PERFORMANCE E PRODUTIVIDADE

2 HABILIDADES PARA TRANSFORMAR SEU RELACIONAMENTO COM BANCO DE DADOS

Rodrigo Moutinho
@rcmoutinho

Dani Monteiro
@DaniMonteiroDBA

#TDCSP

Agenda

- Hackeando sua consulta
- Conhecendo os recursos disponíveis
- Consultas avançadas?
- Automatizando seu ambiente
- DEMO!
- E o NoSQL?
- OUTRA DEMO!

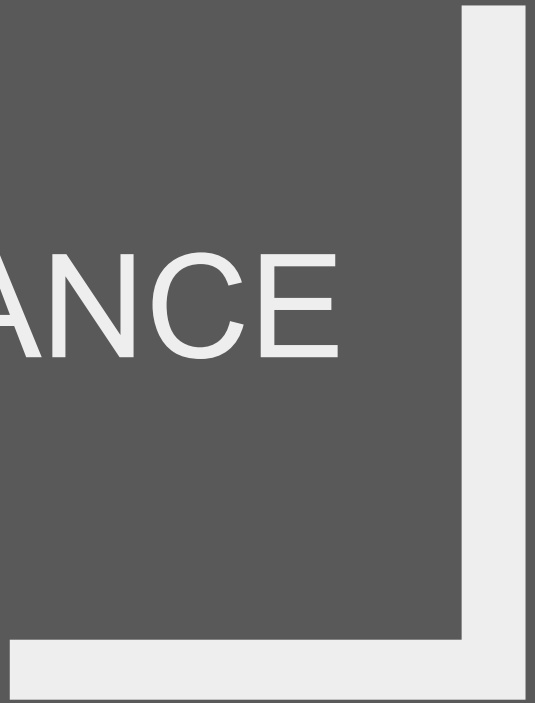
Exemplo Real

- ERP TOTVS
- Fábrica de Camisa Esportiva
- O que tenho para liberar no meu estoque?
- Processo diário de mais de duas horas
  (um clique)

