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 elasticıstudio from Noun Project

Data Privacy Laws

- EU’s GDPR
- California’s CCPA
- Virginia’s VCDPA
- Japan’s APPI
- Canada’s PIPEDA
- …

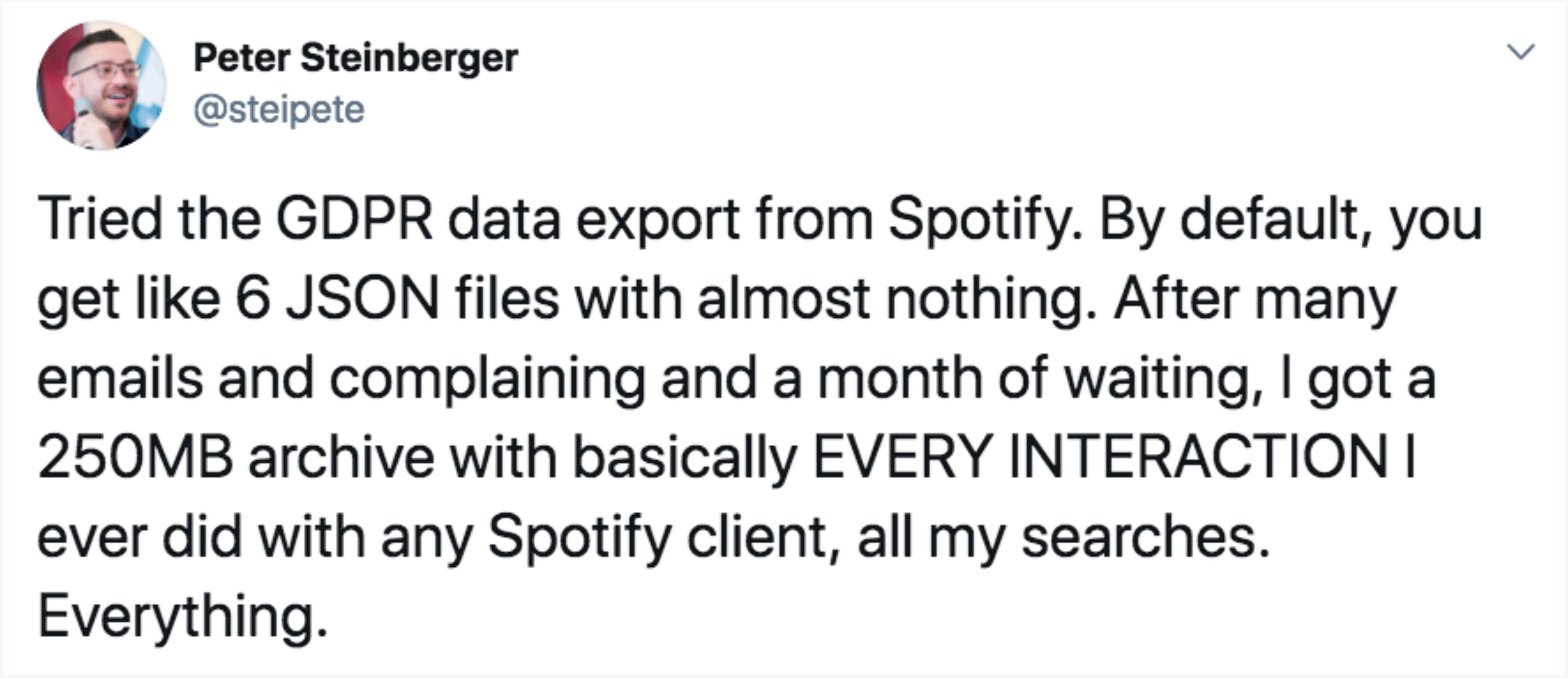
Allow individuals to
request a copy of their data

