GDPRizer

Retrofitting GDPR Compliance onto Legacy Databases

Archita Agarwal, Marilyn George, Aaron Jeyaraj, Malte Schwarzkopf

mongoDB. mongoDB. BROWN BROWN
Data Privacy Laws

• EU’s GDPR
• California’s CCPA
• Virginia’s VCDPA
• Japan’s APPI
• Canada’s PIPEDA
• ...

GDPR by elasticstudio from Noun Project
Data Privacy Laws

- EU’s GDPR
- California’s CCPA
- Virginia’s VCDPA
- Japan’s APPI
- Canada’s PIPEDA
- ...
Identifying & retrieving user-data is hard

Tried the GDPR data export from Spotify. By default, you get like 6 JSON files with almost nothing. After many emails and complaining and a month of waiting, I got a 250MB archive with basically EVERY INTERACTION I ever did with any Spotify client, all my searches. Everything.
Identifying & retrieving user-data is hard

Tried the GDPR data export from Spotify. By default, you get like 6 JSON files with almost nothing. After many emails and complaining and a month of waiting, I got a 250MB archive with basically EVERY INTERACTION I ever did with any Spotify client, all my searches. Everything.
Why is user-data identification so hard?
Why is user-data identification so hard?

• Legacy systems are not built keeping regulations in mind
Why is user-data identification so hard?

• Legacy systems are not built keeping regulations in mind

• User-data distributed across tables
Why is user-data identification so hard?

• Legacy systems are not built keeping regulations in mind
  • User-data distributed across tables
  • Complex relationships between tables
Why is user-data identification so hard?

• Legacy systems are not built keeping regulations in mind
  • User-data distributed across tables
  • Complex relationships between tables

How to identify a user’s information?
How to identify a user’s information?

Fully Manual

DBAs identify and write the queries
How to identify a user’s information?

Too HARD :-(

Fully Manual

DBAs identify and write the queries
How to identify a user’s information?

Too HARD :-(

Fully Manual

DBAs identify and write the queries

Generic

Fully Automated
How to identify a user’s information?

Too HARD :-(

Fully Manual

DBAs identify and write the queries

Likely Impossible :-(

Generic

Fully Automated
How to identify a user’s information?

Need to make application-specific policy choices

Fully Manual

DBAs identify and write the queries

Generic Fully Automated

Likely Impossible :-(

Too HARD :-(

Fully Manual
How to identify a user’s information?

Need to make application-specific policy choices

e.g: TPCH: customers vs suppliers

Fully Manual

DBAs identify and write the queries

Generic

Fully Automated

 Likely Impossible :-(
How to identify a user’s information?

Need to make application-specific policy choices

- e.g: TPCH: customers vs suppliers
- e.g: Should comments on posts be returned to the author?

Fully Manual

DBAs identify and write the queries

Likely Impossible :-(

Generic

Fully Automated
How to identify a user’s information?

- **Fully Manual**: DBAs identify and write the queries.
- **Mostly Automated w/ some Manual Customizations**: Likely Impossible :-(
- **Generic Fully Automated**: GDPRizer

- Too HARD :-(
- Likely Impossible :-(

Automated Solution by ProSymbols from the Noun Project
Hard Work by Llisole from the Noun Project
Talk Outline

• GDPRizer: Design & Architecture
• Experimental Evaluation
  • Prototype in Python
  • Tested its accuracy on four applications
Talk Outline

• GDPRizer: Design & Architecture

  • Experimental Evaluation

    • Prototype in Python

    • Tested its accuracy on four applications
High Level Design of GDPRizer
High Level Design of GDPRizer
High Level Design of GDPRizer
High Level Design of GDPRizer

Relationship Graph

Encodes data dependencies across tables

Data Access Request

Data
Relationship Graph

Encodes data dependencies across tables
Relationship Graph

Encodes data dependencies across tables

Explicit foreign-key constraints
Relationship Graph

Encodes data dependencies across tables

Joins in Queries
Relationship Graph

Queries

Schema

Encodes data dependencies across tables

Joins in Queries

SELECT * FROM Paper, ContactInfo
WHERE
Paper.leadContactId = ContactInfo.contactId
Relationship Graph

Encodes data dependencies across tables
Relationship Graph

Encodes data dependencies across tables

Rich literature on identifying functional dependencies in data

See survey by Abedjan et al., VLDB 2015
Relationship Graph of HotCRP
Using only the joins in queries
Service Data Access Request

Data Access Request

Schema
Queries
Cues from data itself
Service Data Access Request

Schema
Queries
Cues from data itself

Data Access Request

Graph Traversal
Service Data Access Request

Schema
Queries
Cues from data itself

Graph Traversal

SELECT * FROM ContactInfo WHERE contactId = 10
SELECT * FROM Paper WHERE leadContactId = 10
SELECT * FROM PaperComment WHERE contactId = 10
Graph Traversal: Access Request for contactID = 10