O Problema…

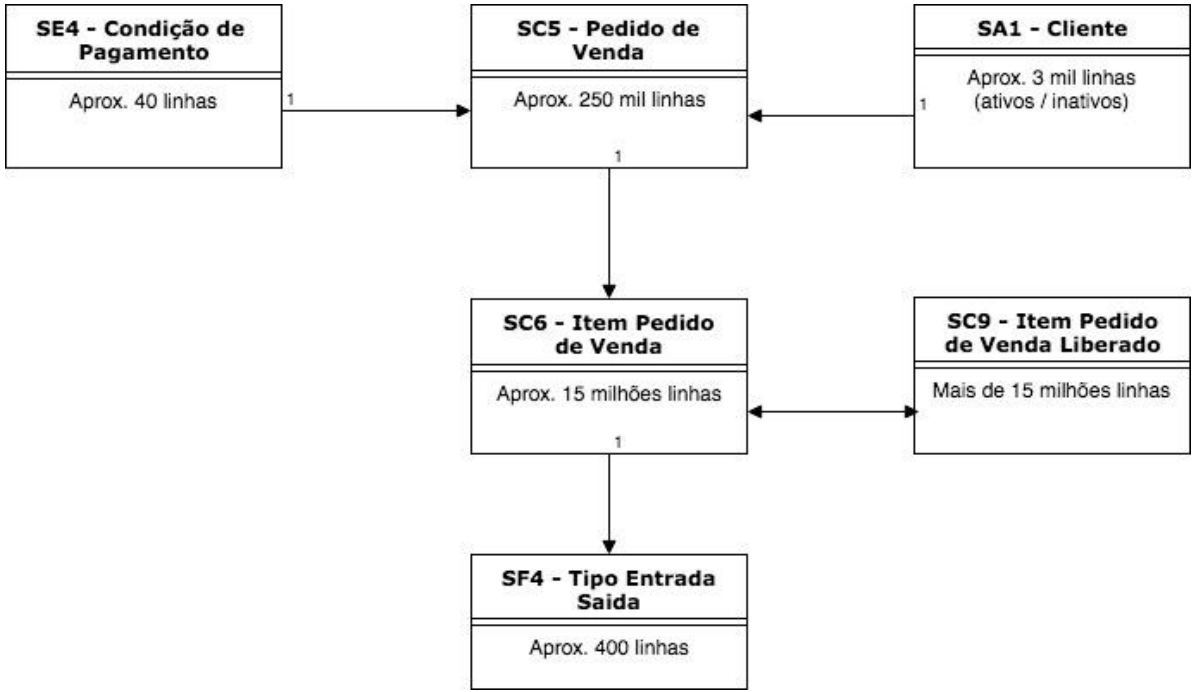
```
SELECT
  SC5.C5_NUM  AS PED,
  SC5.C5_VEND1  AS REPRES,
  SA1.A1_NOME   AS NOME,
  (
    SELECT sum ((C6_QTDVEN-C6_QTDENT))
    FROM SC6010 C6, SC9010 C9
    WHERE C6_NUM = SC5.C5_NUM
    AND C6_NUM = C9_PEDIDO
    AND C6.C6_PRODUTO = C9.C9_PRODUTO
    AND C9.C9_NFISCAL = ''
    AND C9.C9_XLIST = ''
    AND C6.D_E_L_E_T_ = ''
    AND C9.D_E_L_E_T_ = ''
    AND C6_BLQ <> 'R'
    AND C6_LOCAL = '05'
    AND C9_LOCAL = '05'
  ) AS QTDFAL,
  (
    SELECT sum ((C6_QTDVEN-C6_QTDENT) * C6_PRCVEN)
    FROM SC6010 C6, SC9010 C9
    WHERE C6_NUM = SC5.C5_NUM
    AND C6_NUM = C9_PEDIDO
    AND C6.C6_PRODUTO = C9.C9_PRODUTO
    AND C9.C9_NFISCAL = ''
    AND C9.C9_XLIST = ''
    AND C6.D_E_L_E_T_ = ''
    AND C9.D_E_L_E_T_ = ''
    AND C6_BLQ <> 'R'
    AND C6_LOCAL = '05'
    AND C9_LOCAL = '05'
  ) AS VALFAL,
  (
    SELECT sum (((C6_QTDVEN-C6_QTDENT) * C6_PRCVEN)-(C6_XQTD05*C6_PRCVEN))
    FROM SC6010 C6, SC9010 C9
    WHERE C6_NUM = SC5.C5_NUM
    AND C6_NUM = C9_PEDIDO
    AND C6.C6_PRODUTO = C9.C9_PRODUTO
    AND C9.C9_NFISCAL = ''
    AND C9.C9_XLIST = ''
    AND C6.D_E_L_E_T_ = ''
    AND C9.D_E_L_E_T_ = ''
    AND C6_BLQ <> 'R'
    AND C6_LOCAL = '05'
    AND C9_LOCAL = '05'
  ) AS VALOR,
  SE4.E4_COND AS COND,
  SC5.C5_OBSPED AS OBS,
  SC5.C5_PREVIST as PREV,
  SC5.C5_XST05 AS ST,
  SC5.C5_XFRET AS FRET,
  (
    SELECT SUM(C6_VALOR) FROM SC6010 C6, SC9010 C9
    WHERE C6_NUM = SC5.C5_NUM
    AND C6.C6_NUM = C9.C9_PEDIDO
    AND C6.C6_PRODUTO = C9.C9_PRODUTO
    AND C9.C9_NFISCAL = ''
    AND C9.C9_XLIST = ''
    AND C6.D_E_L_E_T_ = ''
    AND C9.D_E_L_E_T_ = ''
    AND C6.C6_LOCAL = '05'
    AND C6.C6_BLQ <> 'R'
  ) AS TOT
FROM SC5010 SC5,SC6010 SC6, SC9010 SC9,SA1010 SA1,SF4010 SF4,SE4010 SE4
WHERE SA1.A1_FILIAL = ' '
AND SE4.E4_FILIAL = ' '
AND SC5.C5_FILIAL = SC6.C6_FILIAL
AND SF4.F4_FILIAL = ' '
AND (SC5.C5_XFLAG = 'PE' OR SC5.C5_XFLAG = 'PAR' OR SC5.C5_XFLAG = 'LIB')
AND (SC5.C5_XSEP ='' or SC5.C5_XSEP ='1')
AND (SC6.C6_XFLAG = 'PE' or SC6.C6_LOCAL = '05' )
AND SC5.C5_NOTA = ''
AND SC5.C5_NUM = SC6.C6_NUM
AND SC6.C6_NUM = SC9.C9_PEDIDO
AND SC5.C5_NUM = SC9.C9_PEDIDO
AND SC6.C6_PRODUTO = SC9.C9_PRODUTO
AND SC9.C9_XLIST = ''
AND SC9.C9_NFISCAL = ''
AND SC5.C5_CLIENT = SA1.A1_COD
AND SC6.C6_BLQ <> 'R'
AND SC5.C5_TIPO = 'N'
AND SC5.C5_CONDPAG = SE4.E4_CODIGO
AND SC6.C6_TES = SF4.F4_CODIGO
AND SF4.F4_ESTOQUE = 'S'
AND SC6.C6_LOCAL = '05'
AND (SC6.C6_QTDVEN-SC6.C6_QTDENT) > 0
AND SC9.D_E_L_E_T_ = ''
AND SC5.D_E_L_E_T_ = ''
AND SC6.D_E_L_E_T_ = ''
AND SF4.D_E_L_E_T_ = ''
AND SE4.D_E_L_E_T_ = ''
AND SA1.D_E_L_E_T_ = ''
AND NOT EXISTS (
SELECT 1
FROM SC9010
WHERE C9_FILIAL = ' '
AND C9_PEDIDO = SC5.C5_NUM
AND D_E_L_E_T_ = ''
AND C9_BLCRED <> ''
AND C9_NFISCAL = ''
AND C9_BLCRED <> '10'
)
GROUP BY
SC5.C5_NUM,
SC5.C5_VEND1,
SA1.A1_NOME,
SE4.E4_COND ,
SC5.C5_OBSPED,
SC5.C5_PREVIST,
SC5.C5_XST05,
SC5.C5_XFRET
ORDER BY SA1.A1_NOME;
```

