:1] = {{1},{3}}
- 第一个[]中的1表示低位subscript, 2表示高位subscript值.
- 第二个[]中左边的1表示低位subscript, 右边的1表示高位subscript值.
- a[2:3][1:2] = {{3,4},{5,6}}
- 分片的另一种写法, 只要其中的一个维度用了分片写法, 其他的维度如果没有使用分片写法, 默认视为高位
- 如a[2:3][2] 等同于 a[2:3][1:2]

- **PostgreSQL ARRAY datatype introduce**
- http://blog.163.com/digoal@126/blog/static/16387704020120127592252 9/

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 数组相关的 函数 与 操作符

oprname	oprleft	oprright	oprresult	oprcode	oprrest	oprjoin
||	2277	2283	2277	array_append	-	-
||	2283	2277	2277	array_prepend	-	-
||	2277	2277	2277	array_cat	-	-
=	2277	2277	16	array_eq	eqsel	eqjoinsel
<>	2277	2277	16	array_ne	neqsel	neqjoinsel
<	2277	2277	16	array_lt	scalarltsel	scalarltjoinsel
>	2277	2277	16	array_gt	scalargtsel	scalargtjoinsel
<=	2277	2277	16	array_le	scalarltsel	scalarltjoinsel
>=	2277	2277	16	array_ge	scalargtsel	scalargtjoinsel
&&	2277	2277	16	arrayoverlap	areasel	areajoinsel
@>	2277	2277	16	arraycontains	contsel	contjoinsel
<@	2277	2277	16	arraycontained	contsel	contjoinsel

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 自定义类型
- create type test as (info text,id int,crt_time timestamp(0));
- 创建表时默认创建一个同名composite type, 因此表名和自定义类名不能重复
- create table test (id int primary key,info text);
- ERROR: relation "test" already exists
- 举例
- CREATE TYPE inventory_item AS (
- name text,
- supplier_id integer,
- price numeric
-);
- CREATE TABLE on_hand (
- item inventory_item,
- count integer
-);

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- 自定义类型
- INSERT INTO on_hand VALUES (ROW('fuzzy dice', 42, 1.99), 1000);
- SELECT (on_hand.item).name FROM on_hand WHERE (on_hand.item).price <10;
- name
- ----------
- fuzzy dice

SET和INTO子句自定义类型不能加括号引用,其他子句中的自定义类型可以加括号引用

- UPDATE on_hand SET item = ROW('fuzzy dice',10,100) WHERE count=1000;
- UPDATE on_hand SET item.price = (on_hand.item).price + 100 WHERE (on_hand.item).name='fuzzy dice';
- INSERT INTO on_hand (item.name, item.supplier_id) VALUES('test', 2.2);
- postgres=# select * from on_hand;
- item | count
- ----------+-------
- ("fuzzy dice",10,200) | 1000
- (test,2,) |

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- oid (object identifier) 4 bytes
- xid (transaction identifier) 4 bytes xmin,xmax
- cid (command identifier) 4 bytes cmin,cmax
- tid (tuple identifier) 6 bytes ctid

- 以下为各系统表对应的oid列的alias, 类型都是oid
- 可使用namespace, 或者默认的search_path先后顺序检索

Name	References	Description	Value Example
oid	any	numeric object identifier	564182
regproc	pg_proc	function name	sum
regprocedure	pg_proc	function with argument types	sum(int4)
regoper	pg_operator	operator name	+
regoperator	pg_operator	operator with argument types	*(integer,integer) or -(NONE,integer)
regclass	pg_class	relation name	pg_type
regtype	pg_type	data type name	integer
regconfig	pg_ts_config	text search configuration	english
regdictionary	pg_ts_dict	text search dictionary	simple

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- test=# create sequence seq_test start with 1;
- CREATE SEQUENCE
- test=# select 'seq_test'::regclass;
- regclass
- seq_test
- test=# select 'seq_test'::regclass::oid;
- oid
- 49247
- test=# select 'sum(int4)'::regprocedure;
- regprocedure
- sum(integer)
- test=# select 'sum(int4)'::regprocedure::oid;
- oid
- 2108

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 数据类型介绍

- Pseudo-Types 伪类型

Name	Description
any	Indicates that a function accepts any input data type.
anyarray	Indicates that a function accepts any array data type (see Section 35.2.5).
anyelement	Indicates that a function accepts any data type (see Section 35.2.5).
anyenum	Indicates that a function accepts any enum data type (see Section 35.2.5 and Section 8.7).
anynonarray	Indicates that a function accepts any non-array data type (see Section 35.2.5).
cstring	Indicates that a function accepts or returns a null-terminated C string.
internal	Indicates that a function accepts or returns a server-internal data type.
language_handler	A procedural language call handler is declared to return language_handler.
fdw_handler	A foreign-data wrapper handler is declared to return fdw_handler.
record	Identifies a function returning an unspecified row type.
trigger	A trigger function is declared to return trigger.
void	Indicates that a function returns no value.
opaque	An obsolete type name that formerly served all the above purposes.