data access request

GDPR by elasticıstudio from Noun Project

Identifying & retrieving user-data is hard

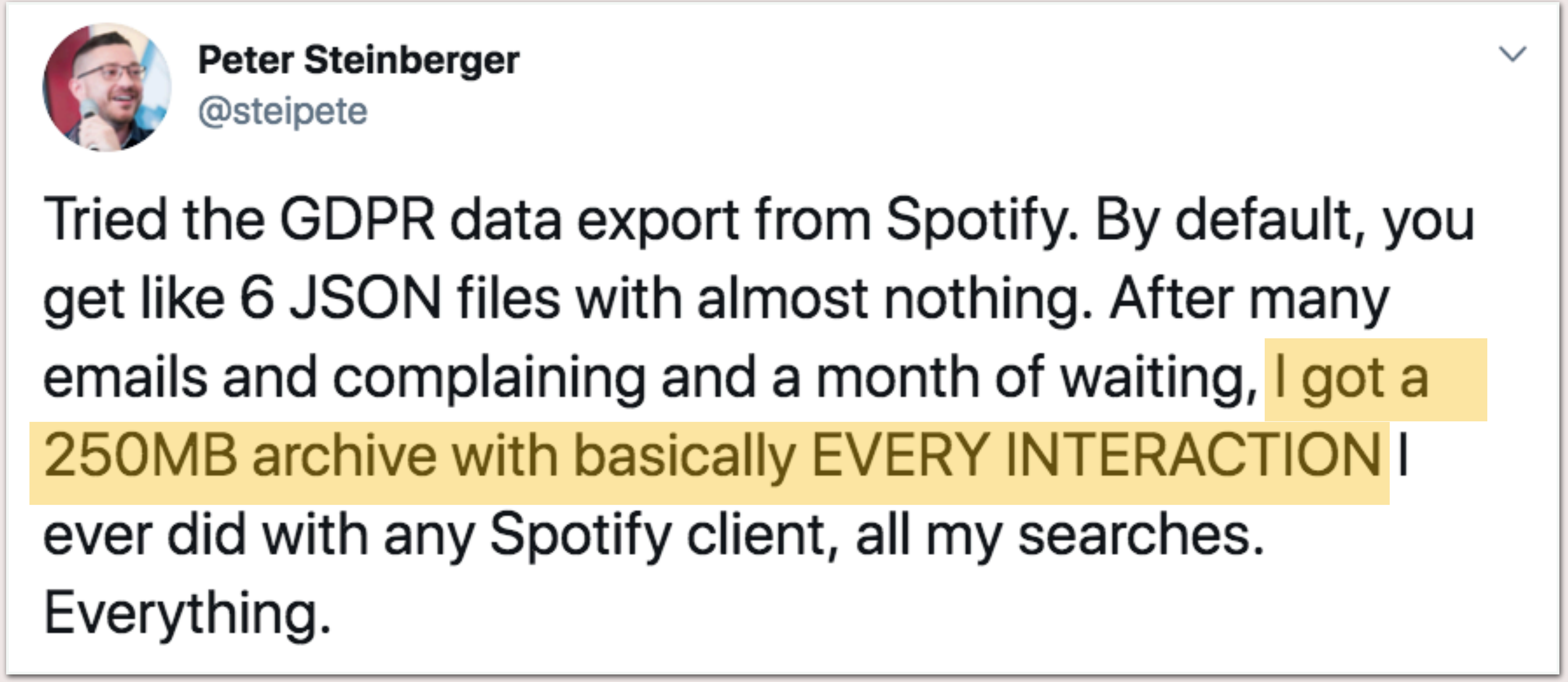
Spotify®



Peter Steinberger
@steipete

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



Peter Steinberger
@steipete

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

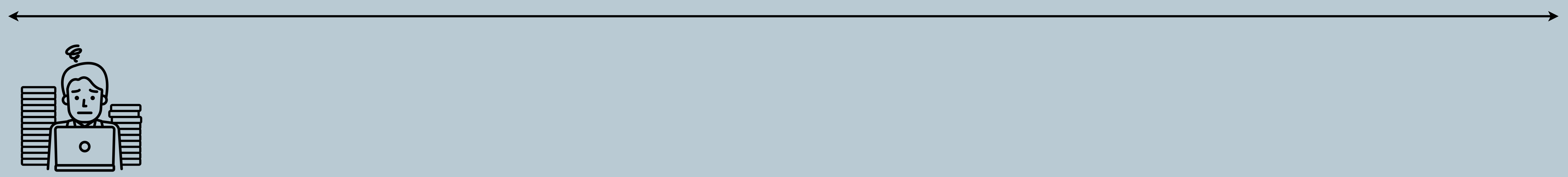
Why is user-data identification so hard?

- 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



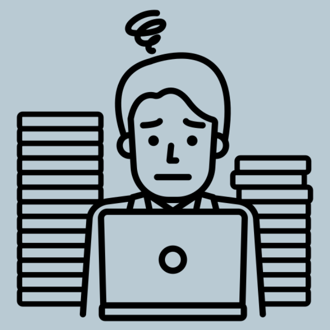
DBAs identify and write the queries

Hard Work by Llisole from the Noun Project

How to identify a user's information?

Too HARD :-(

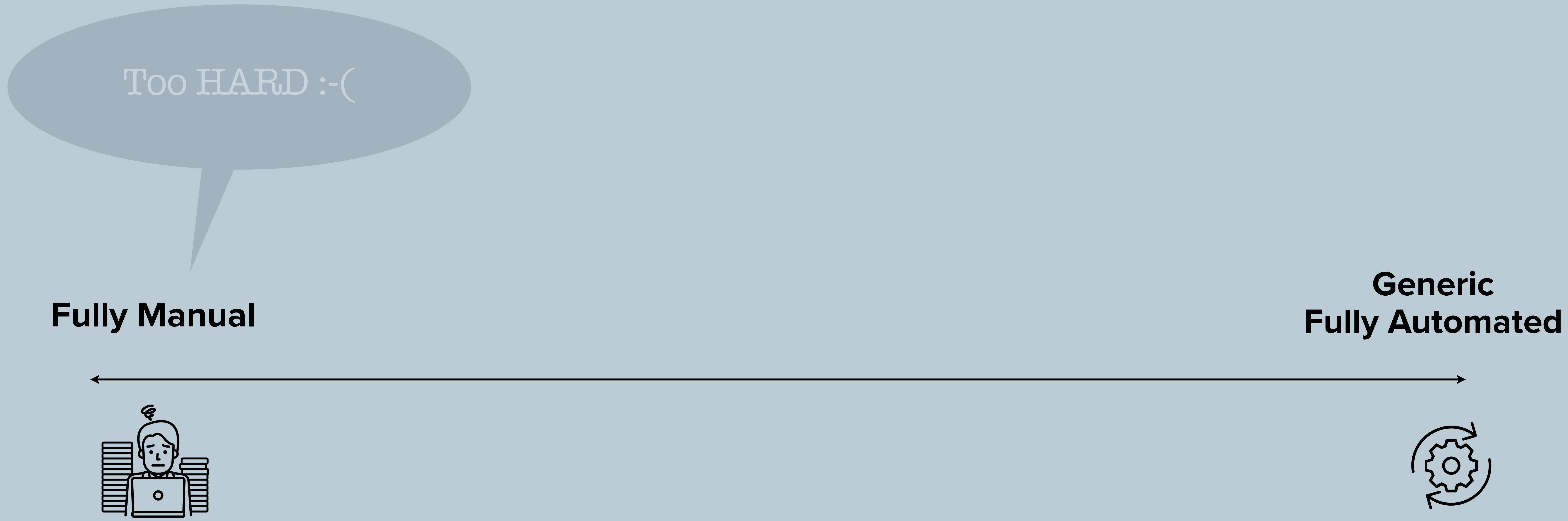
Fully Manual



DBAs identify and write the queries

Hard Work by Llisole from the Noun Project

How to identify a user's information?



Too HARD :-(

Fully Manual

Generic
Fully Automated

DBAs identify and write
the queries

Hard Work by Llisole from the Noun Project

Automated Solution by ProSymbols from the Noun Project

How to identify a user’s information?