1# PERFORMANCE



Hackeando sua consulta

- Tabelas e relacionamentos
- Índices disponíveis
- Volume de dados de cada tabela consultada



Recursos disponíveis vs Níveis de permissão

- Subqueries
- Views
- Materialized Views
- Functions
- **Common Table Expressions (CTE)**

- **SELECT** ...
- CREATE VIEW ...
- CREATE MATERIALIZED VIEW ...
- CREATE OR REPLACE FUNCTION ...
- WITH ... **SELECT** ...

Subqueries

```
SELECT col1
FROM tab1
WHERE EXISTS(
   SELECT 1
   FROM tab2
   WHERE col2 = tab1.col2
);
```

PostgreSQL
Documentation

Views

(PostgreSQL 7.1+)

```
CREATE VIEW comedies AS
   SELECT *
   FROM films
   WHERE kind = 'Comedy';
```

PostgreSQL
Documentation

Materialized Views

(PostgreSQL 9.3+)

```
CREATE MATERIALIZED VIEW sales_summary AS
  SELECT
    seller_no,
    invoice_date,
    sum(invoice_amt)::numeric(13,2) as sales_amt
  FROM invoice
  WHERE invoice_date < CURRENT_DATE
  GROUP BY seller_no, invoice_date
  ORDER BY seller_no, invoice_date;
```

PostgreSQL Documentation

Materialized Views
(PostgreSQL 9.3+)

```
-- Índices
CREATE UNIQUE INDEX sales_summary_seller  ON
sales_summary (seller_no, invoice_date);

-- Atualizar os dados
REFRESH MATERIALIZED VIEW sales_summary;
```

PostgreSQL
Documentation

Functions
(PostgreSQL 7.1+)

```
CREATE OR REPLACE FUNCTION increment(i integer)
RETURNS integer AS $$
  BEGIN
    RETURN i + 1;
  END;
$$ LANGUAGE plpgsql;
```

PostgreSQL
Documentation

Common Table Expressions

(PostgreSQL 8.4+)

```
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;
```

PostgreSQL Documentation

Common Table Expressions

(PostgreSQL 8.4+)

```
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;
```

PostgreSQL Documentation

Common Table Expressions

(PostgreSQL 8.4+)

```
WITH RECURSIVE included_parts(sub_part, part, quantity) AS (
    SELECT sub_part, part, quantity
    FROM parts
    WHERE part = 'our_product'
  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
```

PostgreSQL Documentation

Common Table Expressions

(PostgreSQL 8.4+)

```
WITH RECURSIVE included_parts(sub_part, part, quantity) AS (
    SELECT sub_part, part, quantity
    FROM parts
    WHERE part = 'our_product'
  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
```