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 表操作

- SQL Syntax-Lexical Structure
- 例子
- SELECT * FROM pg_class WHERE relname = 'pg_statistic' LIMIT 1; -- is one comment

- TOKEN
  - keyword (SELECT FROM WHERE LIMIT) 所有的关键字参考数据库手册
  - identifier or quoted identifier (pg_class relname, 表名, 列名,对象名...)
    - 默认小写, 如需大写需使用双引号
  - literal or constant ('pg_statistic' 1)
  - special character symbol (*)
  - comment (-- is one-line comment) or (/* */ multi-line comment)
  - operator (=)

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 表操作

- Identifier最大长度受以下限制
- define NAMEDATALEN 64 (长度限制, 截断到63)
- truncate_identifier(char *ident, int len, bool warn)
- {
- if (len >= NAMEDATALEN)
- {
- len = pg_mbcliplen(ident, len, NAMEDATALEN - 1);
- if (warn)
- {
- char buf[NAMEDATALEN];
- memcpy(buf, ident, len);
- buf[len] = '\0';
- ereport(NOTICE,
- (errcode(ERRCODE_NAME_TOO_LONG),
- errmsg("identifier \"%s\" will be truncated to \"%s\"",
- ident, buf)));
- }
- ident[len] = '\0';
- }
- }

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 表操作

- 改变最大长度限制,需重新*initdb*

- src/include/pg_config_manual.h
- /*
- * Maximum length for identifiers (e.g. table names, column names,
- * function names). Names actually are limited to one less byte than this,
- * because the length must include a trailing zero byte.
- *
- * Changing this requires an initdb.
- */
- #define NAMEDATALEN 64

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 表操作

- implicitly-typed literal or constant
  - string
    - E'digoal\\'
    - $$digoal\$$
    - tagdigoal\tag
  - bit string
    - B'1010101'
  - number
    - 10 or +10
    - -23.4
    - +100.1 or 100.1
    - 10e-1
    - 98e+10 or 98e10
- explicit-typed literal or constant
  - type 'string'
    - time '12:00:00'
  - 'string'::type
    - '1 hour'::interval
  - CAST ('string' AS type)
    - CAST('127.0.0.1' AS inet);

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 表操作

- 操作符
- + - * / < > = ~ ! @ # % ^ & | ` ? ||

- postgres=# select count(*) from pg_operator;
- count
- -------
- 706

- 也可以使用这种用法, 解析器同样支持.
- SELECT 3 OPERATOR(pg_catalog.+) 4;

```
   Table "pg_catalog.pg_operator"
    Column     |  Type    | Modifiers
---------------+----------+-----------
 oprname       | name     | not null
 oprnamespace  | oid      | not null
 oprowner      | oid      | not null
 oprkind       | "char"   | not null
 oprcanmerge   | boolean  | not null
 oprcanhash    | boolean  | not null
 oprleft       | oid      | not null
 oprright      | oid      | not null
 oprresult     | oid      | not null
 oprcom        | oid      | not null
 oprnegate     | oid      | not null
 oprcode       | regproc  | not null
 oprrest       | regproc  | not null
 oprjoin       | regproc  | not null
```

SKY mobi
新手机 新应用 新娱乐

PostgreSQL 表操作

- 特殊字符
- $
  - string quoted
  - positional parameter in function or prepared statement
- ()
  - enforce precedence
- []
  - array selected elements
- ,
  - separate the elements of a list
- ;
  - terminate a SQL
- :
  - slice from array
- *
  - all the fields of a table or composite value
- .
  - numeric , separate schema, table, column names.

SKY mobi
新手机 新应用 新娱乐

事务操作

- 事务相关的TOKEN.
- BEGIN; -- 可指定事务隔离级别, 事务读写属性, 约束延迟校验属性等.
- COMMIT;
- ROLLBACK;
- SAVEPOINT a;
- ROLLBACK to a;

- 二阶事务相关TOKEN,
- prepare transaction
- rollback prepared
- commit prepared

- psql相关的事务模式变量
- ON_ERROR_ROLLBACK, ON_ERROR_STOP
- postgres=# \set ON_ERROR_ROLLBACK on
- 如果开启ON_ERROR_ROLLBACK, 会在每一句SQL前设置隐形的savepoint, 可以继续下面的SQL, 而不用全部回滚

SKY mobi
新手机 新应用 新娱乐

单条SQL插入多行

- INSERT INTO tbl(c1,…,cn) values (…),(…),…,(…);
- 例如
- digoal=# drop table user_info ;
- DROP TABLE
- digoal=# create table user_info(id int, info text);
- CREATE TABLE
- digoal=# insert into user_info(id,info) values(1,'test'),(1,'test'),(1,'test'),(1,'test'),(1,'test');
- INSERT 0 5
- digoal=# select ctid,cmin,cmax,xmin,xmax,* from user_info ;
- ctid | cmin | cmax | xmin | xmax | id | info -- 几个隐含字段的简单解释, 这种SQL的好处.
- -------+------+------+-----------+------+----+------
- (0,1) | 14 | 14 | 216732454 | 0 | 1 | test
- (0,2) | 14 | 14 | 216732454 | 0 | 1 | test
- (0,3) | 14 | 14 | 216732454 | 0 | 1 | test
- (0,4) | 14 | 14 | 216732454 | 0 | 1 | test
- (0,5) | 14 | 14 | 216732454 | 0 | 1 | test
- (5 rows)

