PostgreSQL

The world's most advanced open source database.

Estatística, monitoramento e diagnóstico através do catálogo do PostgreSQL

pgbr 2011 Conferência Brasileira PostgreSQL 3 e 4 novembro São Paulo - SP

PostgreSQL

The world's most advanced open source database.

O que é o catálogo?

Os catálogos são os locais onde os SGBDs armazenam os metadados do esquema, tais como informações sobre tabelas, colunas, e informações de controle interno.

É uma excelente fonte de informações, mas nem sempre é fácil de encontrar.

The world's most advanced open source database.

Porque conhecer o catálogo é importante?

Resumidamente, o catálogo pode ser descrito como um banco de dados que contém informações sobre o seu banco de dados e o seu servidor. Portanto ele pode ser um grande aliado tanto no desenvolvimento quanto na manutenção e monitoramento de uma base de dados

The world's most advanced open source database.

Information Schema

O esquema de informações consiste em um conjunto de visões contendo informações sobre os objetos definidos no banco de dados corrente. O esquema de informações é definido no padrão SQL e, portanto, pode-se esperar que seja portável e permaneça estável — isto é diferente dos catálogos do sistema, que são específicos do PostgreSQL e modelados segundo interesses da implementação. Entretanto, as visões do esquema de informações não contêm informações sobre funcionalidades específicas do PostgreSQL; para obter este tipo de informação, é necessário consultar os catálogos do sistema e outras visões específicas do PostgreSQL.

PostgreSQL

The world's most advanced open source database.

System Catalogs

Os catálogos do sistema do PostgreSQL são tabelas comuns. Estas tabelas podem ser removidas e recriadas, podem ser adicionadas colunas, podem ser inseridos e atualizados valores, mas isso não é recomendável. Normalmente, os catálogos do sistema não devem ser modificados manualmente, sempre existe um comando SQL para fazê-lo (Por exemplo, CREATE DATABASE insere uma linha no catálogo pg_database — e cria realmente o banco de dados no disco).

Information Schema – Postgresql 8.4 / 9.1

PostgreSQL

The world's most advanced open source database.

www.postgresql.org/docs/8.4/ir

Chapter 33. The Information Schema

Table of Contents

www.postgresql.org/docs/9.1/interactive/information-schema.html

Chapter 34. The Information Schema

Table of Contents

System Catalogs – Postgresql 8.4 / 9.1

PostgreSQL

The world's most advanced open source database.

www.postgresql.org/docs/8.4

Table of Contents

www.postgresql.org/docs/9.1/interactive/catalogs.html

Table of Contents

The world's most advanced open source database.

pg_catalog – Tabelas

pg_class - Contém informações de outras tabelas e objetos, é uma das principais tabelas do catálogo, geralmente buscamos informações dela diretamente ou atráves de views do próprio Postgresql, o que vem sendo cada vez mais comum.

pg_database - Armazena informações sobre os bancos de dados disponíveis.
Diferente da maioria dos catálogos do sistema, é compartilhado por todos os bancos de dados do cluster.

pg_extension - Armazena informações sobre as extensões instaladas

pg_language – Armazena as linguagens instaladas que podem ser usadas para criar funções

pg_proc - O catálogo pg_proc armazena informações sobre as funções (ou procedimentos).

pg_statistic - Armazena dados estatísticos sobre o conteúdo do banco de dados. Entradas são criadas por ANALYZE e posteriormente utilizado pelo planejador de comandos

pg_tablespace - Armazena informações sobre os tablespaces disponíveis, é compartilhado entre todas as bases de um cluster

pg_catalog – System Views

Complementando o catálogo do PostgreSQL existem as system views que fornecem acesso rápido e fácil a diversas informações sobre o banco de dados e também ao estado interno do servidor.

PostgreSQL

The world's most advanced open source database.

7.4 - 8.0

View Name	Purpose
pg_indexes	indexes
pg_locks	currently held locks
pg_rules	rules
pg_settings	parameter settings
pg_stats	planner statistics
pg_tables	tables
pg_user	database users
pg_views	views

8.1

View Name	Purpose
pg_group	groups of database users
pg_indexes	indexes
pg_locks	currently held locks
pg_prepared_xacts	currently prepared transactions
pg_roles	database roles
pg_rules	rules
pg_settings	parameter settings
pg_shadow	database users
pg_stats	planner statistics
pg_tables	tables
pg_user	database users
pg_views	views

