

Instalação e Configuração PostgreSQL

Instalação do banco de dados PostgreSQL, configuração da base de dados, ferramenta de banco, backup, etc...

- [Ajustando Horário / Timezone do Postgres](#)
- [Aumentar o número de sessões](#)
- [Configurar base PostgreSQL no PgAdmin](#)
- [Criar banco de dados/conectar/exportar via linha de comando \(cmd\)](#)
- [Definições / Particularidades do PostgreSQL](#)
- [Gerar Backup](#)
- [Gerenciar banco de dados \(Commit, nova conexão\)](#)
- [Identificar índices não utilizados \(ou não eficientes\)](#)
- [Identificar tamanho do banco de dados](#)
- [Instalação PostgreSQL v10.6 - Windows](#)
- [Lições aprendidas](#)
- [Manutenção do PostgreSQL](#)
- [Obter informação sobre o banco de dados PostgreSQL - Tamanho / Encoding / Collate / Versao](#)
- [pg_hba.config](#)
- [PostgreSQL - Verificar sessões ativas](#)

Ajustando Horário / Timezone do Postgres

Olá, se você está enfrentando problemas com formato e/ou data hora errado em comandos retornados de sua base de dados postgres é necessário fazer esse ajuste:

Baixar o arquivo com o timezone atualizado: [clique aqui](#)

Logo após substitua o arquivo original que se encontra no seguinte caminho:

C:\Program Files\PostgreSQL\10\share\timezone\America, assim que substituído, você deverá reiniciar o PostgreSQL. Depois de reiniciar o banco de dados com o novo arquivo baixado aplique as configurações abaixo:

```
ALTER DATABASE EMA SET TIMEZONE TO ' AMERICA/SAO_PAULO' ;  
ALTER SYSTEM SET TIMEZONE=" GMT+3";  
SELECT PG_RELOAD_CONF();
```

Podemos fixar GMT+3 por que não temos mais o conhecido horário de verão, se um dia ele voltar lembre-se de fazer essa alteração na data estipulada para GMT+2 garantindo assim a sincronia entre horários e conversão correta do horário UTC.

Aumentar o número de sessões

**** ATENÇÃO: Avalie com cuidado esta alteração para que não interfira no uso de memória total do servidor de banco de dados, 500 sessões é um número médio usado em nossos clientes e está dentro de uma margem de segurança.**

- Vá até o caminho: **C:\Program Files\PostgreSQL\10\data**
- Edite o arquivo: **postgresql.conf**
- Na linha 64, no comando : **max_connections = 100**
- Altere para **max_connections = 500**
- Reinicie o serviço do PostgreSQL

[image-1646930837110.png](#)

Image not found or type unknown

Para iniciar, parar ou reiniciar o serviço de banco de dados PostgreSQL, basta acessar a tela de Serviços do Windows, procurar pelo serviço "postgresql-..." e clicar na opção desejada.

[image-1646930859997.png](#)