Too HARD :-(

Fully Manual

Likely Impossible :-(

Generic
Fully Automated

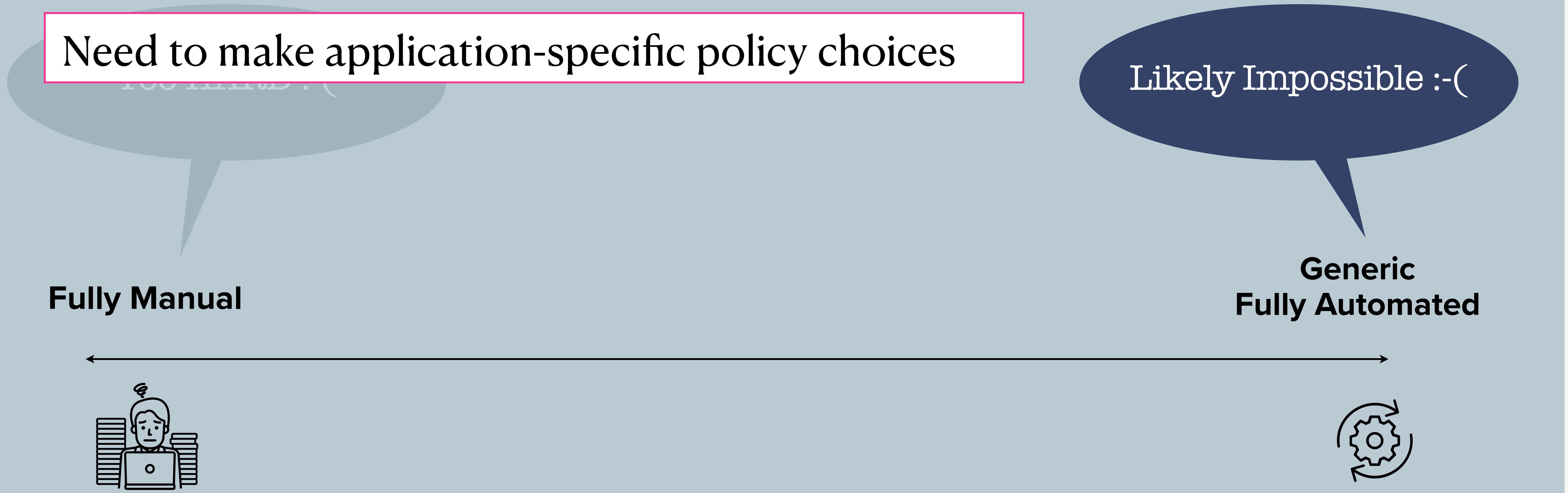
DBAs identify and write the queries

Hard Work by Llisole from the Noun Project

Automated Solution by ProSymbols from the Noun Project

How to identify a user's information?

Need to make application-specific policy choices



Likely Impossible :-(

Fully Manual

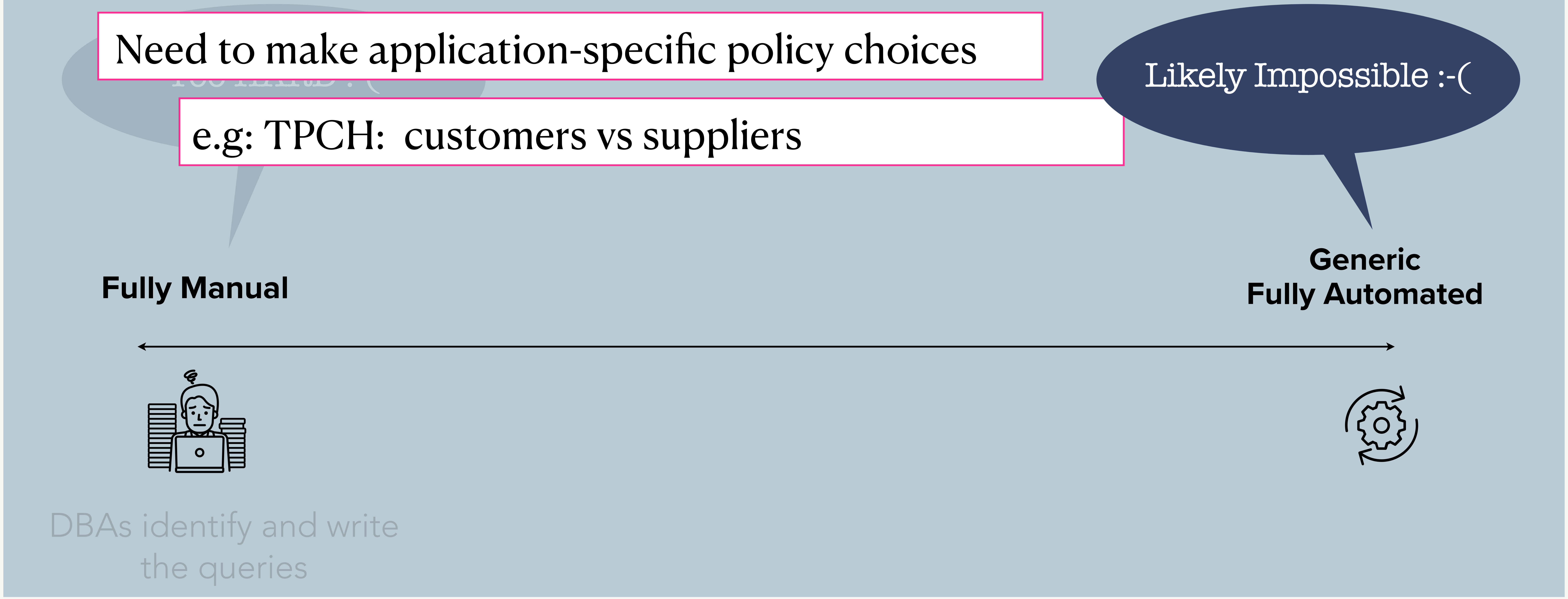
Generic
Fully Automated

DBAs identify and write
the queries

Hard Work by Llisole from the Noun Project

Automated Solution by ProSymbols from the Noun Project

How to identify a user’s information?



Need to make application-specific policy choices

e.g: TPCH: customers vs suppliers

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

How to identify a user’s information?

Need to make application-specific policy choices

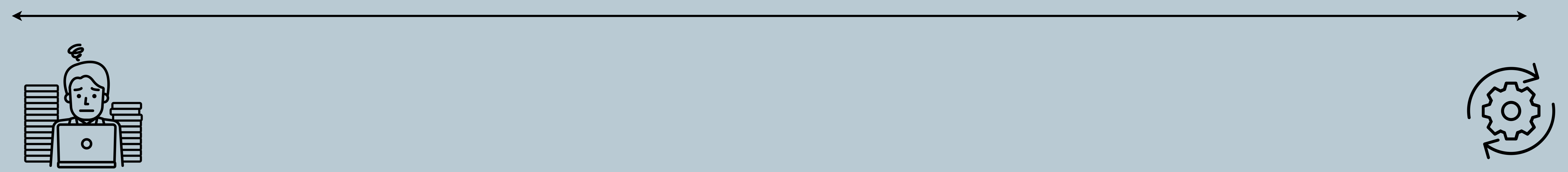
e.g: TPCH: customers vs suppliers

e.g: Should comments on posts be returned to the author?