8.2 - 8.3

View Name	Purpose
pg_cursors	open cursors
pg_group	groups of database users
pg_indexes	indexes
pg_locks	currently held locks
pg_prepared_statements	prepared statements
pg_prepared_xacts	prepared transactions
pg_roles	database roles
pg_rules	rules
pg_settings	parameter settings
pg_shadow	database users
pg_stats	planner statistics
pg_tables	tables
pg_timezone_abbrevs	time zone abbreviations
pg_timezone_names	time zone names
pg_user	database users
pg_views	views

8.4 - 9.0

View Name	Purpose
pg_cursors	open cursors
pg_group	groups of database users
pg_indexes	indexes
pg_locks	currently held locks
pg_prepared_statements	prepared statements
pg_prepared_xacts	prepared transactions
pg_roles	database roles
pg_rules	rules
pg_settings	parameter settings
pg_shadow	database users
pg_stats	planner statistics
pg_tables	tables
pg_timezone_abbrevs	time zone abbreviations
pg_timezone_names	time zone names
pg_user	database users
pg_user_mappings	user mappings
pg_views	views

9.1

View Name	Purpose
pg_available_extensions	available extensions
pg_available_extension_versions	available versions of extensions
pg_cursors	open cursors
pg_group	groups of database users
pg_indexes	indexes
pg_locks	currently held locks
pg_prepared_statements	prepared statements
pg_prepared_xacts	prepared transactions
pg_roles	database roles
pg_rules	rules
pg_seclabels	security labels
pg_settings	parameter settings
pg_shadow	database users
pg_stats	planner statistics
pg_tables	tables
pg_timezone_abbrevs	time zone abbreviations
pg_timezone_names	time zone names
pg_user	database users
pg_user_mappings	user mappings
pg_views	views

The world's most advanced open source database.

View Name	Purpose
pg_available_extensions	available extensions
pg_available_extension_versions	available versions of extensions
pg_cursors	open cursors
pg_group	groups of database users
pg_indexes	indexes
pg_locks	currently held locks
pg_prepared_statements	prepared statements
pg_prepared_xacts	prepared transactions
pg_roles	database roles
pg_rules	rules
pg_seclabels	security labels
pg_settings	parameter settings
pg_shadow	database users
pg_stats	planner statistics
pg_tables	tables
pg_timezone_abbrevs	time zone abbreviations
pg_timezone_names	time zone names
pg_user	database users
pg_user_mappings	user mappings
pg_views	views

Coletor de Estatísticas

O coletor de estatísticas do PostgreSQL é um subsistema de apoio a coleta e relatório de informações sobre as atividades do servidor.

postgresql.conf

```
# RUNTIME STATISTICS
#------------------------------------------------------------------------------

# - Query/Index Statistics Collector -

#track_activities = on
#track_counts = on
track_functions = pl                    # none, pl, all
#track_activity_query_size = 1024       # (change requires restart)
#update_process_title = on
#stats_temp_directory = 'pg_stat_tmp' [1]

# - Statistics Monitoring - [2]

#log_parser_stats = off
#log_planner_stats = off
#log_executor_stats = off
#log_statement_stats = off
```

[1] - File System em RAM diminui o impacto de I/O
[2] - Informações detalhas após cada execução

Views pré-definidas Coletor de Estatísticas – PostgreSQL 9.1

PostgreSQL

The world's most advanced open source database.

pg_stat_activity
pg_stat_bgwriter
pg_stat_database
pg_stat_database_conflicts
pg_stat_replication
pg_stat_all_tables
pg_stat_sys_tables
pg_stat_user_tables
pg_stat_xact_all_tables
pg_stat_xact_sys_tables
pg_stat_xact_user_tables
pg_stat_all_indexes
pg_stat_sys_indexes
pg_stat_user_indexes
pg_statio_all_tables
pg_statio_sys_tables
pg_statio_user_tables
pg_statio_all_indexes
pg_statio_sys_indexes
pg_statio_user_indexes
pg_statio_all_sequences
pg_statio_sys_sequences
pg_statio_user_sequences
pg_stat_user_functions
pg_stat_xact_user_functions

#http://www.postgresql.org/docs/9.1/interactive/monitoring-stats.html#MONITORING-STATS-VIEWS-TABLE

PostgreSQL

The world's most advanced open source database.

As views pg_statio* são úteis para determinar a eficácia de uso do cache, quando o número de blks_read (disk reads) é menor do que blks_hit (buffer hits) o uso do cache é satisfatório, no entanto essas estatísticas não devem servir como única medida para o acerto ideal do sistema.