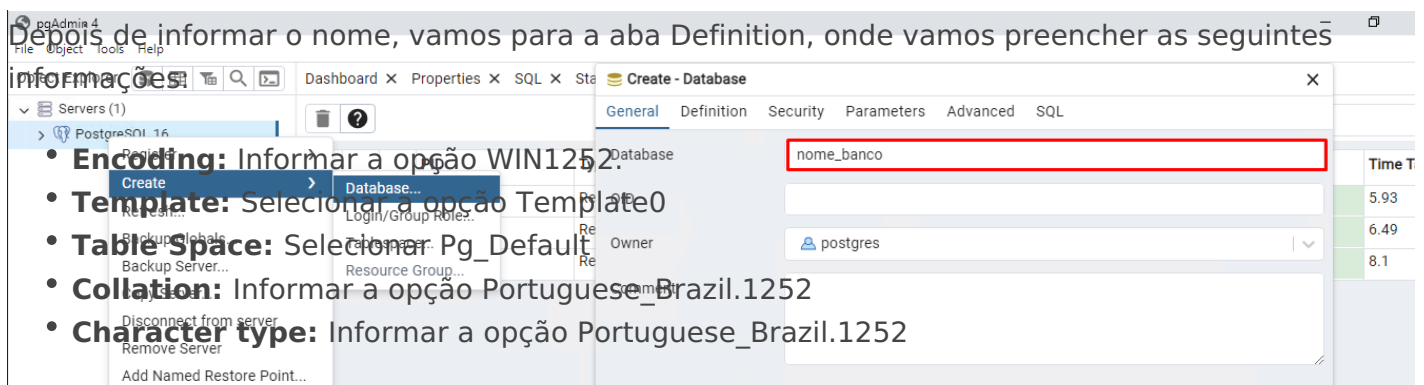
Image not found or type unknown

Configurar base PostgreSQL no PgAdmin

Com o **PostgreSQL 10** instalado, pode-se configurar a base de dados no **pgAdmin 4**. Para isso, temos que criar um banco de dados e fazer o Restore de um arquivo de **backup**. Veja o exemplo a seguir.

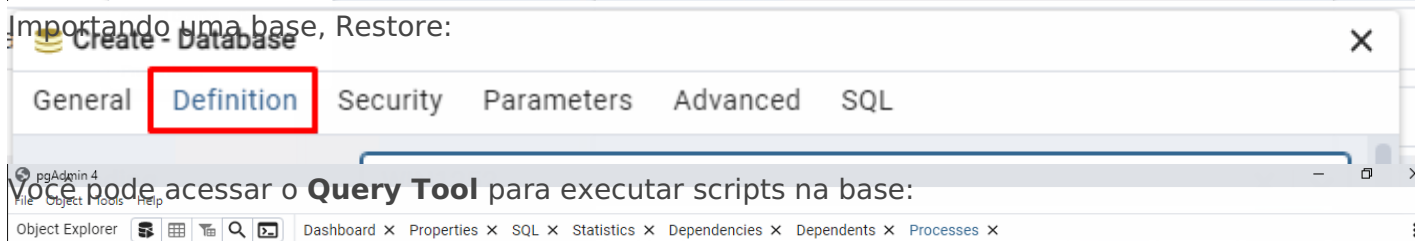
Criação da base de dados:

Depois de informar o nome, vamos para a aba Definition, onde vamos preencher as seguintes informações:



- **Encoding:** Informar a opção WIN1252.
- **Template:** Selecionar a opção Template0
- **Table Space:** Selecionar Pg_Default
- **Collation:** Informar a opção Portuguese_Brazil.1252
- **Character type:** Informar a opção Portuguese_Brazil.1252

Importando uma base, Restore:



Depois de acessar o **Query Tool** para executar scripts na base:

Criar banco de dados/conectar/exportar via linha de comando (cmd)

Criar banco

Siga os passos abaixo:

- Abra o Prompt de Comando em modo administrador.
- Execute os comandos abaixo, alterando caso necessário o nome do banco:
- C:\WINDOWS\system32> cd\
- C:\> psql --username=postgres
- Senha para usuário postgres: senha do banco
- postgres=# CREATE DATABASE ema WITH OWNER="postgres" ENCODING="WIN1252" TEMPLATE="template0";
- postgres=# \q
- C:\> exit

Você pode usar também o pgAdmin4 para, siga as instruções:

- Em um computador com Windows, efetue a conexão com o banco com o PgAdmin
- Em "Databases", clique com botão direito, Create - database
- Insira o nome do banco e clique em "Save"

[image-1646932260973.png](#)

Image not found or type unknown

Conectar ao banco via PSQL:

Para conectar a instância principal

- **psql --username=postgres**

Para conectar em um usuário específico

- **psql --username=postgres --dbname=ema**

Caso ao digitar psql apresente, você deverá inserir uma variável de ambiente no seu windows ou rodar os comandos acima, direto na pasta: C:\Program Files\PostgreSQL\10\bin

Importar banco de dados:

- Clique com botão direito em sua base recém criada - Restore
- Insira manualmente o caminho do arquivo .dmp e clique em restore
- Aguarde o término

image-1646932357327.png

Image not found or type unknown

Exportar base via linha de comando:

- Abra o Prompt de Comando em modo administrador.
- Execute os comandos abaixo, alterando caso necessário os parâmetros do pg_dump (file, dbname, host, port):
- C:\WINDOWS\system32> cd\
- C:\> pg_dump --file=arquivo.dmp -F c -v -b --dbname=ema --host=127.0.0.1 --port=5432 --username=postgres
- Senha: senha do banco
- C:\> exit

Definições / Particularidades do PostgreSQL

Serviço/Processo no Windows

Para iniciar o banco de dados deve-se ser iniciado o serviço do postgres no gerenciador de serviços do Windows, ou configurá-lo como automático.

image-1656358069653.png

Image not found or type unknown

Para cada seção no banco ou thread é criado um registro **postgres.exe** no gerenciador de tarefas, este comportamento é normal, assim que as seções forem encerradas, o processo também será finalizado.

image-1656358102391.png

Image not found or type unknown

Cada PID do sistema operacional Windows poderá ser localizado dentro do banco de dados e sua seção monitorada.

Tuplas

Este é o tipo de dado da coluna do sistema, o identificador de tupla é um par (número do bloco, índice da tupla dentro do bloco) que identifica a posição física da linha dentro de sua tabela.

Transações - Locks e Bloqueios

Os Locks podem ser definidos como bloqueios executados em objetos ou dados no banco de dados. Esses bloqueios podem ser gerados automaticamente (em função do gerenciamento Multiversão/Multiusuário (MVCC) do PostgreSQL) e manualmente (com comandos ou operações que necessitem “travar” uma tabela ou alguns de seus registros).

O PostgreSQL trabalha com diversos tipos ou modos de bloqueios diretamente vinculados a ações específicas ao programa para controlar o acesso simultâneo aos dados e tabelas.

Bloqueio	Descrição	Entra em conflito com
----------	-----------	-----------------------

ACCESS SHARE	O comando SELECT obtém um bloqueio deste modo nas tabelas referenciadas. Em geral, qualquer comando que apenas leia a tabela (sem modificá-la) obtém este modo de bloqueio.	ACCESS EXCLUSIVE
ROW SHARE	O comando SELECT FOR UPDATE obtém o bloqueio neste modo na(s) tabela(s) de destino.	EXCLUSIVE e ACCESS EXCLUSIVE
ROW EXCLUSIVE	Os comandos UPDATE, DELETE e INSERT obtém este modo de bloqueio na tabela de destino (além do modo de bloqueio ACCESS SHARE nas outras tabelas referenciadas). Em geral, este modo de bloqueio é obtido por todos os comandos que modificam os dados da tabela.	SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE, ACCESS EXCLUSIVE
SHARE UPDATE EXCLUSIVE	Obtida pelo comando VACUUM (sem a opção FULL). Protege a tabela contra mudanças simultâneas no esquema durante a execução do comando VACUUM	SHARE UPDATE EXCLUSIVE, SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE.
SHARE	Obtido pelo comando CREATE INDEX.	ROW EXCLUSIVE, SHARE UPDATE EXCLUSIVE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE.
SHARE ROW EXCLUSIVE	Este modo de bloqueio não é obtido automaticamente por nenhum comando do PostgreSQL.	ROW EXCLUSIVE, SHARE UPDATE EXCLUSIVE, SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE
EXCLUSIVE	Este modo de bloqueio não é obtido automaticamente por nenhum comando do PostgreSQL.	ROW SHARE, ROW EXCLUSIVE, SHARE UPDATE EXCLUSIVE, SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE.
ACCESS EXCLUSIVE	Obtido pelos comandos ALTER TABLE, DROP TABLE e VACUUM FULL. Este é também o modo de bloqueio padrão para o comando LOCK TABLE sem a especificação explícita do modo.	Entra em conflito com todos os modos de bloqueio Este modo garante que a transação que o obteve seja a única que esteja acessando a tabela.

****Para ver o tipo de deadlock utilize o dashboard do pgAdmin 4 e vá até a coluna *locks* ou utilize o comando:**

```
SELECT *
FROM PG_LOCKS
```


Gerar Backup

Neste tópico veremos como gerar o **backup** via linha de comando ou manualmente de uma base PostgreSQL.