BFS-like: visits closer vertices before visiting farther ones
Graph Traversal: Access Request for contactID = 10
Graph Traversal: Access Request for contactID = 10

Start at the primary table
Graph Traversal: Access Request for contactID = 10

SELECT * FROM ContactInfo WHERE contactId = 10

Q1: Extract contact info of user 10
Graph Traversal: Access Request for contactID = 10

Q1: Extract contact info of user 10

SELECT * FROM ContactInfo WHERE contactId = 10

Q2: Extract all the papers user 10 wrote

SELECT * FROM Paper WHERE LeadContactId in {10}
Graph Traversal: Access Request for contactID = 10

Q1: Extract contact info of user 10

```
SELECT * FROM ContactInfo WHERE contactId = 10
```

Q2: Extract all the papers user 10 wrote

```
SELECT * FROM Paper WHERE LeadContactId in {10}
```

Q3: Extract all the comments of user 10

Q4: Extract all the reviews user 10 made

```
SELECT * FROM PaperReview WHERE ContactId = 10
```
Q2: Extract all the papers user 10 wrote

SELECT * FROM Paper
WHERE LeadContactId in {10}
Graph Traversal: Access Request for contactID = 10

Q2: Extract all the papers user 10 wrote

SELECT * FROM Paper
WHERE LeadContactId in {10}
Graph Traversal: Access Request for contactID = 10

Q2: Extract all the papers user 10 wrote

SELECT * FROM Paper
WHERE LeadContactId in {10}

Extract all the reviews user 10 received on their papers
Graph Traversal: Access Request for contactID = 10

Q2: Extract all the papers user 10 wrote

- SELECT * FROM Paper
- WHERE LeadContactId in {10}

Extract all the reviews user 10 received on their papers

- ContactId of the reviewer
Customizations

Output

Column Filtering
Customizations
Customizations

1. Column Filtering

1. Edge Pruning

Graph

1. Output

2. Column Filtering

3. Edge Pruning

CONTACT INFO

- CONTACT ID

TOPIC INTEREST

- CONTACT ID

PAPER

- LEAD CONTACT ID
- PAPER ID
- SHEPHERD CONTACT ID

PAPER COMMENT

- PAPER ID
- CONTACT ID

PAPER REVIEW

- PAPER ID
Customizations

1. Column Filtering
2. Edge Pruning
3. Edge Addition
4. Column Addition

Graph:

- CONTACT INFO
- PAPER REVIEW
- TOPIC INTEREST
- CONTACT ID
- LEAD CONTACT ID
- SHEPHERD CONTACT ID
- AUTHOR 2
- PAPER ID
- PAPER COMMENT
- CONTACT ID

Edits:
- Edge Pruning
- Column Addition
GDPRizer: Architecture

Schema Query Cues from data itself Customizations

Data Access Request

Data
Talk Outline

• GDPRizer: Design & Architecture

• Experimental Evaluation
  • Prototype in Python
  • Tested its accuracy on four applications
Experimental Evaluation
Experimental Evaluation

Q1: Does GDPRizer correctly identify user-data?
Experimental Evaluation

Q1: Does GDPRizer correctly identify user-data?

Q2: What is the impact of customizations?
Experimental Evaluation

Q1: Does GDPRizer correctly identify user-data?

Q2: What is the impact of customizations?

Q3: How many customizations are needed?
Experimental Evaluation

Q1: Does GDPRizer correctly identify user-data?

Q2: What is the impact of customizations?

Q3: How many customizations are needed?

Q4: How does GDPRizer compare to third-party plug-ins?
Experimental Evaluation

Q1: Does GDPRizer correctly identify user-data?

Q2: What is the impact of customizations?

Q3: How many customizations are needed?

Q4: How does GDPRizer compare to third-party plug-ins?
# Experimental Evaluation

<table>
<thead>
<tr>
<th>Q1:</th>
<th>Does GDPRizer correctly identify user-data?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2:</td>
<td>What is the impact of customizations?</td>
</tr>
<tr>
<td>Q3:</td>
<td>How many customizations are needed?</td>
</tr>
<tr>
<td>Q4:</td>
<td>How does GDPRizer compare to third-party plug-ins?</td>
</tr>
</tbody>
</table>

1. TPC-H
2. Lobsters
3. HotCRP
4. WordPress
Q1: Does GDPRizer correctly identify user-data?
Q1: Does GDPRizer correctly identify user-data?

Ground Truth
Wrote our own ground truth queries
Q1: Does GDPRizer correctly identify user-data?
Q1: Does GDPRizer correctly identify user-data?

- Precision:
- Recall:
- F1-Score:
Q1: Does GDPRizer correctly identify user-data?