SKY mobi
新手机 新应用 新娱乐

DML

- INSERT
- UPDATE
- DELETE

- 一个事务最大2^32条SQL(因为cmin,cmax的长度是4Bytes)
- PostgreSQL一个事务中可以包含DML, DDL, DCL.
  - 除了以下
  - create tablespace
  - create database
  - 使用concurrently并行创建索引
  - 其他未尽情况略
- (Oracle执行DDL前自动将前面的未提交的事务提交,所以Oracle不支持在事务中执行DDL语句)

SKY mobi
新手机 新应用 新娱乐

Query

- JOIN
- ALIAS
- Table as Function's Return data type
- GROUP BY [HAVING]
- DISTINCT
- COMBINING QUERY
- SORT
- LIMIT [OFFSET]
- WITH

SKY mobi
新手机 新应用 新娱乐

JOIN

- T1 CROSS JOIN T2 (T1 INNER JOIN T2 ON TRUE)
- T1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] } JOIN T2 ON boolean_expression
- T1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] } JOIN T2 USING (join column list)
- T1 NATURAL { [INNER] | { LEFT | RIGHT | FULL } [OUTER] } JOIN T2

- The words INNER and OUTER are optional in all forms. INNER is the default; LEFT, RIGHT, and FULL imply an outer join.

num	name
1	a
2	b
3	c

num	value
1	xxx
3	yyy
5	zzz

SKY mobi
新手机 新应用 新娱乐

JOIN

- CROSS JOIN 产生笛卡尔积

```
=> SELECT * FROM t1 CROSS JOIN t2;
 num | name | num | value
-----+------+-----+-------
   1 | a    |   1 | xxx
   1 | a    |   3 | yyy
   1 | a    |   5 | zzz
   2 | b    |   1 | xxx
   2 | b    |   3 | yyy
   2 | b    |   5 | zzz
   3 | c    |   1 | xxx
   3 | c    |   3 | yyy
   3 | c    |   5 | zzz
(9 rows)
```

- 当关联的两个表有两列同名同类型同长度, 以下三种写法等同

```
=> SELECT * FROM t1 INNER JOIN t2 ON t1.num = t2.num;
 num | name | num | value
-----+------+-----+-------
   1 | a    |   1 | xxx
   3 | c    |   3 | yyy
(2 rows)

=> SELECT * FROM t1 INNER JOIN t2 USING (num);
 num | name | value
-----+------+-------
   1 | a    | xxx
   3 | c    | yyy
(2 rows)

=> SELECT * FROM t1 NATURAL INNER JOIN t2;
 num | name | value
-----+------+-------
   1 | a    | xxx
   3 | c    | yyy
(2 rows)
```

SKY mobi
新手机 新应用 新娱乐

JOIN

- 左或右连接, 不满足条件的右或左表的值置空

```
=> SELECT * FROM t1 LEFT JOIN t2 ON t1.num = t2.num;
 num | name | num | value
-----+------+-----+-------
   1 | a    |   1 | xxx
   2 | b    |     |
   3 | c    |   3 | yyy
(3 rows)

=> SELECT * FROM t1 LEFT JOIN t2 USING (num);
 num | name | value
-----+------+-------
   1 | a    | xxx
   2 | b    |
   3 | c    | yyy
(3 rows)

=> SELECT * FROM t1 RIGHT JOIN t2 ON t1.num = t2.num;
 num | name | num | value
-----+------+-----+-------
   1 | a    |   1 | xxx
   3 | c    |   3 | yyy
     |      |   5 | zzz
(3 rows)
```

- 全关联,不满足条件的值都置空

```
=> SELECT * FROM t1 FULL JOIN t2 ON t1.num = t2.num;
 num | name | num | value
-----+------+-----+-------
   1 | a    |   1 | xxx
   2 | b    |     |
   3 | c    |   3 | yyy
     |      |   5 | zzz
(4 rows)
```

SKY mobi
新手机 新应用 新娱乐

ALIAS

- table alias:
- FROM table_reference AS alias
- FROM table_reference alias

- colum alias:
- SELECT expression [[AS] output_name]