Backup gerado manualmente (PgAdmin)

- Clique com botão direito em cima da base desejada
- Escolha a opção backup

image-1646922940108.png

Image not found or type unknown

*****Não há necessidade de informar os outros campos, apenas o "Filename" com o caminho de onde o backup deve ser salvo, e o nome que deve dar para ele.***

- Na tela a seguir, selecione o local e nome do arquivo
- Clique em **backup**
- Aguarde a geração do arquivo **.SQL**

Backup via Linha de Comando (CMD)

- Primeiro, acesse o gerenciador de tarefas e siga até a aba de serviços, parando o serviço **Ema_Start** e reiniciando os serviços do postgres.
- Após reiniciado os serviços do postgresQL, deve abrir o terminal (**cmd**), pressionando **windows + R** e digite **cmd** ou procure por **prompt de comando** no windows
- Ao acessar o cmd, digite o seguinte comando: **cd /**

image-1646923029584.png

Image not found or type unknown

****** Este comando é responsável em acessar a pasta raiz, no caso fica na pasta C:***

- Após acessar a pasta raiz, digite o seguinte comando: **cd Program Files\PostgreSQL\10\bin**

image-1646923061266.png

Image not found or type unknown

- Assim que acessar a pasta bin do **postgreSQL** iremos executar o seguinte comando

parar gerar o arquivo dump:

```
pg_dump --file=C:\BackupPo
```

```
stgreSQL\arquivo.dmp -
```

```
F c -v -b --dbname=ema --host=127.0.0.1 --port=5432 --username=postgres
```

Obs: --file= neste comando é informado o caminho que vai ser salvo o arquivo .dmp, o nome pode ser alterado para outro como por exemplo:

```
C:\BackupPostgreSQL\backup.dmp
```

```
--dbname= Nome da base cadastrada.
```

```
--username= Usuário para acessar o database.
```

*****Caso não seja informado um caminho para o arquivo ele irá criar dentro de C:***

- Após informar todos os parâmetro no comando do pg_dump pressione [Enter] para executar o comando, logo após ele irá pedir a senha da base, basta digitar e confirmar.

image-1646923495804.png

Image not found or type unknown

******A senha ao digitar não é mostrada, cuidado ao digitar a senha.***

- Assim que terminar de digitar a senha e confirmar pressionando [Enter] o pg_dump começará a criar o arquivo na pasta informada no --file :
- Assim que finalizar ele estará na pasta informada, para sair do prompt pasta digitar exit.
- Realizado este procedimento já pode ser iniciado novamente os serviços da Ema.

*****Este procedimento não pode ser executado com os serviços da Ema rodando, pois pode corromper o arquivo.**

Gerenciar banco de dados (Commit, nova conexão)

Desativar Auto Commit

COMMIT consolida a transação, ou seja, executa os comandos em definitivo, seja um **delete**, **update**, etc. Já o **ROLLBACK** desfaz a transação inteira – nenhuma declaração SQL contida na transação é executada.

É uma ação importante pois imagine que você executa um **update** sem **where** e altera muitos dados importantes de forma errônea. Se o **auto** commit estiver desligado, você não precisa se preocupar pois esse comando não foi executado em definitivo. É possível fazer um commit manual, quando você já tem certeza que o comando SQL está correto, ou pode escolher deixar automático, sempre que um comando foi executado, automaticamente já está consolidado a transação.

Siga os passos abaixo:

- Escolha o banco desejado
- Clique com o botão direito e vá até Query tool
- Na tela de SQL que sera mostrada, localize o ícone mostrado na imagem abaixo
- Desmarque a opção Auto commit ?
- Você deverá fazer esta configuração toda vez que fizer login na ferramenta

[image-1646931260438.png](#)

Image not found or type unknown

- Para desativar o Auto Commit de forma permanente siga os passos da imagem abaixo

[image-1646931420913.png](#)

Image not found or type unknown

Nova conexão:

- Clique com o segundo botão em Servers, e siga para **Create > Server...**

image-1646931873181.png

Image not found or type unknown

- Na aba General, insira apenas o nome da conexão, ex. **ema**

image-1646931910412.png

Image not found or type unknown

- Vá para a aba **connection**
- No campo **Host**, insira o **IP** do servidor onde está o banco
- Em **Maintenance database** insira postgres
- Em **username** insira postgres
- Insira a **senha** do banco e clique em "Save".
- Todos os bancos de dados do usuário postgres serão mostrados dentro de "Databases"

image-1646931970602.png

Image not found or type unknown

Identificar índices não utilizados (ou não eficientes)

```
with
table_scans as (
select
[]tables.relid as relid,
[]tables.schemaname as schemaname,
[]tables.relname as tablename,
[]tables.idx_scan as table_idx_scan_count,
[]tables.idx_tup_fetch as table_idx_tup_fetch,
[]tables.seq_scan as table_seq_scan_count,
[]tables.seq_tup_read as table_seq_tup_read,
[]tables.idx_scan + tables.seq_scan as table_sum_all_scans,
[]tables.n_tup_ins as table_write_insert_count,
[]tables.n_tup_upd as table_write_update_count,
[]tables.n_tup_del as table_write_delete_count,
[]tables.n_tup_ins + tables.n_tup_upd + tables.n_tup_del as table_sum_all_writes,
[]tables.n_tup_hot_upd as table_tup_hot_upd_count,
[]tables.n_live_tup as table_live_tup_count,
[]pg_relation_size(relid) as table_bytes
from
[]pg_stat_user_tables as tables
),
database_writes as (
select
[]sum(table_sum_all_writes) as database_sum_all_writes
from
[]table_scans
),
indexes as (
select
[]idx_stat.relid as relid,
[]idx_stat.indexrelid as indexrelid,
[]idx_stat.schemaname as schemaname,
```

```

idx_stat.relname as tablename,
idx_stat.indexrelname as indexname,
idx_stat.idx_scan as index_idx_scan_count,
idx_stat.idx_tup_read as index_idx_tup_read,
idx_stat.idx_tup_fetch as index_idx_tup_fetch,
pg_relation_size(idx_stat.indexrelid) as index_bytes,
indexes.indexdef ~* 'USING btree' as idx_is_btree
from
pg_stat_user_indexes as idx_stat
join pg_index as pg_index
  on (idx_stat.indexrelid = pg_index.indexrelid)
join pg_indexes as indexes
  on (idx_stat.schemaname = indexes.schemaname
      and idx_stat.relname = indexes.tablename
      and idx_stat.indexrelname = indexes.indexname)
where
pg_index.indisunique = false
),
index_ratios as (
select
indexes.schemaname as schemaname,
indexes.tablename as tablename,
indexes.indexname as indexname,
indexes.index_idx_scan_count as index_idx_scan_count,
indexes.index_idx_tup_read as index_idx_tup_read,
indexes.index_idx_tup_fetch as index_idx_tup_fetch,
round(case when indexes.index_idx_scan_count = 0 or table_scans.table_live_tup_count = 0
then -1 :: numeric
      else indexes.index_idx_tup_fetch :: numeric / indexes.index_idx_scan_count /
table_scans.table_live_tup_count * 100 end, 2) as idx_pct_table_fetched,
table_scans.table_idx_scan_count as table_idx_scan_count,
table_scans.table_seq_scan_count as table_seq_scan_count,
table_scans.table_seq_tup_read as table_seq_tup_read,
table_scans.table_sum_all_scans as table_sum_all_scans,
round((case when table_scans.table_sum_all_scans = 0
then -1 :: numeric
      else indexes.index_idx_scan_count :: numeric / table_scans.table_sum_all_scans * 100 end),
2) as index_scan_pct,
table_scans.table_write_insert_count as table_write_insert_count,