O uso de ferramentas do sistema operacional, estatísticas do servidor e medição de cache de I/O tanto de controladoras de disco quanto do próprio kernel devem ser levados em conta.

PostgreSQL

The world's most advanced open source database.

Além das views também podemos obter informações do coletor de estatísticas através de funções pré definidas.

Uma relação de todas as funções é listada na documentação.

http://www.postgresql.org/docs/9.1/interactive/monitoring-stats.html#MONITORING-STATS-FUNCS-TABLE

The world's most advanced open source database.

Exemplos, dicas e uso prático do catálogo.

Versão utilizada PostgreSQL 9.1!

PostgreSQL

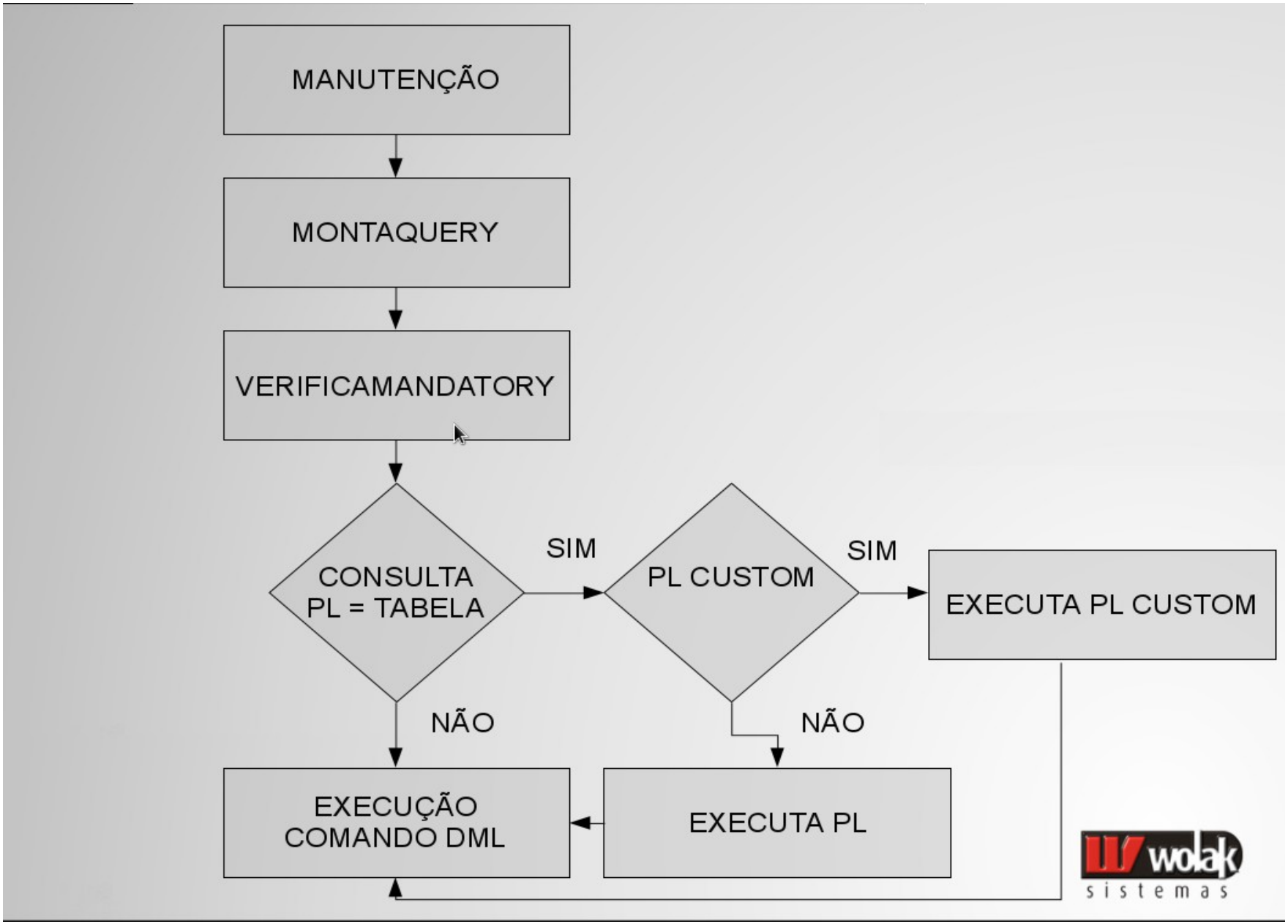
The world's most advanced open source database.