Likely Impossible :-(

Fully Manual

**Generic
Fully Automated**



DBAs identify and write
the queries

Hard Work by Llisole from the Noun Project

Automated Solution by ProSymbols from the Noun Project

How to identify a user's information?

Too HARD :-(

Likely Impossible :-(

GDPRizer

Fully Manual

Mostly Automated
w/ some Manual Customizations

Generic
Fully Automated

DBAs identify and write
the queries

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

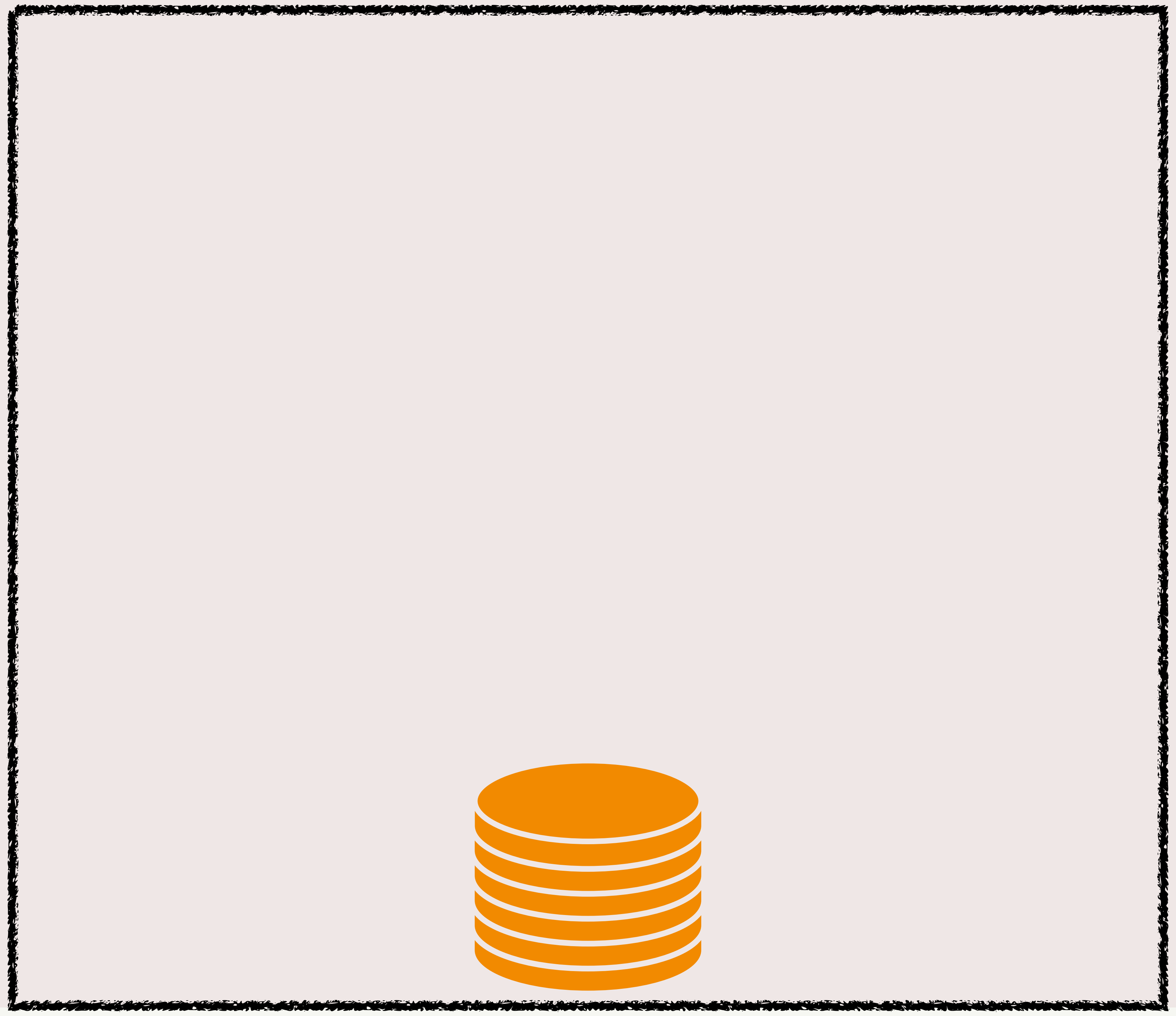
Talk Outline

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