```

```

table_scans.table_write_update_count as table_write_update_count,
table_scans.table_write_delete_count as table_write_delete_count,
table_scans.table_sum_all_writes as table_sum_all_writes,
round((case when table_scans.table_sum_all_writes = 0
then indexes.index_idx_scan_count :: numeric
      else indexes.index_idx_scan_count :: numeric / table_scans.table_sum_all_writes end), 2)
as scans_per_write,
table_scans.table_tup_hot_upd_count as table_tup_hot_upd_count,
table_scans.table_live_tup_count as table_live_tup_count,
indexes.index_bytes as index_bytes,
pg_size_pretty(indexes.index_bytes) as index_size,
table_scans.table_bytes as table_bytes,
pg_size_pretty(table_scans.table_bytes) as table_size,
indexes.idx_is_btree as idx_is_btree
from
indexes
join table_scans
on using (relid)
),
index_groups as (
select
1 as grp,
' Never Used Indexes' as reason,
*
from
index_ratios
where
index_ratios.index_idx_scan_count = 0
and index_ratios.idx_is_btree
union all
select
2 as grp,
' Low Scans, High Writes' as reason,
*
from
index_ratios
where
scans_per_write <= 1
and index_scan_pct < 10
and index_idx_scan_count > 0

```

```

and table_sum_all_writes > 100
and idx_is_btree
union all
select
3 as grp,
'Seldom Used Large Indexes' as reason,
*
from
index_ratios
where
index_scan_pct < 5
and scans_per_write > 1
and index_idx_scan_count > 0
and idx_is_btree
and index_bytes > 100000000
union all
select
4 as grp,
'High-Write Large Non-Btree' as reason,
index_ratios.*
from
index_ratios,
database_writes
where
(table_sum_all_writes :: numeric / coalesce(nullif(database_sum_all_writes, 0), 1)) > 0.02
and not idx_is_btree
and index_bytes > 100000000
union all
select
5 as grp,
'(+ Sem Efetividade)' as reason,
index_ratios.*
from
index_ratios
where
idx_is_btree
and index_idx_scan_count > 0
and idx_pct_table_fetched > 20
union all
select

```



```

    6 as grp,
    (+) Índice Médio (100MB a 500MB)' as reason,
    index_ratios.*
from
    index_ratios
where
    index_bytes >= 100000000
    and index_bytes < 500000000
union all
select
    7 as grp,
    (+) Índice Grande (500MB a 1 GB)' as reason,
    index_ratios.*
from
    index_ratios
where
    index_bytes >= 500000000
    and index_bytes < 1000000000
union all
select
    8 as grp,
    (+) Índice Enorme (mais de 1 GB)' as reason,
    index_ratios.*
from
    index_ratios
where
    index_bytes >= 1000000000
order by
    grp,
    index_bytes desc
)
select
    reason,
    schemaname,
    tablename,
    indexname,
    table_size,
    index_size
from

```


Identificar tamanho do banco de dados

Como o PostgreSQL tem uma forma diferente de armazenar as informações, ou seja, não é um arquivo único, o comando abaixo auxilia na identificação do tamanho do banco de dados.

```
SELECT
table_name nome_tabela,
pg_size_pretty(table_size) || ' (' || CASE WHEN total_size = 0 THEN 0.00 ELSE
round(table_size * 100 / total_size) END || ' %)' AS tamanho_dados,
pg_size_pretty(indexes_size) || ' (' || CASE WHEN total_size = 0 THEN 0.00 ELSE
round(indexes_size * 100 / total_size) END || ' %)' AS tamanho_indice,
pg_size_pretty(total_size)
AS tamanho_total
FROM (
(SELECT
table_name,
pg_table_size(table_name) AS table_size,
pg_indexes_size(table_name) AS indexes_size,
pg_total_relation_size(table_name) AS total_size
FROM (
SELECT ('"' || table_schema || '". ' || table_name || '"') AS table_name
FROM information_schema.tables
WHERE NOT table_schema IN ('pg_catalog', 'information_schema')
) AS all_tables
ORDER BY total_size DESC)
UNION ALL
(SELECT
' TOTAL',
sum(pg_table_size(table_name)) AS table_size,
sum(pg_indexes_size(table_name)) AS indexes_size,
sum(pg_total_relation_size(table_name)) AS total_size
FROM (
SELECT ('"' || table_schema || '". ' || table_name || '"') AS table_name
FROM information_schema.tables
```

```
WHERE NOT table_schema IN ('pg_catalog', 'information_schema')  
) AS all_tables)  
) AS pretty_sizes;
```

Instalação PostgreSQL v10.6 - Windows

ATENÇÃO: Esta instrução é destinada à quem deseja utilizar com a ferramentas **Ema Software**.