Exemplo de informação usada no desenvolvimento:

Exemplo 1: SQL1.sql

PostgreSQL

The world's most advanced open source database.



Front-End chama uma PL passando um array com campo e conteúdo.

```
SELECT  pk as
pknotafiscalentrada,mandatory,mandatorymensagem,mensagem
FROM fnc.manutencao(
1, -- pempressa
2, -- pfilial
3, -- punidade
0, -- pusuario
array['notafiscalentrada','update','NotaFiscalEntrada_Form',
'MODIF'], -- pacao
ARRAY['fknotafiscalentradaxml' , '84','fkordemcompra' ,
NULL,'fktipooperacao_naturezaoperacao' ,
'22','fknaturezaoperacao' , '130','fkcadastro_documento' ,
'251','fkcadastro' , '244','fkcadastro_vinculo' ,
'213','fktipooperacao' , '11','fkdocumentofiscalicms' ,
'30','fkseriedocumentofiscal' , '2','numero' ,
'238085','dataemissao' , '2011-10-13','dataentradasaida' ,
'2011-10-26 10:43:00.000','valorbasecalculoicms' ,
'4528.34','valoricms' ,
'543.4','valorbasecalculoicmssubstituicaotributaria' ,
```

The world's most advanced open source database.

```
CREATE OR REPLACE FUNCTION fnc.manutencao(
    pempresa           INTEGER,
    pfilial            INTEGER,
    punidade           INTEGER,
    pusuario           INTEGER,
    pcacao             VARCHAR[],
    pccampoconteudo    VARCHAR[],
    pccampoconteudopk VARCHAR[])
RETURNS fnc.manutencao_retorno AS
```

Pesquisando o catalogo "pg_proc" para verificar se existe uma PL com o nome da tabela que foi passada como parâmetro, caso exista, a PL manutenção executa uma outra PL com o mesmo nome do parâmetro passada em fnc.manutencao.

```
IF (SELECT TRUE FROM pg_proc JOIN pg_namespace ON nspname = 'fnc'
AND pg_namespace.oid = pg_proc.pronamespace
     WHERE proname = LOWER(pcAcao[1]) LIMIT 1) = TRUE AND
UPPER(pcAcao[2]) IN('UPDATE','UPDATE_ONLY','DELETE','LOCK') THEN
    IF (rFuncaoCustom.nometabela IS NOT NULL AND
rFuncaoCustom.bloquearfnc = 2) OR (rFuncaoCustom.nometabela IS
NULL) THEN
        FOR rResultadoFuncaoBefore IN EXECUTE 'SELECT fnc.'||
pcAcao[1]||'('||QUOTE_LITERAL('BEFORE')||','||
pcCampoConteudoPK[2]::integer||','||
QUOTE_LITERAL(ARRAY_TO_STRING(pcAcao,','))||') AS mensagem'
        LOOP
            cResultadoFuncaoBefore =
rResultadoFuncaoBefore.mensagem;
        END LOOP;
    END IF;
END IF;
```

```
Executa a funcao fnc.montaquery para escrever o comando DML;

IF UPPER(pcAcao[2]) <> 'LOCK' THEN
    FOR rResultado IN EXECUTE
fnc.montaquery(pEmpresa,pFilial,pUnidade,pUsuario,pcAcao[2],pcAcao[
1],pcCampoConteudo,pcCampoConteudoPK) LOOP END LOOP;
END IF;
```

Verifica se o campo é nulo, caso não seja busca a propriedade DEFAULT da coluna, monta a coluna com a sua propriedade original de comando.

```
cCampoConteudo = pcCampoConteudo;
FOR i IN 2..ARRAY_UPPER(pcCampoConteudo,1) BY 2
LOOP
    SELECT INTO cDefault,cIsNull column_default,is_nullable FROM
information_schema.columns WHERE table_name = QUOTE_IDENT(pcTabela)
AND column_name = QUOTE_IDENT(pcCampoConteudo[i-1]);
    IF cIsNull = 'NO' AND pcCampoConteudo[i] IS NULL THEN
        cCampoConteudo[i] = cDefault;
    END IF;
    pcCampoConteudoAlterado[i] =
QUOTE_LITERAL(TRIM(fnc.extraiconteudo(pcCampoConteudo[i-
1],cCampoConteudo,NULL)))||COALESCE('::'||
    (SELECT UPPER(udt_name) FROM information_schema.columns WHERE
table_name = QUOTE_IDENT(pcTabela) AND column_name =
QUOTE_IDENT(pcCampoConteudo[i-1])),'');
END LOOP;
```

PostgreSQL

The world's most advanced open source database.