- **Precision**: Measures what fraction of what GDPRizer extracted was actually user-data.

- **Recall**:

- **F1-Score**:

  😊 100%  extracted only your data

  😞 0%   extracted only other people's data
Q1: Does GDPRizer correctly identify user-data?

- **Precision**: Measures what fraction of what GDPRizer extracted was actually user-data
- **Recall**: Measures what fraction of the user-data did GDPRizer manage to extract
- **F1-Score**:

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☺ 100 %</td>
<td>☺ 100 %</td>
<td>extracted all your data</td>
</tr>
<tr>
<td>☹ 0 %</td>
<td>☹ 0 %</td>
<td>extracted only your data</td>
</tr>
<tr>
<td>☹ 0 %</td>
<td>☺ 100 %</td>
<td>extracted only other people’s data</td>
</tr>
<tr>
<td>☺ 100 %</td>
<td>☹ 0 %</td>
<td>did not extract any of your data</td>
</tr>
</tbody>
</table>
Q1: Does GDPRizer correctly identify user-data?

- **Precision**: Measures what fraction of what GDPRizer extracted was actually user-data
- **Recall**: Measures what fraction of the user-data did GDPRizer manage to extract
- **F1-Score**: Combination of precision and recall

<table>
<thead>
<tr>
<th>Precision</th>
<th>100 %</th>
<th>extracted only your data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>100 %</td>
<td>extracted all your data</td>
</tr>
<tr>
<td></td>
<td>0 %</td>
<td>extracted only other people’s data</td>
</tr>
<tr>
<td></td>
<td>0 %</td>
<td>did not extract any of your data</td>
</tr>
</tbody>
</table>
Q1: Does GDPRizer correctly identify user-data?

HotCRP

![Graph showing Precision, Recall, and F1 measure for HotCRP]

- Precision
- Recall
- F1 measure

- $R^Q/R^S,Q$ only
- + filtering
- + pruning
- + col addition
- + manual edges
Q1: Does GDPRizer correctly identify user-data?

HotCRP

![Graph showing Precision, Recall, and F1 measure for different data processing steps: R^Q/R^S,Q only, + filtering, + pruning, + col addition, + manual edges.](image-url)
Q2: What is the impact of customizations?

**HotCRP**

![Graph showing the impact of customizations on precision, recall, and F1 measure.](image)

- **Precision**: 0, 20, 40, 60, 80, 100 percent [%]
- **Recall**: 0, 50, 100 percent [%]
- **F1 measure**: (not clearly visible)

- **$R^Q/R^{S,Q}$ only**
- **+ filtering**
- **+ pruning**
- **+ col addition**
- **+ edge addition**
Q2: What is the impact of customizations?

HotCRP

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
<th>F1 measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>Recall</td>
<td>F1 measure</td>
</tr>
</tbody>
</table>

- \( R^Q/R^{S,Q} \) only
- \(+\) filtering
- \(+\) pruning
- \(+\) col addition
- \(+\) edge addition
Q2: What is the impact of customizations?

HotCRP

![Graph showing the impact of customizations on Precision, Recall, and F1 measure.]

- $R^Q/R^{S,Q}$ only
- + filtering
- + pruning
- + col addition
- + edge addition
Q2: What is the impact of customizations?

HotCRP

![Graph showing the impact of customizations on Precision, Recall, and F1 measure.](image)
Q2: What is the impact of customizations?

HotCRP

Similar results for all the other applications
Q5: How many customizations are needed?

<table>
<thead>
<tr>
<th></th>
<th>Total number of customizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPC-H (customer)</td>
<td>4</td>
</tr>
<tr>
<td>TPC-H (supplier)</td>
<td>7</td>
</tr>
<tr>
<td>HotCRP</td>
<td>31</td>
</tr>
<tr>
<td>Lobsters</td>
<td>16</td>
</tr>
<tr>
<td>WordPress</td>
<td>4</td>
</tr>
<tr>
<td>WordPress (w/ plugins)</td>
<td>12</td>
</tr>
</tbody>
</table>
Impact of different sources of information
Impact of different sources of information

• More reliable sources of information
• better relationship graph
• fewer customizations
Impact of different sources of information

- More reliable sources of information
  - better relationship graph
  - fewer customizations

- In our experience,
  - Foreign Keys in Schema > Joins in Queries > Data itself
Conclusion

- GDPRizer: a tool for user-data extraction in legacy databases
Conclusion

• GDPRizer: a tool for user-data extraction in legacy databases

• A fully-automated, general solution for legacy systems is unlikely
Conclusion

• GDPRizer: a tool for user-data extraction in legacy databases

• A fully-automated, general solution for legacy systems is unlikely

• Mostly automates user-data identification but still requires some manual input
Questions?