Trustly

Migrations with no Pain

Trustly

Lucio Chiessi

- De Guaratinguetá, SP para Vitória, ES.
- Senior Database Administrator na Trustly Inc.
- Presidente da Associação Brasileira de Usuários de PostgreSQL.
- Fui professor do Curso de Extensão em Banco de Dados PostgreSQL na Universidade do Estado do Rio de Janeiro (UERJ).
- Palestrante em PostgreSQL Conference Brasil (PGConf), PHP Conference Brasil, ConFLOSS e também ministrei treinamentos de Zabbix e PostgreSQL.

lucio.chiessi@trustly.com
www.linkedin.com/in/lucio-chiessi

Patrocínio

Diamante

AGGRANDIZE

COMMVAULT

TD SYNNEX

Platina

GOLDENGATEBR
Consultoria & Treinamentos

DISCOVER

Ouro

scansource

VERTICA
by opentext

Prata

TRACES

CAFÉ COM CLOUD

ROX
We take care of your data

Apoio

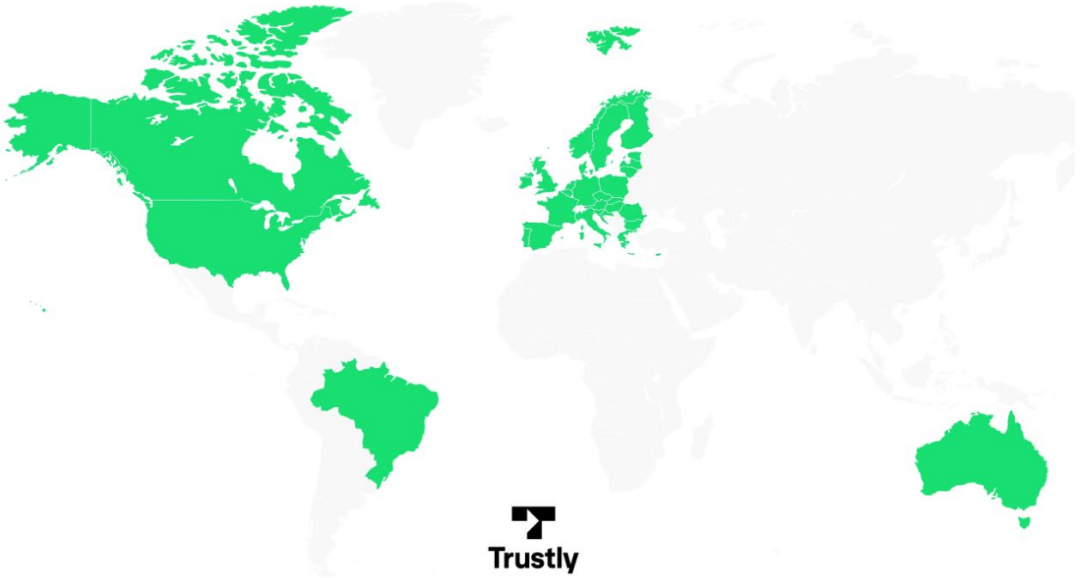
FIAP

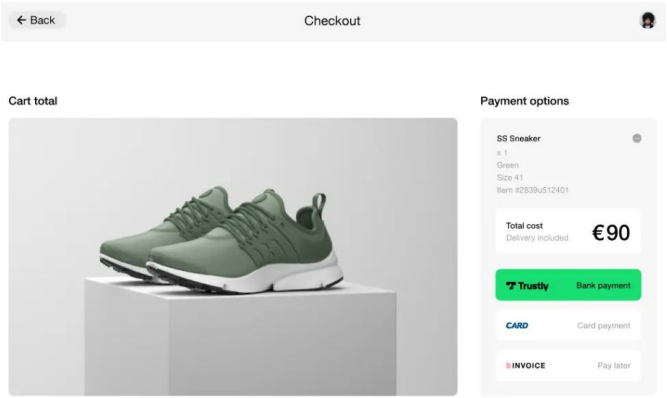
GRUPO POSEIDON DIGITAL

Trustly

Global Fintech

Pagamentos de conta para conta (A2A), sem cartões, sem fazer download de aplicativo, sem registro.