PostgreSQL Documentation

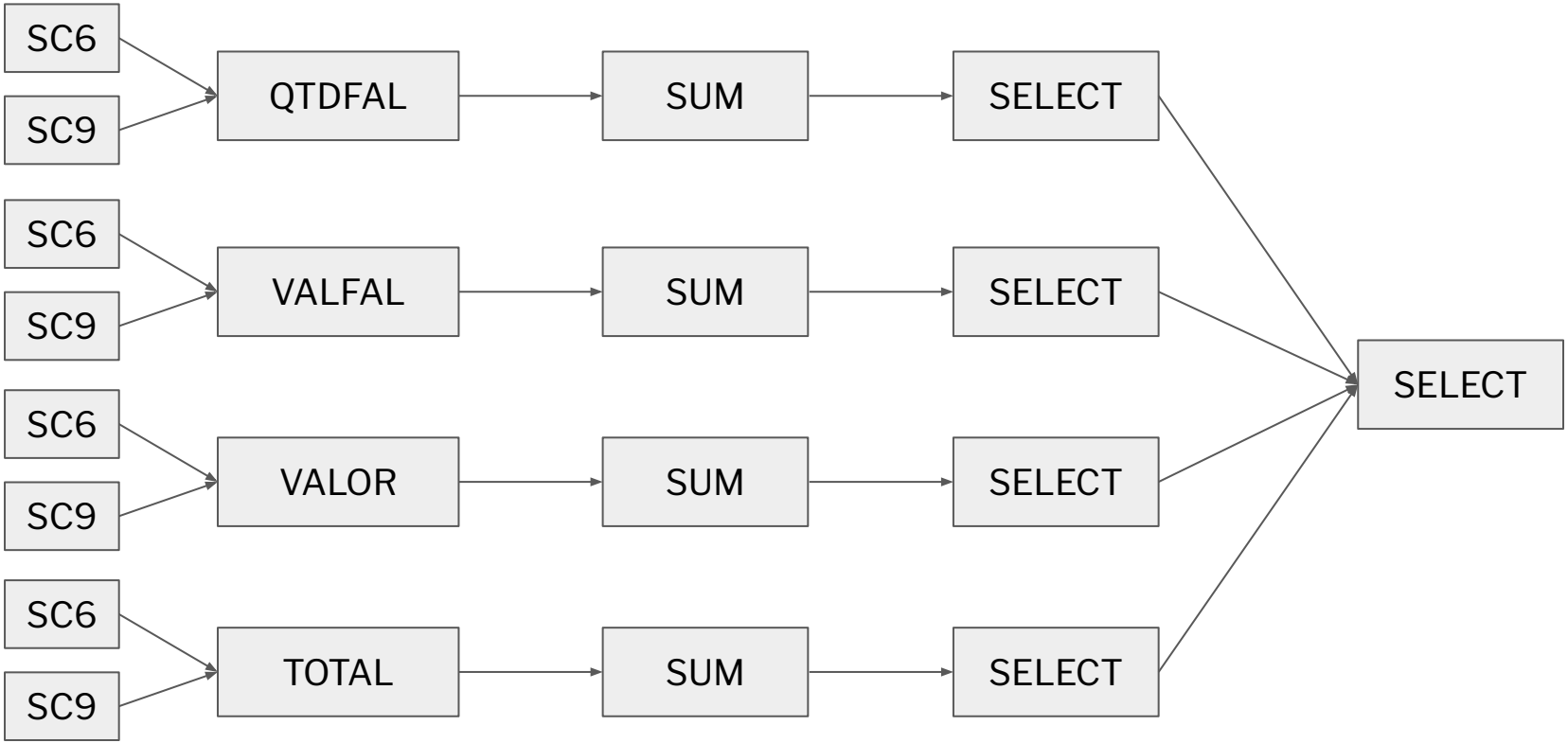
Exemplo Real

- Subqueries
- Common Table Expressions (CTE)
  Também conhecida como WITH Clause

