Data anonymization for iDempiere development environments

IWC 2019, Lyon

Murilo Habermann Torquato

About Me

Murilo Habermann Torquato

muriloht@devcoffee.com.br

- co-founder devCoffee
- ***piere world since 2008
- BrERP

Agenda

1. What is anonymization?
2. Scenario
3. Common Strategies
4. Data Anonymization
5. PostgreSQL Anonymizer
6. Conclusion

2

What is anonymization?

Just some concepts

What is anonymization?

Concept

What is anonymization ?

Modify a dataset to avoid any identification while remaining suitable for testing, data analysis and data processing.

Static x Dynamic Anonymization

- Dynamic Masking
  - altered view of the real data
  - controlled with users permissions
- Permanent Alteration
  - definitive action of substituting the sensitive information
  - the authentic data cannot be retrieved

Why is used?

Development, CI, testing, analytics, etc

What is anonymization?

Concept

It looks like easy, but...

- Singling out
  - The possibility to isolate a record and identify a subject in the dataset
- Linkability
  - Identify a subject in the dataset using other datasets
- Indirect Identifiers
  - When you use indirect information and are able to identify a subject: date of birth + gender + zip code

Cat and mouse game!

It doesn't matter how good is your job: you can't prove that your anonymized dataset is not useful and its re-identification is impossible!

1

Scenario

Why do we need it?

Scenario

LGPD

- LGPD / Brazil
  - General Data Protection Law (LGPD) is based on the European General Data Protection Regulation (GDPR)
  - approved by the government in August 2018
  - will become valid in August 2020
  - a set of rules for personal data collection, storage, processing and sharing

- one of our largest customers is managing > 100,000 collection contracts from a bank
  - first step is remove personal data for development, CI, functional testing, etc
    - what is personal data?
      - natural Person: name, ID, phone, email, address…

2

Common Strategies

A quick overview about strategies we used

Scenario

Common Strategies

- **Sampling** - is not anonymization, but implements data minimization from lgpd)
  - work only on 30% of a table
  - select * from c_bpartner tablesample bernoulli(30);
  - For more complex resultsets (referential integrity) take a look into pg_sample extension;

- **Suppression** - simple, break constraints and useless for testing
  - remove the data
  - UPDATE c_bpartner SET name = NULL;

- **Random Substitution** - simple, don't break constraints and also useless for testing
  - Update data with random value
  - UPDATE c_bpartner SET name = md5(random()::text);

- **Adding Noise -** only dates and numeric and may cause singling out
  - Randomly shifting the value of +/- 30%
  - Update c_bpartner set so_creditlimit = so_creditlimit * (1 + (2*random()-1) * 0.30) ;

Scenario

Common Strategies

- **Encryption -** you need to take care of your key and test is weird
  - generate a random salt and throw it away (may need a extension like pgcrypto)
  - UPDATE c_bpartner SET name = crypt('name' , gen_salt('md5'));
- **Shuffling -** don't break FK, meaningful dataset problem with boolean
  - Mix values within the same column - more complex query
- **Faking / Mocking -** hard to write functions that produces relevant synthetic data
  - replace data with random but legible values
  - UPDATE c_location SET address1 = z_fake_address();
- **Partial Suppression -** only for text/data and users can recognize your data
  - +55 19 993 247 735 will become XXX XX XX32477XX
  - UPDATE ad_user set phone = 'XXX XX XX' || substring(phone FROM 10 FOR 5) || 'XX';

Scenario

Common Strategies

- it takes time to produce an anonymized dataset useful and with a low risk of reidentification
- for the same dataset, you might need to use different strategies depending on the final destination of the data (development, CI, analytics, etc)

3

Data Anonymization

how i implement that?

Data Anonymization
Should I write from scratch?

- Ruby -> gems: faker, data::anonymization, CPF Faker
- Jailer
- Talend
- Bash Script

4

PostgreSQL Anonymizer

Our choice!

PostgreSQL Anonymizer

README.md

PostgreSQL Anonymizer

Anonymization & Data Masking for PostgreSQL

`postgresql_anonymizer` is an extension to mask or replace personally identifiable information (PII) or commercially sensitive data from a PostgreSQL database.

The projet is aiming toward a **declarative approach** of anonymization. This means we're trying to extend PostgreSQL Data Definition Language (DDL) in order to specify the anonymization strategy inside the table definition itself.

Once the maskings rules are defined, you can access the anonymized data in 3 different ways :

- Anonymous Dumps : Simply export the masked data into an SQL file
- In-Place Anonymization : Remove the PII according to the rules
- Dynamic Masking : Hide PII only for the masked users

In addition, various Masking Functions are available : randomization, faking, partial scrambling, shufflin, noise or even your own custom function !

Read the Concepts section for more details and NEWS.md for information about the latest version.

Reference: https://gitlab.com/dalibo/postgresql_anonymizer

PostgreSQL Anonymizer

Declaring The Masking Rules

The main idea of this extension is to offer **anonymization by design**.

The data masking rules should be written by the people who develop the application because they have the best knowledge of how the data model works. Therefore masking rules must be implemented directly inside the database schema.

This allows to mask the data directly inside the PostgreSQL instance without using an external tool and thus limiting the exposure and the risks of data leak.

The data masking rules are declared simply by using security labels :

```
=# CREATE EXTENSION IF NOT EXISTS anon CASCADE;

=# SELECT anon.load();

=# CREATE TABLE player( id SERIAL, name TEXT, points INT);

=# SECURITY LABEL FOR anon ON COLUMN player.name
-# IS 'MASKED WITH FUNCTION anon.fake_last_name()';
```

Reference: https://gitlab.com/dalibo/postgresql_anonymizer

PostgreSQL Anonymizer

Highlights of the extension

- Transform data inside PostgreSQL
- Implement useful features (noise, shuffling, faking, etc.)
- Define anonymization policy with SQL statements
- PoC for Dynamic Masking

Declarative Dynamic masking

- Regular user can see the real data
- Others can only view anonymized data

Limits

- PostgreSQL 9.6 and newest
- Only one schema

5

Conclusion

My last comments

Conclusion

- Faster than other options (scripts, etc)
- Ideal for our use case (development environment in cloud)

NOT JUST ABOUT GDPR....

Free software communities must lead the way to build a future where privacy and anonymity are available to everyone. And of course PostgreSQL has an important role to in this domain because it's by far the wolrd's most dynamic and innovative database engine.

Demien Clochard, pg_anonymizer creator

Any Questions?

Thank you!

muriloht@devcoffee.com.br

devCoffee
SISTEMAS DE GESTÃO EMPRESARIAL

BrERP
SISTEMA DE GESTÃO EMPRESARIAL

Follow us:

@devcoffee.br

/DevcoffeeBr

/devcoffeebr

/devcoffee

References

https://www.pwc.com.br/pt/sala-de-imprensa/artigos/lgpd-muda-pratica-plc-53.html

https://ecoit.com.br/protecao-de-dados-pessoais/

https://blog.taadeem.net/english/2018/10/29/Introducing-PostgreSQL-Anonymizer

https://github.com/mla/pg_sample

https://github.com/faker-ruby/faker

https://github.com/smithoss/gonymizer

https://www.talend.com/resources/anonymize-data/

http://jailer.sourceforge.net/faq.html

https://gitlab.com/dalibo/postgresql_anonymizer

https://www.postgresql.eu/events/fosdem2019/sessions/session/2287/slides/151/postgresql_anonymizer.reveal..pdf

https://wiki.postgresql.org/images/1/1e/PGConf.Brasil_2019_-_Christiane_Faleiro.pdf