6.3K *bancos conectados*

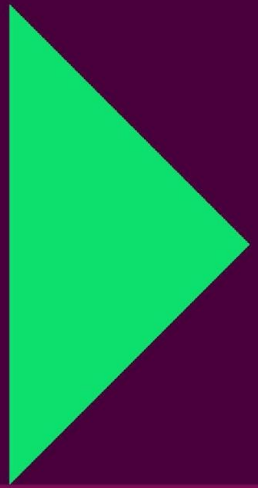
8.1K *comerciantes*

525M *consumidores*

Trustly

Migrations

- **Procedimentos automatizados** (tools) para atualizar a estrutura de nossas tabelas e outros;
- Geralmente feitas por ferramentas de deploy;
- Implementar as mudanças necessárias para as novas releases;
- Deploys sem problemas, locks ou tempos de parada;



Trustly

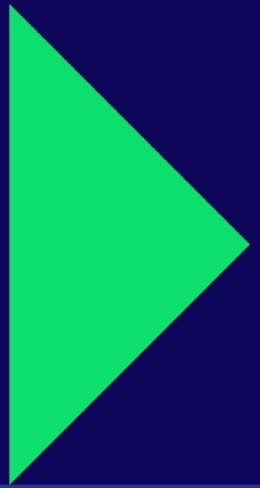
Problemas

- Aplicação parada/travada, aguardando;
- Erros / perda de faturamento;
- Muita dor de cabeça;

Trustly

Procedimentos

- Alter table;
- Add column (not null? / default value?);
- Change column type / rename;
- Criação/drop de índices;
- Add check constraint;
- Unique / Primary Key / Foreign Key;



Trustly

Steps

Trustly

01

Versão do PostgreSQL atualizada

version() >= 11

Trustly

02

Precisamos entender bem os mecanismos de locks (como funciona)

Trustly

03

Sempre faça uso dos `timeouts`, com força

Trustly

04

Transações explícitas longas

monitore / evite

Trustly

05

Evite escritas ou leituras longas no processo

Lembre-se dos locks

Trustly

06

Use o `if not exists` / `if exists` se disponível ou necessário

Trustly

07

A opção de `concurrently` sempre salva

Mas faça fora da migration

Trustly

08

Sempre monitore todo o processo e saiba o que está acontecendo

Trustly

locks

DDL quase sempre necessitam de **locks exclusivos**

Devemos **entender** e **esperar** por eles

Devem ser **rápidos**

Trustly

locks

```
- processo 1 -
#> begin transaction;
#> select * from tb1 where id <= 5 order by id;

- migrator -
#> alter table tb1 add column c19 varchar; (wait aqui)
```

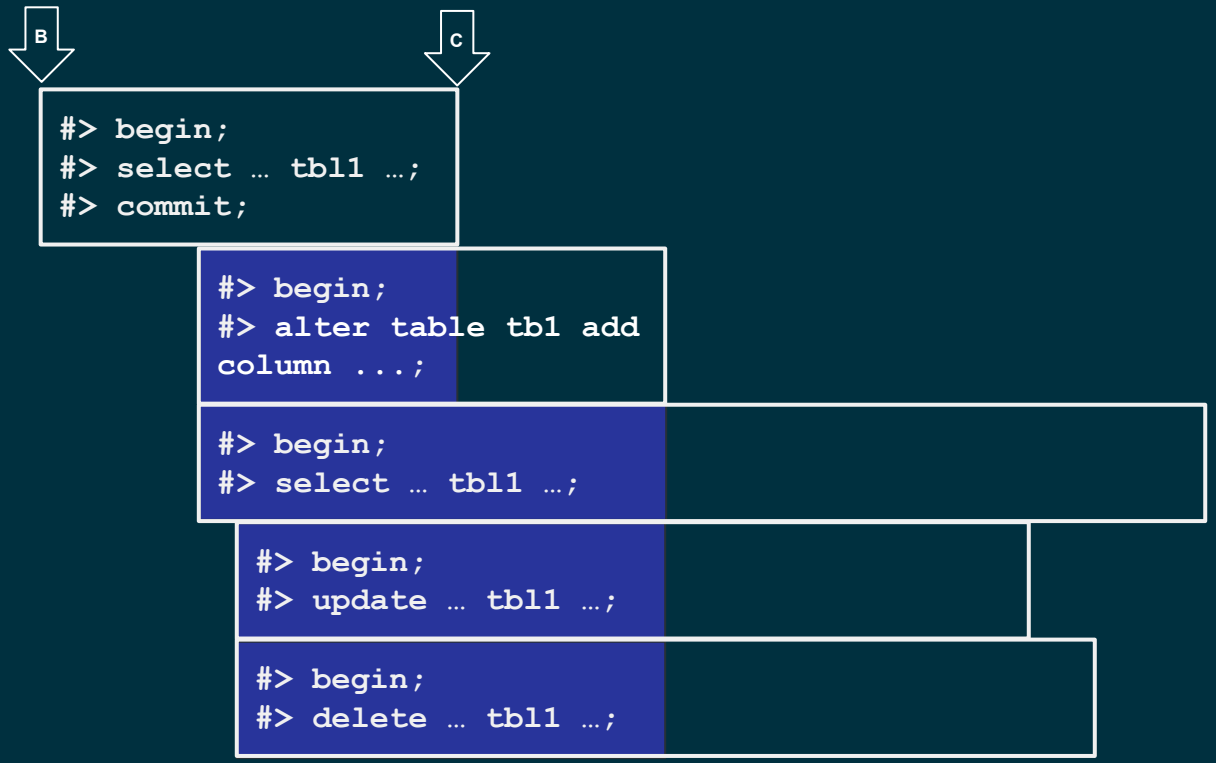
```
- processo 2 -
#> begin transaction;
#> select * from tb1 where id = 168 order by id; (wait aqui)

- processo N -
#> begin transaction;
#> select * from tb1 where id = 722 order by id; (wait aqui)

- processo 1 -
#> commit;
```