Subqueries

```
SELECT
  SC5.C5_NUM AS PED,
  SC5.C5_VEND1 AS REPRES,
  SA1.A1_NOME AS NOME,
  (SELECT SUM((C6_QTDVEN - C6_QTDENT))
    FROM SC6010 C6, SC9010 C9 WHERE ... [10 condições] ) AS QTDFAL,
  (SELECT SUM((C6_QTDVEN - C6_QTDENT) * C6_PRCVEN)
    FROM SC6010 C6, SC9010 C9 WHERE ... [10 condições] ) AS VALFAL,
  (SELECT SUM (((C6_QTDVEN - C6_QTDENT) * C6_PRCVEN) - (C6_XQTD05 * C6_PRCVEN))
    FROM SC6010 C6, SC9010 C9 WHERE ... [10 condições] ) AS VALOR,
  SE4.E4_COND AS COND,
  SC5.C5_OBSPED AS OBS,
  SC5.C5_PREVIST as PREV,
  SC5.C5_XST05 AS ST,
  SC5.C5_XFRET AS FRET,
  (SELECT SUM(C6_VALOR)
    FROM SC6010 C6, SC9010 C9 WHERE ... [9 condições] ) AS TOT
  FROM SC5010 SC5, SC6010 SC6, SC9010 SC9, SA1010 SA1, SF4010 SF4, SE4010 SE4
    WHERE ... [29 condições]
    GROUP BY SC5.C5_NUM, SC5.C5_VEND1, SA1.A1_NOME, SE4.E4_COND,
      SC5.C5_OBSPED, SC5.C5_PREVIST, SC5.C5_XST05, SC5.C5_XFRET
    ORDER BY SA1.A1_NOME
```

Subqueries


SC6
SC9
QTDFAL
SUM
SELECT
SC6
SC9
VALFAL
SUM
SELECT
SC6
SC9
VALOR
SUM
SELECT
SC6
SC9
TOTAL
SUM
SELECT
SELECT

Common Table Expressions

Reutilizar consultas de forma temporária

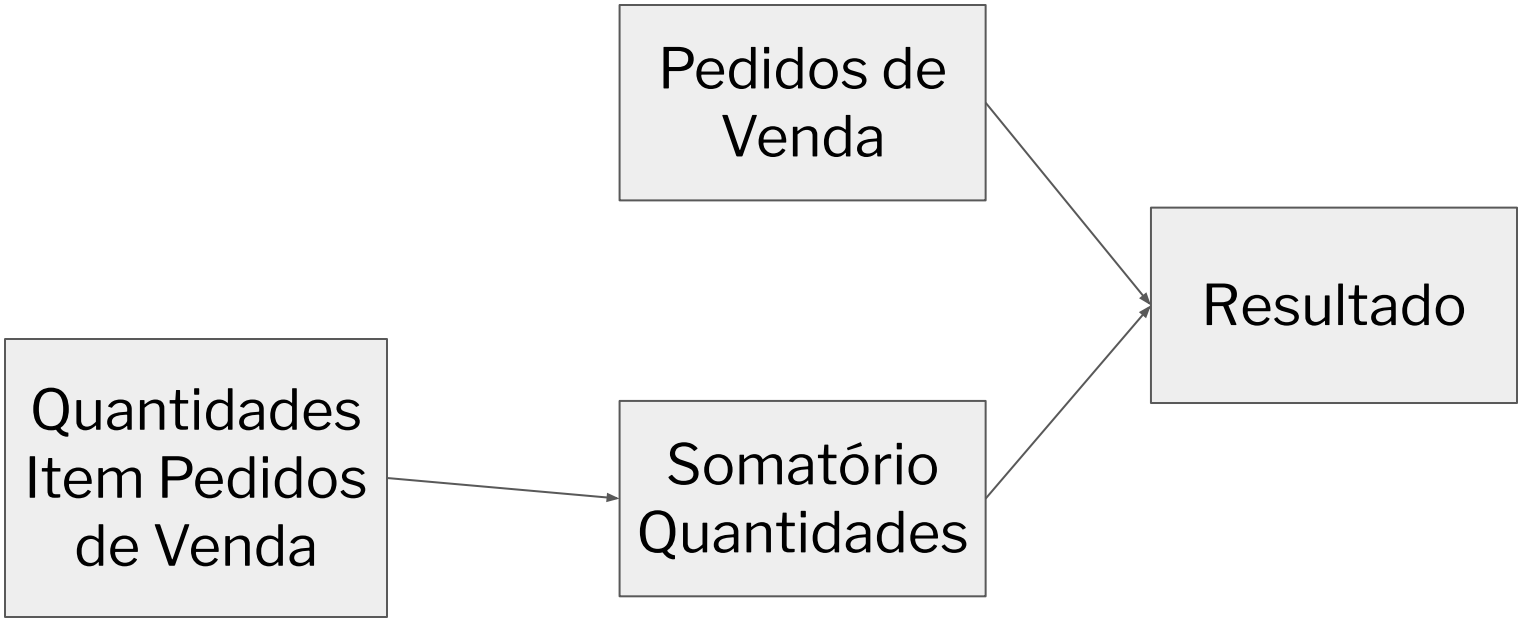
```
WITH TABLE_PEDIDO AS (
  SELECT ...
), TABLE_QUANTIDADE AS (
  SELECT ...
), TABLE_ITENS_C6 AS (
  SELECT ... FROM TABLE_QUANTIDADE ...
)
SELECT ...
  FROM
    TABLE_PEDIDO AS T_PEDIDO,
    TABLE_ITENS_C6 AS T_ITENS_C6, ...
  WHERE ...
  ORDER BY ...
```

```
), TABLE_QUANTIDADE AS (
 SELECT
   C6_NUM AS C6NUM, C6_QTDVEN AS C6QTDVEN, C6_QTDENT AS C6QTDENT,
   C6_PRCVEN AS C6PRCVEN, C6_XQTD05 AS C6XQTD05, C6_VALOR  AS C6VALOR,
   C6_XFLAG  AS C6XFLAG, C6_LOCAL  AS C6LOCAL, C6_TES AS C6TES
 FROM SC6010 C6, SC9010 C9
 WHERE ... [11 condições]

), TABLE_ITENS_C6 AS (
 SELECT
   C6NUM AS NUM,
   C6TES AS C6TES,
   SUM((C6QTDVEN-C6QTDENT)) AS QTDFAL,
   SUM((C6QTDVEN-C6QTDENT) * C6PRCVEN) AS VALFAL,
   SUM(((C6QTDVEN-C6QTDENT) * C6PRCVEN) - (C6XQTD05*C6PRCVEN)) AS VALOR,
   SUM(C6VALOR) AS TOT
 FROM TABLE_QUANTIDADE
 GROUP BY C6NUM, C6TES
)
```