Para fazer a instalação do banco de dados **PostgreSQL v10.6**, baixar o instalador e seguir os passos das imagem à seguir.

[image-1646919921971.jpg](#)

Image not found or type unknown

[image-1646920058232.jpg](#)

Image not found or type unknown

[image-1646920073492.jpg](#)

Image not found or type unknown

Lições aprendidas

Aqui documentamos a evolução de aprendizado com a utilização do banco de dados em relação a nossos produtos.

Incidentes na instalação:

Permissões de administrador:

- Para instalar o PostgreSQL, tenha certeza que o usuário do servidor possua total permissão de administrador.
- Ele precisa também ser um administrador no domínio, caso cliente utilize o active directory.
- Do contrário, uma mensagem de erro será reportada no momento da instalação na tentativa de criação do banco e do usuário padrão postgres

*****OBSERVAÇÃO:** Em alguns casos, você deverá instalar o PostgreSQL antes de inserir o servidor no domínio

DLL libpq faltando, ou erro de Firedac

Mensagem: **[FireDAC][Phys][PG]-314. Cannot load vendor library [C:\EXE\libpq.dll]. Hint: check it is in the PATH or application EXE directories, and has x86 bitness.**

Este erro poderá ser apresentado no momento de clicar em testar conexão no Ema configurador e não necessariamente é relacionado a falta da DLL libpq e sim a ausência do Microsoft Visual C++.

Ele também pode aparecer caso você tenha instalado o PostgreSQL 64bits com o Microsoft Visual C++ incluso.

O Ema Servidor é 32bits então vai precisar do Microsoft Visual C++ 32 bits, basta apenas baixar e instalar o Microsoft Visual C++ 32bits que o Ema Configurador já vai funcionar. Em resumo, você terá instalado:

- Visual c++ 64bits para funcionar o banco e o PgAdmin.
- Visual c++ 32bits para funcionar os sistemas Ema.

- Acesse o site da microsoft
- Baixe a versão 32
- Pra Windows 10, versão 2015
- Para Windows 7, versão 2013

- Instale, feche o ema configurador e abra novamente e repita o teste de conexão
- Confira se a DLL libpq está localizada na pasta SysWow64 ou em Contas ERP

Caso as você faça as etapas acima e mesmo assim continue dando erro de [FireDAC][Phys][PG]-314. Cannot load vendor library [C:\EXE\libpq.dll], copie para a pasta Contas Erp acesse esse link:

<https://pt.stackoverflow.com/questions/166085/erro-dll-conex%C3%A3o-firedac-com-postgresql>

invalid password packet size

Você não efetuou corretamente as configurações recomendadas neste página meu jovem!
Reveja estas configurações.

- **pg_hba.config**
- **Aumentar o número de sessões**

no pg_hba.conf entry for host

Este erro geralmente apresenta quando colocamos o ip do servidor no Ema configurador e clicamos em testar.

Para resolver:

- Pare o serviço do PostgreSQL
- Edite o arquivo: C:\PostgreSQL\data\pg10\pg_hba.conf
- Logo abaixo de #IPv4 local & Remote connections, insira a linha:
- **host all all 0.0.0.0/0 password**
- Salve o arquivo, o mesmo deverá ficar igual a imagem abaixo

Inicie o postgresQL e refaça o teste

[image-1646933006876.png](#)

Image not found or type unknown

application server could not be contacted

Esse erro pode aparecer quando você está iniciando o PgAdmin V4.

Para resolver:

- Deletar a pasta em AppData\Roaming\pgAdmin do seu usuário e executar o PgAdmin 4 como administrador.
- Depois pode iniciar o PgAdmin V4 normalmente.

Manutenção do PostgreSQL

Vacuum/analyze

Para garantir ainda mais performance em nosso banco e retirar toda sujeira de dados ,no momento em que o comando **VACUUM** é executado, é feita uma varredura em todo o banco a procura de registros inúteis, onde estes são fisicamente removidos diminuindo o tamanho físico do banco.