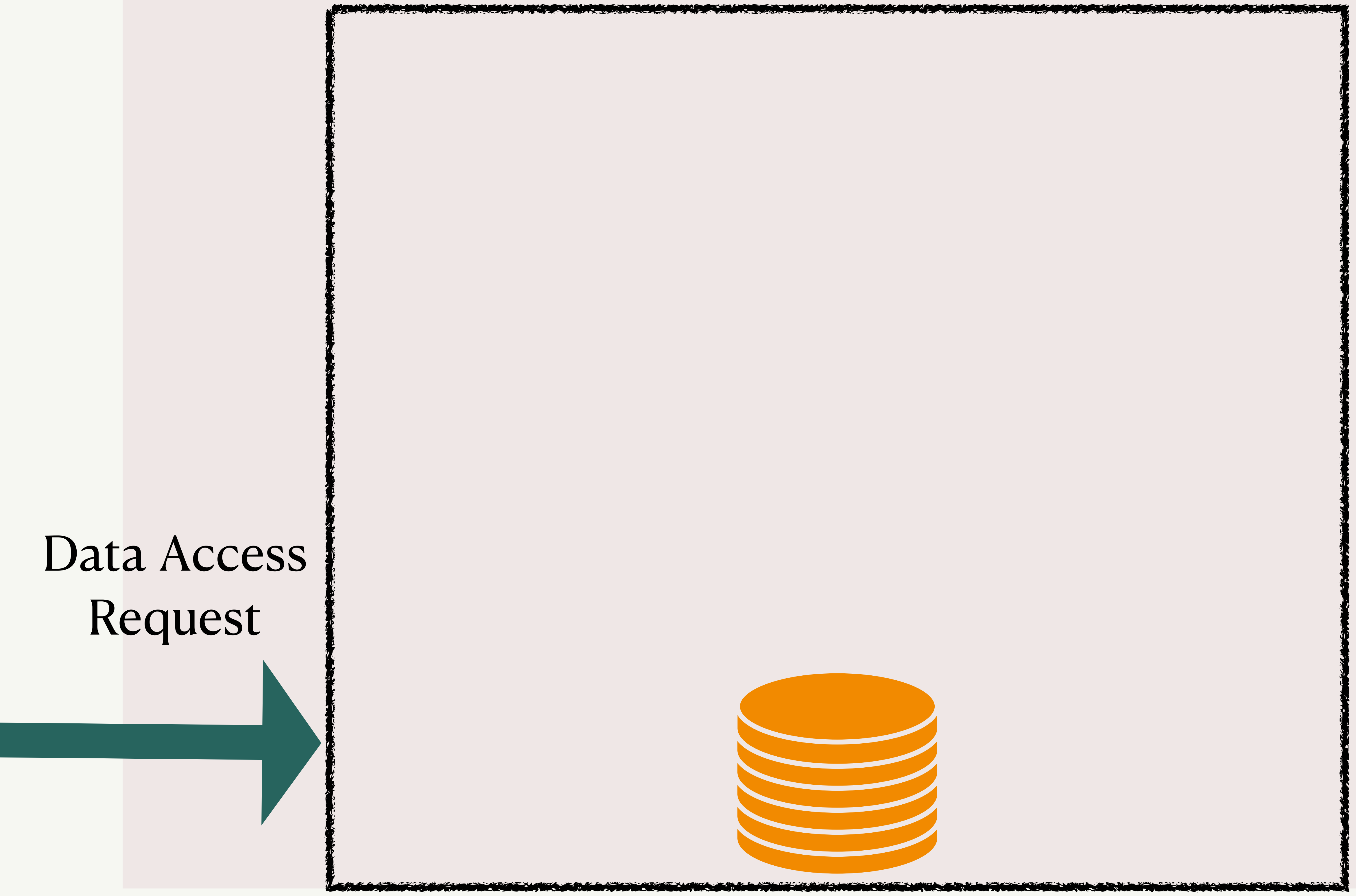
Talk Outline

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

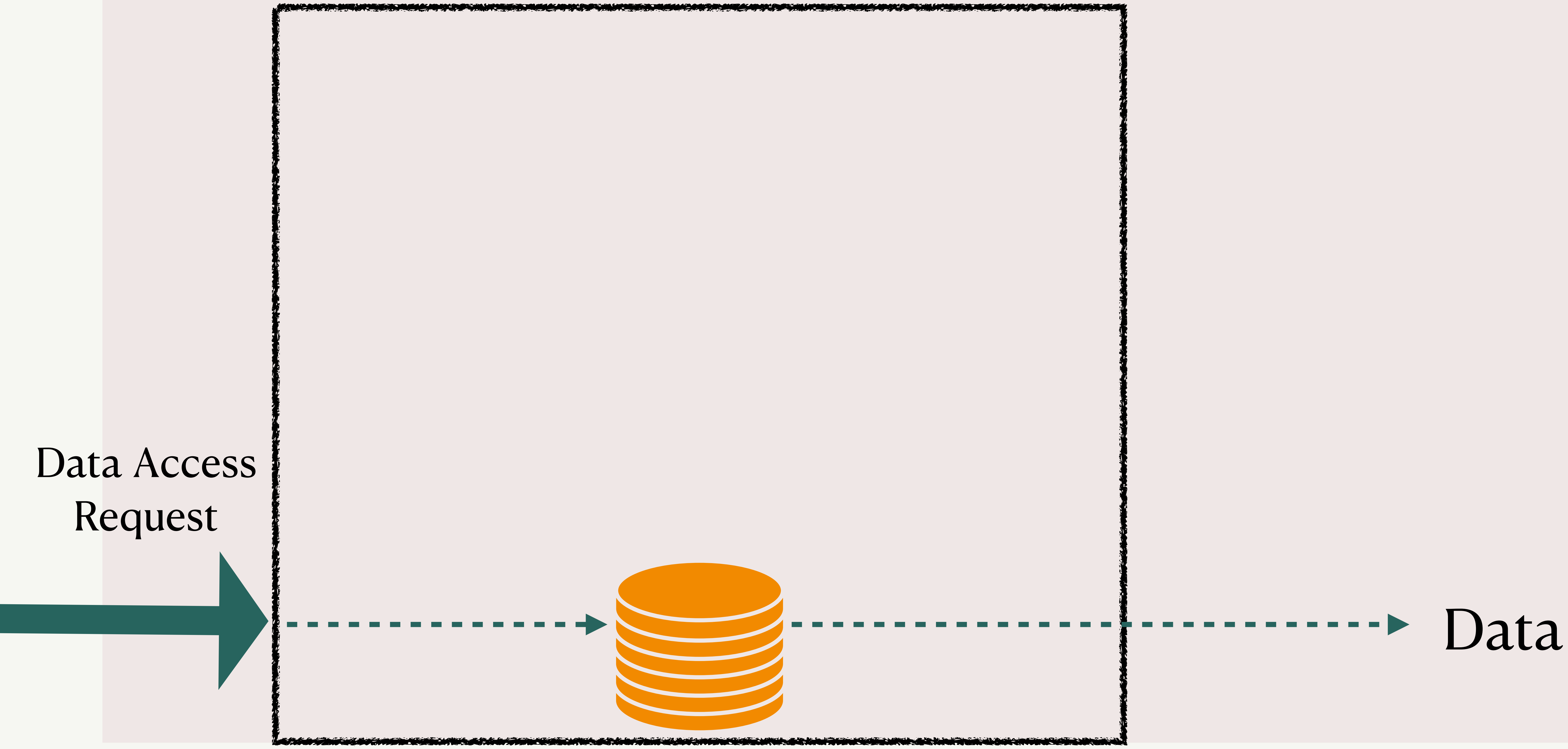
High Level Design of GDPRizer



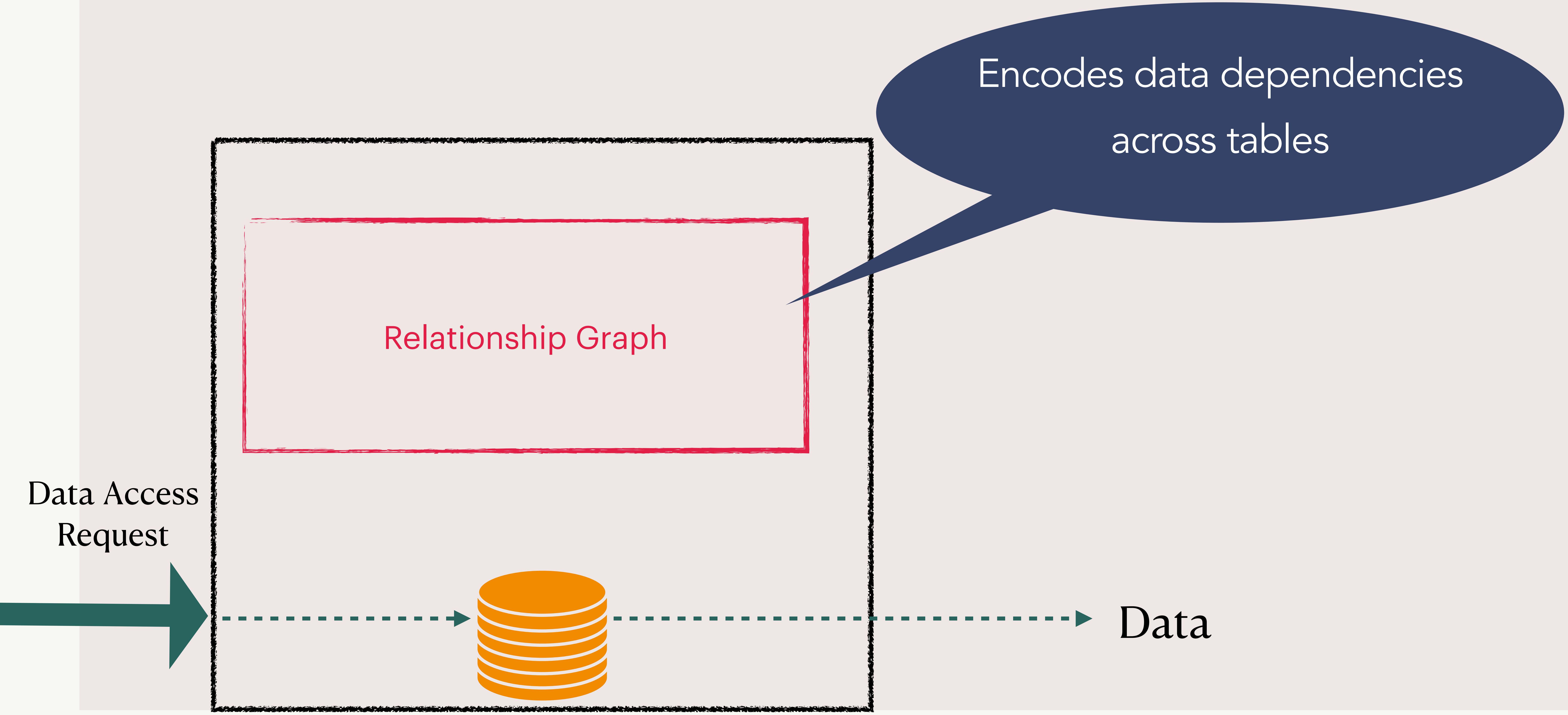
High Level Design of GDPRizer