Melhor Performance
e Legibilidade

Melhor Performance



Explain
(PostgreSQL 8.2+)

```
EXPLAIN SELECT * FROM tenk1 WHERE unique1 < 7000;

QUERY PLAN
-------------------------------------------------------------
 Seq Scan on tenk1 (cost=0.00..483.00 rows=7001 width=244)
   Filter: (unique1 < 7000)
```

https://www.postgresql.org/docs/current/using-explain.html

PostgreSQL Documentation

Explain Analyze

(PostgreSQL 8.2+)

```
EXPLAIN ANALYZE SELECT * FROM tenk1 WHERE ten < 7;

QUERY PLAN
-------------------------------------------------------------------------------
 Seq Scan on tenk1 (cost=0.00..483.00 rows=7000 width=244) (actual
time=0.016..5.107 rows=7000 loops=1)
   Filter: (ten < 7)
   Rows Removed by Filter: 3000
 Planning time: 0.083 ms
 Execution time: 5.905 ms
```

https://www.postgresql.org/docs/current/using-explain.html

PostgreSQL Documentation

2# PRODUTIVIDADE

Automatizando ambientes para testes
(inclusive de performance)

Falta de produtividade com...

- Tarefas diárias e repetitivas
- Baixo ou nenhum isolamento entre cenários
- Praticamente impossível lidar com várias versões de software ao mesmo tempo

Ferramentas para começar!

- Vagrant
- Virtual Box

```
vagrant init hashicorp/precise64
vagrant up
```

rcmoutinho / public-speaking

Unwatch 1 | Star 1 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Branch: master ▾ public-speaking / cte-example / Create new file | Upload files | Find file | History


rcmoutinho quick content review to adjust TOTVS information — Latest commit 94bc101 on Aug 1, 2018

..		
conf	enabling the access of any IP to the database	6 months ago
database	quick content review to adjust TOTVS information	6 months ago
queries	comment to warn that the query wont show results, only the query perf...	6 months ago
.gitignore	ignoring vagrant and macos files	6 months ago
README.md	quick content review to adjust TOTVS information	6 months ago
Vagrantfile	removing database provisioning script	6 months ago
provision.sh	file location adjustment	6 months ago