Resultado da função MONTAQUERY.

```
UPDATE notafiscalentrada SET
controle.fkusuarioregistrobloqueado =
NULL,controle.datainativo = NULL,controle.confirmainclusao =
NULL,controle.fkusuarioinclusao = 0,controle.datainclusao =
'2011-10-26 10:43:33.734',controle.fkusuarioalteracao =
0,controle.dataalteracao = '2011-10-29
22:04:03.514',fknotafiscalentradaxml =
'84'::INT4,fkordemcompra =
NULL,fktipooperacao_naturezaoperacao =
'22'::INT4,fknaturezaoperacao =
'130'::INT4,fkcadastro_documento = '251'::INT4,fkcadastro =
…
…

WHERE pknotafiscalentrada = '148'::INT4 RETURNING
pknotafiscalentrada AS pk,0 AS inclusao;
```

PostgreSQL

The world's most advanced open source database.

Exemplo de informação usada no monitoramento:

Exemplo 2: SQL2.sql

```
SELECT * FROM pg_stat_activity;

SELECT datname,usename,client_addr,waiting,current_query
FROM pg_stat_activity ;

SELECT datname,usename,client_addr,waiting,current_query
FROM pg_stat_activity WHERE current_query != '<IDLE>' ;

SELECT (current_timestamp - query_start) as tempoquery,
datname, usename,waiting,current_query
FROM pg_stat_activity WHERE current_query != '<IDLE>'
ORDER BY 1 DESC;
```

PostgreSQL

The world's most advanced open source database.

Exemplo de informação usada no monitoramento (Bloqueios)

Exemplo 3: SQL3.sql

```
SELECT locktype, pg_database.datname, pg_class.relname, transactionid,
      classid, objid, objsubid, virtualtransaction, pid, mode, granted
FROM pg_locks
JOIN pg_database ON pg_database.oid = pg_locks.database
JOIN pg_class ON pg_class.oid = pg_locks.relation

SELECT
esperando.current_query AS query_esperando,
esperando.procpid AS pid_esperando,
esperando.usename AS user_esperando,
parado.current_query AS query_parado,
parado.procpid AS pid_parado,
parado.usename AS user_parado,
tabelas_travadas.schemaname||'.'||tabelas_travadas.relname AS tabelas_travadas
FROM pg_stat_activity esperando
JOIN pg_locks locks_esperando ON locks_esperando.pid = esperando.procpid
JOIN pg_locks locks_parado ON locks_parado.relation = locks_esperando.relation
JOIN pg_stat_activity parado ON parado.procpid = locks_parado.pid
JOIN pg_stat_user_tables tabelas_travadas ON tabelas_travadas.relid =
locks_esperando.relation
WHERE esperando.waiting IS TRUE
AND NOT locks_esperando.granted
AND locks_parado.granted IS TRUE
```

PostgreSQL

The world's most advanced open source database.

Exemplo de informação sobre tamanho do banco e das tabelas:

Exemplo 4: SQL4.sql

```
--Tamanho dos bancos
SELECT pg_database.datname,
pg_size_pretty(pg_database_size(pg_database.datname))
FROM pg_database
ORDER BY pg_database_size(pg_database.datname) DESC,
pg_database.datname;

--Tamanho das tabelas do banco atual
SELECT retorno.schemaname, retorno.tablename,
pg_size_pretty(pg_total_relation_size(schemaname||'.'||tablename)) AS
tamanho
FROM (SELECT tablename, schemaname
        FROM pg_tables
       WHERE schemaname NOT IN ('pg_catalog', 'information_schema',
'pg_toast') ) retorno
 ORDER BY pg_total_relation_size(schemaname||'.'||tablename) DESC
```

PostgreSQL
POWERED

The world's most advanced open source database.