Trustly

timeline



Trustly

timeout

Parâmetros de config personalizados para a seção (conexão) do migrador:

```
set lock_timeout to '10s';

set statement_timeout to '45s';
```

Trustly

idle in transaction

Transações **explícitas em aberto**, impedem os **locks** necessários para a DDL:

```
select pg_cancel_backend(<pid>);

select pg_terminate_backend(<pid>);
```

alter table

```
-- Ok
alter table ... add column if not exists ativo boolean not null default true;
alter table ... add column if not exists verified boolean default false;

alter table ... add column if not exists auth_code varchar(10) default '-';
alter table ... alter column auth_code type varchar(40);

-- Ruim (vai ter reescrita e lock)
alter table ... alter column verified type char(1) using verified::char(1);

-- set not null
alter table ... alter column ... set not null;  (RUIM!!!)

alter table ... add constraint check ... (... is not null) not valid;
/* updates pra coluna sem valores nulos */
alter table ... validate constraint ...;
alter table ... alter column ... set not null;  (Agora ok por conta do check)
```

create index

Trustly

```
-- Ruim (vai ter escrita e lock)
create index ... on ... using ... (...);


-- fazer por fora da migration / antes do deploy
create index concurrently ... on ... using ... (...);  (Sem locks aqui)


-- na migration
create index if not exists ... on ... using ... (...);  (ok, sem erros)
```

Trustly

foreign key

```
-- Ruim (vai ter leitura e lock)
alter table ... add constraint ... foreign key (...) references ... (...);


-- fazer na migration criando como não válida
alter table ... add constraint ... foreign key (...) references ... (...) not
valid;  (ok, linhas novas terão a FK checada)


-- numa transação nova
alter table ... validade constraint ...;  (ok, lock mais permissivo)
```

Trustly

pk / unique

```
-- Ruim (vai ter escrita e lock)
alter table ... add constraint ... primary key (...);
alter table ... add constraint ... unique (...);


-- fazer por fora da migration / antes do deploy
create unique index concurrently ... on ... using ... (...);
(Sem locks aqui)


-- na migration
alter table ... add constraint ... primary key using index ...;
alter table ... add constraint ... unique using index ...;
(locks bem leves e rápidos)
```

Trustly

migration

```
begin;

set lock_timeout to '15s';
set statement_timeout to '30s';

alter table ...      --(todas as ddl pra mesma tabela juntas)
     add column if not exists ativo boolean not null default true,
     add column if not exists verified boolean default false,
     add constraint check ... (... is not null) not valid,
     add constraint ... foreign key (...) references ... (...) not valid;

create index if not exists ... on ... using ... (...);

alter table ... add constraint ... unique using index ...;

commit;
```

Transações explícitas longas

monitore / evite

Trustly

Obrigado

Trustly

LinkedIn