Mas além de apenas remover os registros, o comando **VACUUM** encarrega-se de organizar os registros que não foram deletados, garantindo que não fiquem espaços/lacunas em branco após a remoção dos registros inúteis.

****Opções:** a função de vacuum possui 3 parâmetros básicos conforme abaixo:

Full

Quando o vacuum é utilizado em conjunto com este parâmetro, então é feita uma limpeza completa de todo o banco, em todas as tabelas e colunas. Este processo geralmente é demorado e evita que qualquer outra operação no banco seja realizada.

Freeze

Força o congelamento de qualquer entrada e saída do banco no momento de rodar o vacuum.

Analyze

Ao usar o ANALYSE junto ao seu comando VACUUM ele irá atualizar as estatística do banco de dados a fim de melhorar a performance das pesquisas.

- Para executar abra o gerenciador de banco pgAdmin3 ou pgAdmin 4;
- Clique com botão direito em cima do banco de dados desejado;
- Escolha maintenance;
- Marque a opção Vacuum;

Escolha a opção (Full, freeze ou analyze) conforme descrição acima. O comando também pode ser usado apenas em uma tabela, ou até uma coluna do banco.

Obter informação sobre o banco de dados PostgreSQL - Tamanho / Enconding / Collate / Versao

Abaixo estamos compartilhando um comando para ter acesso a algumas informações sobre seu PostgreSQL e seu sistema.

Dados retornados:

- *Nome do banco*
- *Tamanho do banco de dados*
- *Tamanho do banco de dados em MB*
- *Enconding da base*
- *Collate da base*
- *Versão do PostgreSQL*
- *Versão Software EMA*
- *Versão do Banco de dados EMA*

```
SELECT DATNAME AS BANCO,  
       PG_DATABASE_SIZE(DATNAME) AS TAMANHO,  
       PG_SIZE_PRETTY(PG_DATABASE_SIZE(DATNAME)) AS TAMANHO_MB,  
       PD."encoding" AS ENCONDING,  
       PD.DATCOLLATE AS COLLATE,  
       PD.DATDBA AS VERSAO_PG,  
       V.VERSAO AS VERSAO_EMA,  
       V.VERSAODB AS VERSAO_EMADB  
FROM PG_DATABASE PD  
JOIN VERSAODB V ON 1=1  
WHERE UPPER(DATNAME) = ' EMA' ;
```

***Lembre-se:** O Enconding e o Collate são do banco de dados, seu 'cliente' de banco de dados pode estar configurado com outro Enconding por exemplo, causando problemas com acentuação

ou caracteres especiais.

pg_hba.config

Após a instalação do PostgreSQL é necessário configurar o arquivo pg_hba.conf para mudar as opções de md5 para password.

O PostgreSQL, é configurado por padrão para não exigir senha de conexões locais com o banco. Porém devemos alterar para exigir sempre senha para conexão. Caso não façamos isso qualquer pessoa que utilizar a máquina em que o banco está instalado conseguirá acessar o banco sem informar a senha.

Portanto SEMPRE deve-se aplicar esta configuração.

O arquivo estará no caminho C:\Program Files\PostgreSQL\10\data.

- Abra o arquivo:C:\Program Files\PostgreSQL\10\data\pg_hba.conf
- Na coluna "METHOD" substitua "**trust**" e "**MD5**" (conexão de confiança) por "**password**"

image-1646930777511.png

Image not found or type unknown

- Salve e feche o arquivo novamente.

PostgreSQL - Verificar sessões ativas

O SQL abaixo é utilizado para identificar em banco de dados PostgreSQL todas as sessões que estão ativas no seu banco de dados, que eventualmente podem ocasionar em locks nas suas tabelas, travando assim as operações.

```
select datname,  
       procpid,  
       username,  
       application_name,  
       client_addr,  
       client_hostname,  
       backend_start,  
       query  
from pg_stat_activity
```

****OBSERVAÇÃO:** Em versões acima da 9.2, a coluna **procpid** teve seu nome alterado para **pid**, basta renomea-la e seu SQL irá rodar corretamente.