High Level Design of GDPRizer

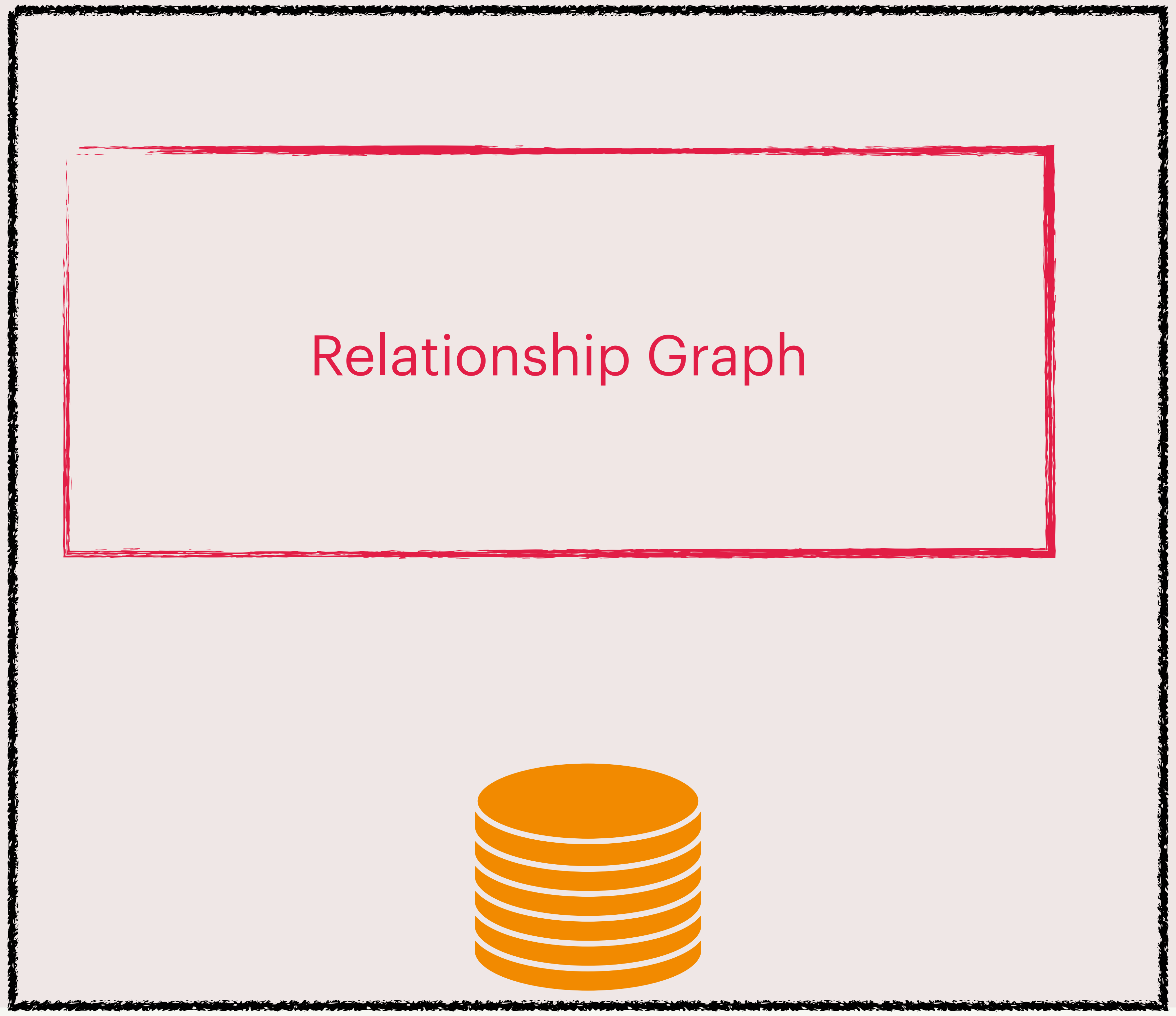


High Level Design of GDPRizer



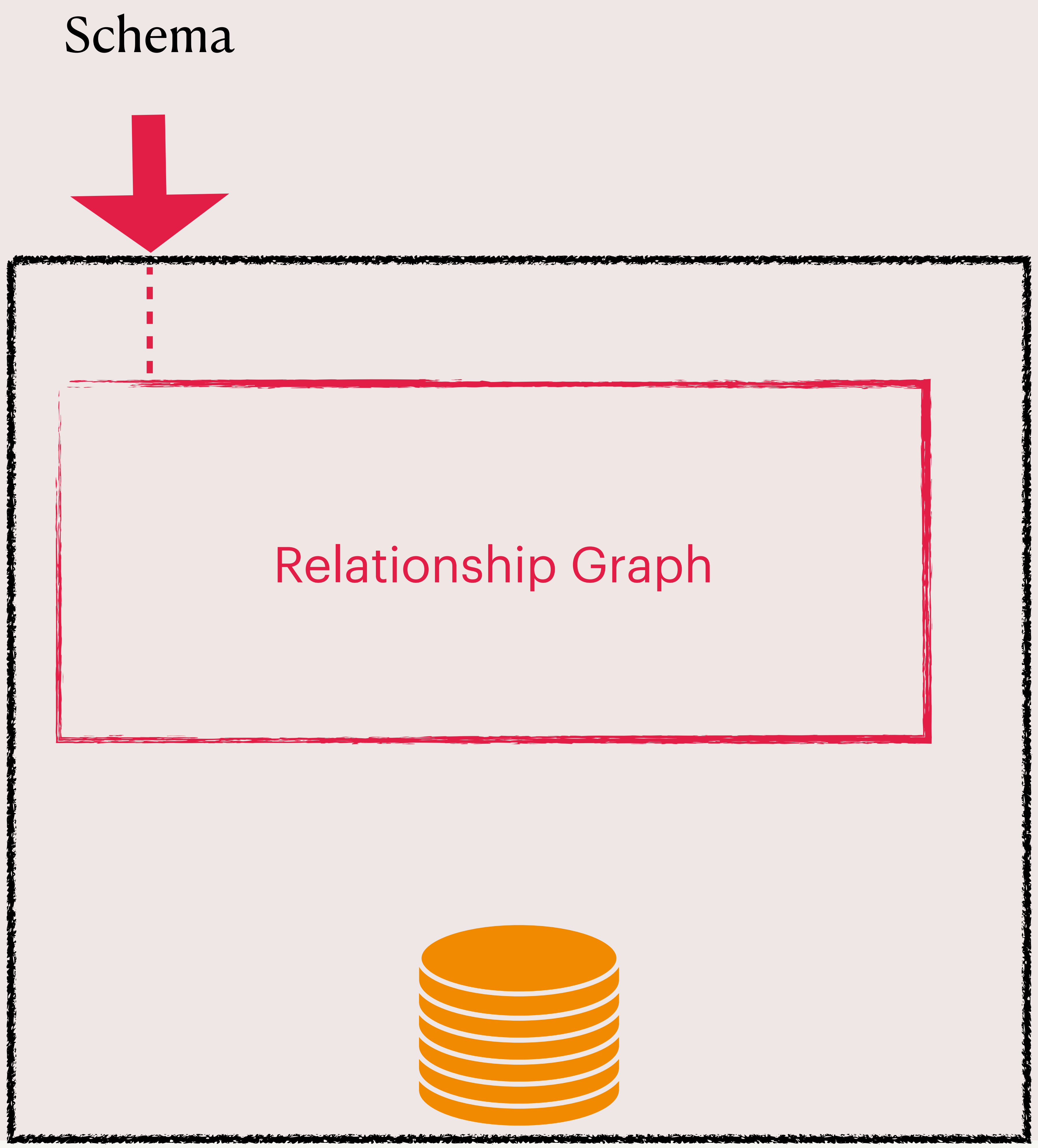
Relationship Graph

Encodes data dependencies across tables



Relationship Graph

Relationship Graph



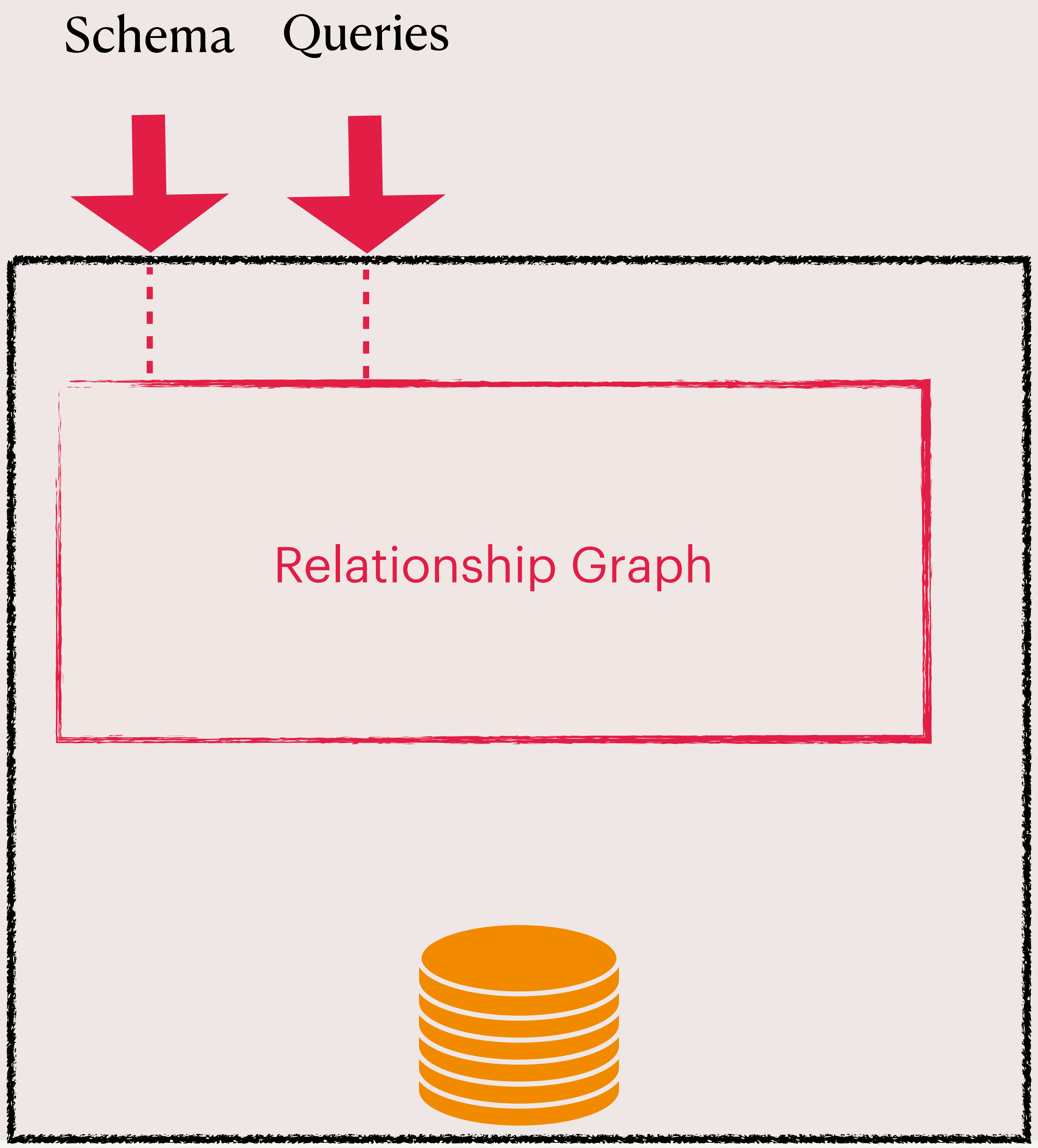