- subquery alias:
- FROM (SELECT * FROM table1) AS alias_name

SKY mobi
新手机 新应用 新娱乐

Table as Function's Return data type

- return table's row type
  - create table t1 (id int,name text,crt_time timestamp(0));
  - create or replace function f_t1 (i_id int) returns setof t1 as $$
  - declare
  - begin
  - return query select * from t1 where id=i_id;
  - return;
  - end;
  - $$ language plpgsql;
  - insert into t1 values(1,'digoal',now());
  - insert into t1 values(1,'DIGOAL',now());
  - select * from f_t1(1);
  - id | name | crt_time
  - ----+--------+---------------------
  - 1 | digoal | 2012-04-26 08:15:09
  - 1 | DIGOAL | 2012-04-26 08:15:15

SKY mobi
新手机 新应用 新娱乐

Composite Type as Function's Return data type

- return composite type
  - create type type1 as (id int,name text,crt_time timestamp(0));
  - create or replace function f_type1 (i_id int) returns setof type1 as $$
  - declare
  - begin
  - return query select * from t1 where id=i_id;
  - return;
  - end;
  - $$ language plpgsql;

  - select * from f_type1(1);
  - id | name | crt_time
  - ----+--------+---------------------
  - 1 | digoal | 2012-04-26 08:15:09
  - 1 | DIGOAL | 2012-04-26 08:15:15

SKY mobi
新手机 新应用 新娱乐

Record as Function's Return data type

- return record
  - create or replace function f_record1 (i_id int) returns setof record as $$
  - declare
  - begin
  - return query select * from t1 where id=i_id;
  - return;
  - end;
  - $$ language plpgsql;

  - select * from f_record1(1) as (id int,name text,crt_time timestamp(0));
  - id | name | crt_time
  - ----+--------+---------------------
  - 1 | digoal | 2012-04-26 08:15:09
  - 1 | DIGOAL | 2012-04-26 08:15:15

SKY mobi
新手机 新应用 新娱乐

DISTINCT

- SELECT DISTINCT select_list ... (NULL在DISTINCT [ON] 中视为相等)
  - postgres=# select * from t1 ;
  - id | name | crt_time
  - 1 | digoal | 2012-04-26 08:15:09
  - 1 | DIGOAL | 2012-04-26 08:15:15

  - postgres=# select distinct id from t1;
  - id
  - 1
- SELECT DISTINCT ON (expression [, expression ...]) select_list ... -- ON()里面必须出现在order by中作为前导列
  - Here expression is an arbitrary value expression that is evaluated for all rows. A set of rows for which all the expressions are equal are considered duplicates, and only the first row of the set is kept in the output. Note that the "first row" of a set is unpredictable unless the query is sorted on enough columns to guarantee a unique ordering of the rows arriving at the DISTINCT filter. (DISTINCT ON processing occurs after ORDER BY sorting.)

SKY mobi
新手机 新应用 新娱乐

DISTINCT

- postgres=# select distinct on (id) id,name,crt_time from t1 ;
- id | name | crt_time
- ----+--------+--------------------
- 1 | digoal | 2012-04-26 08:15:09

- postgres=# select distinct on (id) id,name,crt_time from t1 order by crt_time;
- ERROR: SELECT DISTINCT ON expressions must match initial ORDER BY expressions
- LINE 1: select distinct on (id) id,name,crt_time from t1 order by cr...
- ^
- postgres=# select distinct on (id) id,name,crt_time from t1 order by id;
- id | name | crt_time
- ----+--------+--------------------
- 1 | digoal | 2012-04-26 08:15:09

SKY mobi
新手机 新应用 新娱乐

DISTINCT

- postgres=# select distinct on (id) id,name,crt_time from t1 order by id,crt_time;
- id | name | crt_time
- ----+--------+--------------------
- 1 | digoal | 2012-04-26 08:15:09

- postgres=# select distinct on (id) id,name,crt_time from t1 order by id,crt_time desc;
- id | name | crt_time
- ----+--------+--------------------
- 1 | DIGOAL | 2012-04-26 08:15:15

SKY mobi
新手机 新应用 新娱乐

DISTINCT

- postgres=# select distinct on (id,name) id,name,crt_time from t1 order by id,crt_time desc;
- ERROR: SELECT DISTINCT ON expressions must match initial ORDER BY expressions
- LINE 1: select distinct on (id,name) id,name,crt_time from t1 order ...
- ^
- postgres=# select distinct on (id,name) id,name,crt_time from t1 order by id,name,crt_time desc;
- id | name | crt_time
- ----+--------+---------------------
- 1 | DIGOAL | 2012-04-26 08:15:15
- 1 | digoal | 2012-04-26 08:15:09

SKY mobi
新手机 新应用 新娱乐

DISTINCT

- 使用DISTINCT ON实现用窗口函数实现的取第一名的功能

- postgres=# CREATE TABLE window_test(id int, name text, subject text, score numeric);
- postgres=# INSERT INTO window_test VALUES (1,'digoal','数学',99.5), (2,'digoal','语文',89.5),
- (3,'digoal','英语',79.5), (4,'digoal','物理',99.5), (5,'digoal','化学',98.5),
- (6,'刘德华','数学',89.5), (7,'刘德华','语文',99.5), (8,'刘德华','英语',79.5),
- (9,'刘德华','物理',89.5), (10,'刘德华','化学',69.5),
- (11,'张学友','数学',89.5), (12,'张学友','语文',91.5), (13,'张学友','英语',92.5),
- (14,'张学友','物理',93.5), (15,'张学友','化学',94.5);