README.md

English | Português

Dramatically increase your database performance

Real example using CTE (Common Table Expression) to decrease the runtime of a query drastically. Most of the structure was preserved to respect TOTVS' ERP (Enterprise Resource Planning) project format

```
cte-example/
├── README.md
├── Vagrantfile
├── conf
│   ├── pg_hba.conf
│   └── postgresql.conf
├── database
│   ├── generate-fake-data.sh
│   └── totvs-example-schema.sql
├── provision.sh
└── queries
    ├── high-performance-query.sql
    └── low-performance-query.sql
```

Branch: master ▾ **public-speaking** / cte-example / **Vagrantfile** Find file | Copy path

rcmoutinho removing database provisioning script 431ee2c on Jul 27, 2018

1 contributor

18 lines (13 sloc) | 434 Bytes Raw | Blame | History

```
1  # -*- mode: ruby -*-
2  # vi: set ft=ruby :
3
4  Vagrant.configure("2") do |config|
5    config.vm.box = "hashicorp/precise64"
6    config.vm.network "private_network", ip: "192.168.33.10"
7
8    config.vm.provision "shell", path: "provision.sh"
9
10   config.vm.synced_folder ".", "/vagrant", type: "rsync", rsync__exclude: ".git/"
11
12   config.vm.provider :virtualbox do |vb|
13     vb.name = "cte-postgresql"
14     vb.memory = 1536
15     vb.cpus = 2
16   end
17 end
```

```
# cleanup
apt-get clean

# configuration to allow any connection (example/development purpose)
cp /vagrant/conf/pg_hba.conf /etc/postgresql/9.1/main/pg_hba.conf
chown postgres.postgres /etc/postgresql/9.1/main/pg_hba.conf

cp /vagrant/conf/postgresql.conf /etc/postgresql/9.1/main/postgresql.conf
chown postgres.postgres /etc/postgresql/9.1/main/postgresql.conf

# restarting services
```

```
#
# Database administrative login by Unix domain socket
# local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             all                     trust
# IPv6 local connections:
# host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     postgres                                peer
```

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost', '*' = all
                                # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
# Note:  Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3     # (change requires restart)
unix_socket_directory = '/var/run/postgresql'           # (change requires restart)
```

DEMO!

(a primeira!)

Curiosidades

(enquanto a query lenta executa)

https://fontawesome.com/icons?d=gallery&c=emoji

Curiosidades

(enquanto a query lenta executa)