Exemplo de informação de varreduras nas tabelas

Exemplo 5: SQL5.sql

```
--Leituras Sequencias/Indexadas
SELECT schemaname,relname,seq_scan,seq_tup_read,idx_scan,idx_tup_fetch
FROM pg_stat_user_tables
WHERE (idx_scan + seq_scan) <> 0
ORDER BY seq_scan DESC

--Tabela que tiveram maiores indíces de Insert/Update/Delete
SELECT relname,n_tup_ins AS insert, n_tup_upd AS update, n_tup_del AS delete
FROM pg_stat_user_tables
--ORDER BY relname;
--ORDER BY n_tup_ins
--ORDER BY n_tup_upd
ORDER BY n_tup_del
DESC

--Tabelas que tiveram maiores buscas no cache ou no disco/kernel
SELECT relname,heap_blks_hit,heap_blks_read FROM pg_statio_user_tables
WHERE (heap_blks_hit + heap_blks_read)<>0
ORDER BY heap_blks_hit
--ORDER BY heap_blks_read
DESC
```

http://www.postgresql.org/docs/9.1/interactive/monitoring-stats.html#MONITORING-STATS-VIEWS-TABLE

PostgreSQL
POWERED

The world's most advanced open source database.

Exemplo de informação de funções (PL)

Exemplo 6: SQL6.sql

```
SELECT
pg_proc.proname,
pg_proc.pronargs,
pg_stat_user_functions.*
FROM pg_stat_user_functions
JOIN pg_proc ON pg_proc.oid = pg_stat_user_functions.funcid
```

postgresql.conf
track_functions = pl # none, pl, all

Exemplo de informação dos índices

SELECT * FROM pg_stat_user_indexes;

idx_tup_read – Número de varreduras por index scan
idx_tup_fetch – Número de linhas buscadas por index scan usando o índice

Leituras bitmap scan não atualizam a coluna idx_tup_fetch, geralmente porque nessas condições dois ou mais índices são usados, avaliados e combinados, portanto é difícil eleger qual o índice que realmente foi o “responsável” pelo retorno da informação, por isso apenas a coluna idx_tup_read é incrementada, portanto valores menores em idx_tup_fetch são normais

Considere idx_tup_read como a melhor alternativa em caso de comparações entre índices.

PostgreSQL

The world's most advanced open source database.

Exemplo de informação dos índices

Exemplo 7: SQL7.sql

```
SELECT indexrelname,idx_scan
FROM pg_stat_user_indexes
--WHERE relname='tabela'
WHERE idx_scan = 0
AND indexrelname NOT LIKE '%pk%' AND indexrelname NOT LIKE
'%uk%' --Se houver padrão de nome de chave pode utilizar para não
buscar UKs e PKs
ORDER BY idx_scan DESC

SELECT schemaname, relname, indexrelname, idx_scan,
indnatts AS numerodecolunas
FROM pg_stat_user_indexes
JOIN pg_index ON pg_index.indexrelid =
pg_stat_user_indexes.indexrelid
WHERE pg_index.indisunique IS FALSE --Não seleciona índices
únicos
AND indisprimary IS FALSE --Não seleciona chaves primárias
--ORDER BY idx_scan,relname
ORDER BY idx_scan DESC
```

PostgreSQL

The world's most advanced open source database.

Exemplo de informação dos índices

Exemplo 8: SQL8.sql

Verificando o uso do cache nas varreduras de índices

```
SELECT relname,indexrelname,idx_blks_hit,idx_blks_read
FROM pg_statio_user_indexes
JOIN pg_index ON pg_index.indexrelid =
pg_statio_user_indexes.indexrelid
WHERE (idx_blks_hit + idx_blks_read)<> 0
AND pg_index.indisunique IS FALSE
AND indisprimary IS FALSE
```

PostgreSQL
POWERED

The world's most advanced open source database.

Exemplo de informação dos índices

Desabilitando um índice

```
UPDATE pg_index
SET indisvalid = false
WHERE indexrelid = ?
```

Não faça isso para índices únicos!

O índice continua sendo atualizado por INSERT e UPDATE mas não será utilizado em nenhum consulta.

Desta maneira é possível testar índices antes de remover.

PostgreSQL
POWERED

The world's most advanced open source database.

Ferramentas de monitoramento para PostgreSQL

Existem várias ferramentas disponíveis no mercado, tanto livres quanto proprietárias.

O uso delas aliado ao conhecimento do catálogo e das estruturas internas do PostgreSQL facilita a tarefa de administrar uma ou várias bases de dados, segue uma lista resumida das ferramentas mais comuns.

The world's most advanced open source database.

Zabbix
Pgwatch
PgFouine
Collectd
Cedrus

The world's most advanced open source database.

Perguntas?

The world's most advanced open source database.

Muito Obrigado!

Fabiano Machado Dias
fabiano@wolaksistemas.com.br

Eduardo Wolak
wolak@wolaksistemas.com.br

www.wolaksistemas.com.br
www.erpws.com.br