- -- 取出每门课程的第一名.
- postgres=# select distinct on (subject) id,name,subject,score from window_test order by subject,score desc;

SKY mobi
新手机 新应用 新娱乐

DISTINCT

- id | name | subject | score
- ----+--------+---------+-------
- 5 | digoal | 化学 | 98.5
- 1 | digoal | 数学 | 99.5
- 4 | digoal | 物理 | 99.5
- 13 | 张学友 | 英语 | 92.5
- 7 | 刘德华 | 语文 | 99.5
- (5 rows)

- 与使用窗口函数得到的结果一致, 并且写法更简洁.

SKY mobi
新手机 新应用 新娱乐

COMBINING QUERY

- query1 UNION [ALL] query2
- query1 INTERSECT [ALL] query2
- query1 EXCEPT [ALL] query2

- UNION effectively appends the result of query2 to the result of query1 (although there is no guarantee that this is the order in which the rows are actually returned)

- INTERSECT returns all rows that are both in the result of query1 and in the result of query2.

- EXCEPT returns all rows that are in the result of query1 but not in the result of query2.

- Combining Query eliminates duplicate rows from its result, in the same way as DISTINCT, unless ALL is used
- query1 and query2 must return the same number of columns and the corresponding columns have compatible data types.

SKY mobi
新手机 新应用 新娱乐

SORT

- SELECT select_list
- FROM table_expression
- ORDER BY sort_expression1 [ASC | DESC] [NULLS { FIRST | LAST }]
- [, sort_expression2 [ASC | DESC] [NULLS { FIRST | LAST }] ...]

SKY mobi
新手机 新应用 新娱乐

LIMIT [OFFSET]

- SELECT select_list
- FROM table_expression
- [ORDER BY ...]
- [LIMIT { number | ALL }] [OFFSET number]

SKY mobi
新手机 新应用 新娱乐

函数三态简介

- 非常重要的函数三态
- Immutable
- Stable
- Volatile

- digoal=# create sequence seq;
- CREATE SEQUENCE
- digoal=# select provolatile from pg_proc where proname='nextval';
- provolatile
- ----------
- v
- (1 row)
- digoal=# select nextval('seq'::regclass) from generate_series(1,3);
- nextval
- ---------
- 1
- 2
- 3
- (3 rows)

SKY mobi
新手机 新应用 新娱乐

函数三态简介

- digoal=# alter function nextval(regclass) immutable;
- ALTER FUNCTION
- digoal=# select nextval('seq'::regclass) from generate_series(1,3);
- nextval
- ---------
- 4
- 4
- 4
- (3 rows)

SKY mobi
新手机 新应用 新娱乐

函数三态简介

- digoal=# alter function nextval(regclass) stable;
- ALTER FUNCTION
- digoal=# select nextval('seq'::regclass) from generate_series(1,3);
- nextval
- ---------
- 5
- 6
- 7
- (3 rows)

SKY mobi
新手机 新应用 新娱乐

函数三态简介

- A VOLATILE function can do anything, including modifying the database. It can return different results on successive calls with the same arguments. The optimizer makes no assumptions about the behavior of such functions. A query using a volatile function will re-evaluate the function at every row where its value is needed.

- A STABLE function cannot modify the database and is guaranteed to return the same results given the same arguments for all rows within a single statement. This category allows the optimizer to optimize multiple calls of the function to a single call. In particular, it is safe to use an expression containing such a function in an index scan condition. (Since an index scan will evaluate the comparison value only once, not once at each row, it is not valid to use a VOLATILE function in an index scan condition.)

- An IMMUTABLE function cannot modify the database and is guaranteed to return the same results given the same arguments forever. This category allows the optimizer to pre-evaluate the function when a query calls it with constant arguments. For example, a query like SELECT ... WHERE x = 2 + 2 can be simplified on sight to SELECT ... WHERE x = 4, because the function underlying the integer addition operator is marked IMMUTABLE.

SKY mobi
新手机 新应用 新娱乐

函数三态简介

- Stable和immutable的不同之处 :
- Stable函数在execute时执行.
- Immutable函数在plan时执行.

- 所以stable和immutable仅仅当使用planed query时才有区别.
- Planed query属于一次plan, 多次execute. Immutable在planed query中也只会执行一次. 而stable函数在这种场景中是会执行多次的.