```
20  psql -U postgres -d totvs_example << EOF
21    --
22    -- CONDIÇÃO DE PAGAMENTO
23    --
24    WITH CARGA_SE4 AS (
25      SELECT
26        '  ' AS FILIAL,
27        AUTO_COD AS CODIGO,
28        'C' || LPAD(AUTO_COD::text, 2, '0') AS CONPAG,
29        '' AS D_E_L_E_T_
30      FROM generate_series(1,40) AS AUTO_COD
31    )
32    INSERT INTO SE4010
33      SELECT * FROM CARGA_SE4;
34  EOF
35
```

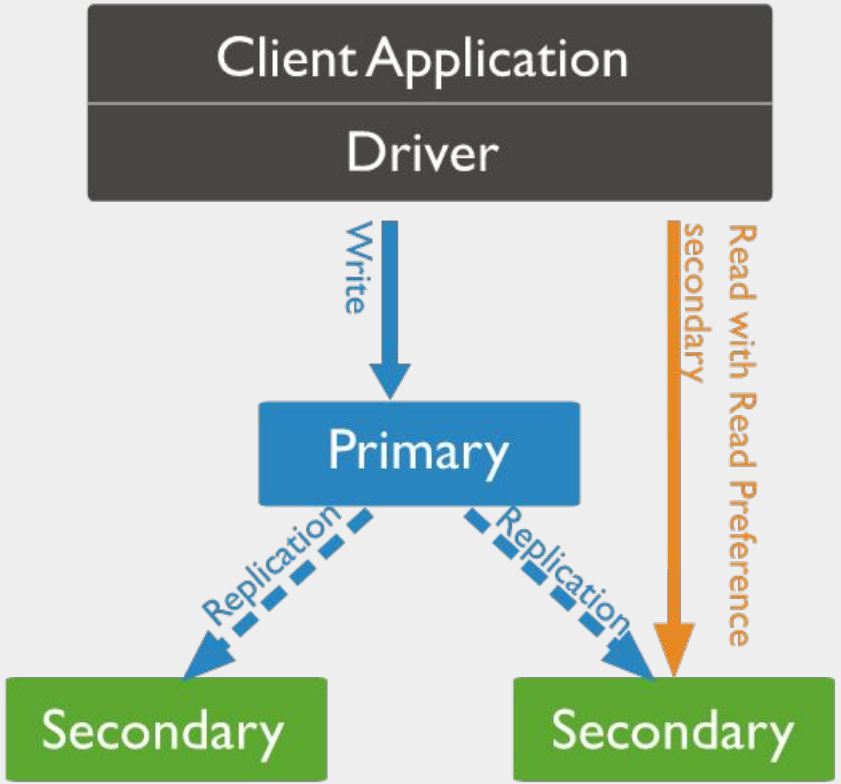
3# PERFORMANCE E PRODUTIVIDADE

Em um banco NoSQL como o MongoDB

Não fuja da modelagem!

- Análise seu cenário
- Conheça os padrões
- Sem schema <> bagunça generalizada
- Data types errados afetam o desempenho!
- Quais as operações principais?
    - Leituras ou escrita?

A replicação nativa é linda!



https://docs.mongodb.com/manual/replication/

Tipos de índice:

- Single field
- Compound index
- Multikey index
- Geospatial Index
- Text Indexes
- Hashed Indexes

Propriedades dos índices:

- Unique Indexes
- Partial Indexes
- Sparse Indexes
- TTL Indexes
- *****Em beta!*** MongoDB Atlas Full-Text Search**

Novidades da versão 4.2

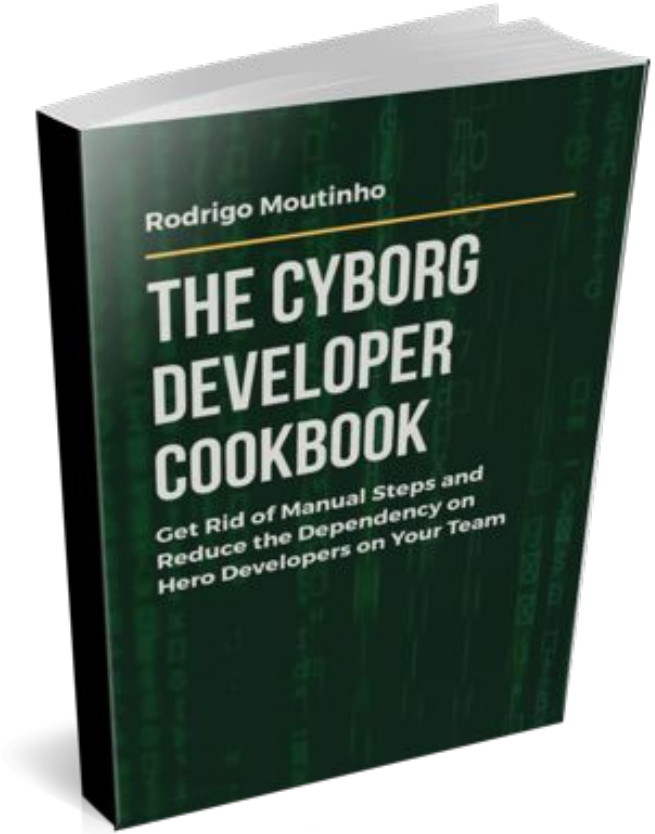
- MongoDB Atlas Full-Text Search
- Wildcard indexes
- On-Demand Materialized Views
- Operadores novos do Aggregation Framework
- Velocidade na sincronização da replicação

DEMO!

Vamos conhecer o MongoDB Compass

Extra!

- **Crie os melhores aplicativos** (eBook Gratuito) [Dani Monteiro]
  http://db4beginners.com/e-book/
- Public Speaking (palestras, demos, projetos, etc) [Rodrigo Moutinho]
  https://github.com/rcmoutinho/public-speaking
- Career
  https://code4.life
- Speaker Secrets
  https://we.code4.life/speaker-secrets
- Portal para Palestrantes do TDC
  https://we.code4.life/tdc



Demo + Slides +
eBook Automação (FREE)

cyborgdeveloper.tech

Rodrigo Moutinho
@rcmoutinho

Dani Monteiro
@DaniMonteiroDBA

#TDCSP