- Immutable稳定性> stable> volatile

SKY mobi
新手机 新应用 新娱乐

函数三态简介

- 另一些需要注意的 :
- 函数内有多条SQL时, 例如
- digoal=# Create or replace function f1() returns setof int as $$
- declare
- v_id int;
- begin
- For v_id in select id from t loop
- return next v_id;
- End loop;
- -- 在这之间如果t发生了变更并提交了, volatile函数能看到这个变更.
- perform pg_sleep(10);
- For v_id in select id from t loop
- return next v_id;
- End loop;
- Return;
- End;
- $$ language plpgsql strict volatile;
- volatile处理每个SQL都是一个新的snapshot. Stable和immutable将调用函数时作为这个snapshot. (演示)

SKY mobi
新手机 新应用 新娱乐

函数三态简介

- 对于大运算量的函数, 且一条SQL如果返回多值, 并且传入的参数一致的情况下. 调用一次显然比调用多次耗费小.
- 实际使用过程中选择好三态是非常重要的.

SKY mobi
新手机 新应用 新娱乐

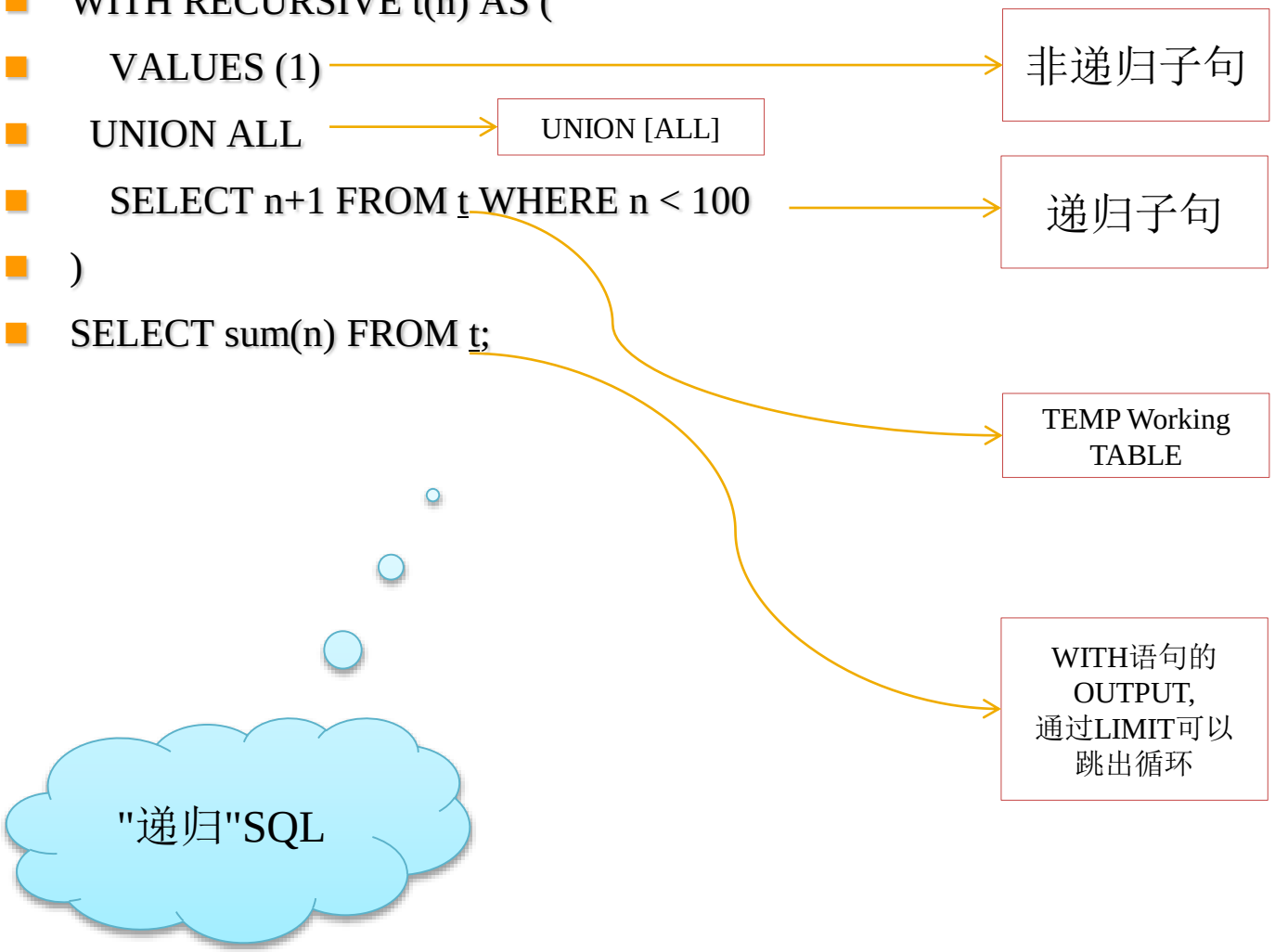
WITH(Common Table Expressions)

```
WITH regional_sales AS (
        SELECT region, SUM(amount) AS total_sales
        FROM orders
        GROUP BY region
    ), top_regions AS (
        SELECT region
        FROM regional_sales
        WHERE total_sales > (SELECT SUM(total_sales)/10 FROM regional_sales)
    )
SELECT region,
       product,
       SUM(quantity) AS product_units,
       SUM(amount) AS product_sales
FROM orders
WHERE region IN (SELECT region FROM top_regions)
GROUP BY region, product;
```

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

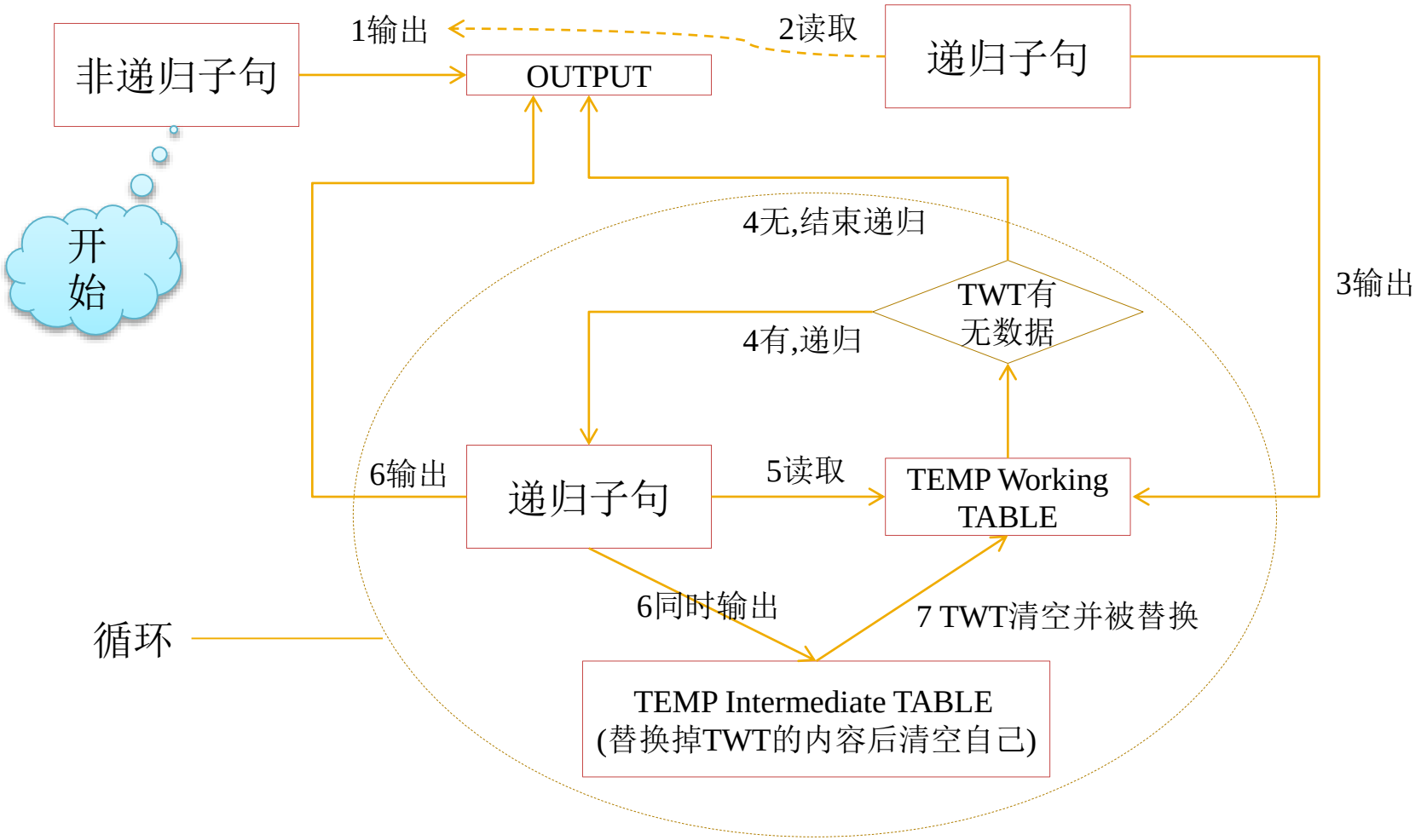
- WITH RECURSIVE t(n) AS (
- VALUES (1)
- UNION ALL
- SELECT n+1 FROM t WHERE n < 100
-)
- SELECT sum(n) FROM t;



SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- UNION ALL 去重复(去重复时NULL 视为等同)
- 图中所有输出都涉及UNION [ALL]的操作, 包含以往返回的记录和当前返回的记录



SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- TEMP Working Table 没有ctid, cmin, cmax, xmin, xmax, tableoid字段

- postgres=# create table test (id int,name text);
- postgres=# insert into test values (1,'digoal1'),(2,'digoal2');
- postgres=# begin;
- postgres=# with t1 as (update test set name='DIGOAL2' where id=2 returning *)
- select ctid from t1;
- ERROR: column "ctid" does not exist
- LINE 2: select ctid from t1;
- ^
- postgres=# rollback;

- 其他字段(cmin,cmax,xmin,xmax,tableoid)同样错误

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 递归查询输出产品部以及该部门的所有子部门信息.
- 然后输出各个子部门以及各个子部门的人数

```
WITH RECURSIVE included_parts(sub_part, part, quantity) AS (
    SELECT sub_part, part, quantity FROM parts WHERE part = '产品部'
  UNION ALL
    SELECT p.sub_part, p.part, p.quantity
    FROM included_parts pr, parts p
    WHERE p.part = pr.sub_part
  )
SELECT sub_part, SUM(quantity) as total_quantity
FROM included_parts
GROUP BY sub_part
```

初始输出

第一步时读取的是初始输出,
后面都是TEMP Working
TABLE

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 死循环

```
WITH RECURSIVE search_graph(id, link, data, depth) AS (
        SELECT g.id, g.link, g.data, 1
        FROM graph g
      UNION ALL
        SELECT g.id, g.link, g.data, sg.depth + 1
        FROM graph g, search_graph sg
        WHERE g.id = sg.link
)
SELECT * FROM search_graph;
```

每次递归输出的记录与以往的记录都不一样,
TEMP Working Table 永远都有记录,
因此无限循环.

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 规避上一个死循环的方法
- 让递归SQL有机会没记录输出

```
WITH RECURSIVE search_graph(id, link, data, depth, path, cycle) AS (
        SELECT g.id, g.link, g.data, 1,
          ARRAY[g.id],
        false
        FROM graph g
      UNION ALL
        SELECT g.id, g.link, g.data, sg.depth + 1,
          path || g.id,
        g.id = ANY(path)
        FROM graph g, search_graph sg
        WHERE g.id = sg.link AND NOT cycle
)
SELECT * FROM search_graph;
```

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 多值比较需使用ROW类型的ARRAY.

```
WITH RECURSIVE search_graph(id, link, data, depth, path, cycle) AS (
        SELECT g.id, g.link, g.data, 1,
          ARRAY[ROW(g.f1, g.f2)],
          false
        FROM graph g
      UNION ALL
        SELECT g.id, g.link, g.data, sg.depth + 1,
          path || ROW(g.f1, g.f2),
          ROW(g.f1, g.f2) = ANY(path)
        FROM graph g, search_graph sg
        WHERE g.id = sg.link AND NOT cycle
)
SELECT * FROM search_graph;
```

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 还有什么情况可以跳出循环

- WITH RECURSIVE t(n) AS (
- SELECT 1
- UNION ALL
- SELECT n+1 FROM t
-)
- SELECT n FROM t LIMIT 100;

- 注意如果t表在外围被join了然后再limit的. 还死循环
- 使用递归查询注意防止死循环

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 把属于产品部以及它的子部门的记录删除.

```
WITH RECURSIVE included_parts(sub_part, part) AS (
    SELECT sub_part, part FROM parts WHERE part = '产品部'
  UNION ALL
    SELECT p.sub_part, p.part
    FROM included_parts pr, parts p
    WHERE p.part = pr.sub_part
  )
DELETE FROM parts
  WHERE part IN (SELECT part FROM included_parts);
```

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- WITH的所有子句包括MAIN子句查看到的是一个SNAPSHOT.
- 各个子句对记录的变更相互看不到, 如果要看到变更的数据需使用RETURNING子句.

```
WITH t AS (
    UPDATE products SET price = price * 1.05 WHERE id = 10
    RETURNING *
)
SELECT * FROM products WHERE id = 10;
```

```
WITH t AS (
    UPDATE products SET price = price * 1.05 WHERE id = 10
    RETURNING *
)
SELECT * FROM t;
```

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 测试表
- postgres=# create table test (id int,name text);
- CREATE TABLE
- postgres=# insert into test values(1,'digoal1'),(2,'digoal2');

这样会看到老数据

- postgres=# with t1 as (update test set name='NEW' where id=2 returning *)
- postgres-# select * from test where name='NEW';
- id | name
- (0 rows)

这样才能看到新数据

- postgres=# with t1 as (update test set name='NEWNEW' where id=2 returning *)
- postgres-# select * from t1 where name='NEWNEW';
- id | name
- 2 | NEWNEW
- (1 row)

SKY mobi
新手机 新应用 新娱乐

WITH(Common Table Expressions)

- 避免类似SQL:
- postgres=# with t1 as (update test set name='digoal1' where id=1)
- postgres-# delete from test where id=1;
- DELETE 1
- postgres=# select * from test where id=1;
- id | name
- (0 rows)

这段测试和手册不符,
手册上表示update和delete子句同时针对一条记录操作时, delete子句不会执行.

- 避免类似SQL:
- postgres=# with t1 as (update test set name='DIGOAL2' where id=2)
- postgres-# update test set name='NEW' WHERE id=2;
- UPDATE 1
- postgres=# select * from test where id=2;
- id | name
- 2 | NEW

SKY mobi
新手机 新应用 新娱乐

练习

- psql的使用熟悉
- 数据库基本操作练习, 例如建表, 表的操作, 建立测试模型, 使用pgbench测试.