Schema

Relationship Graph

Encodes data dependencies across tables

Explicit foreign-key constraints

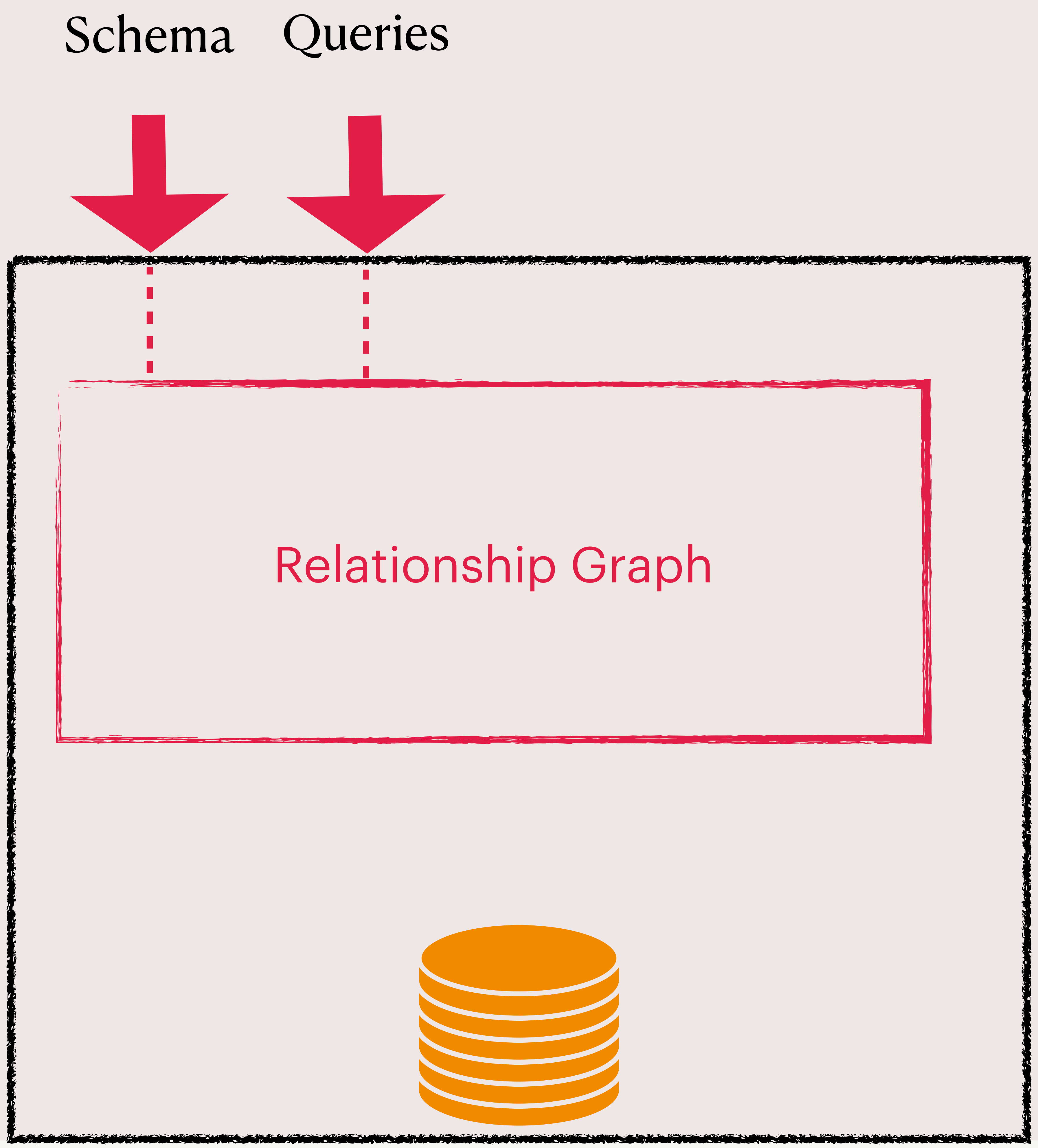
Relationship Graph



Encodes data dependencies across tables

Joins in Queries

Relationship Graph



Encodes data dependencies across tables

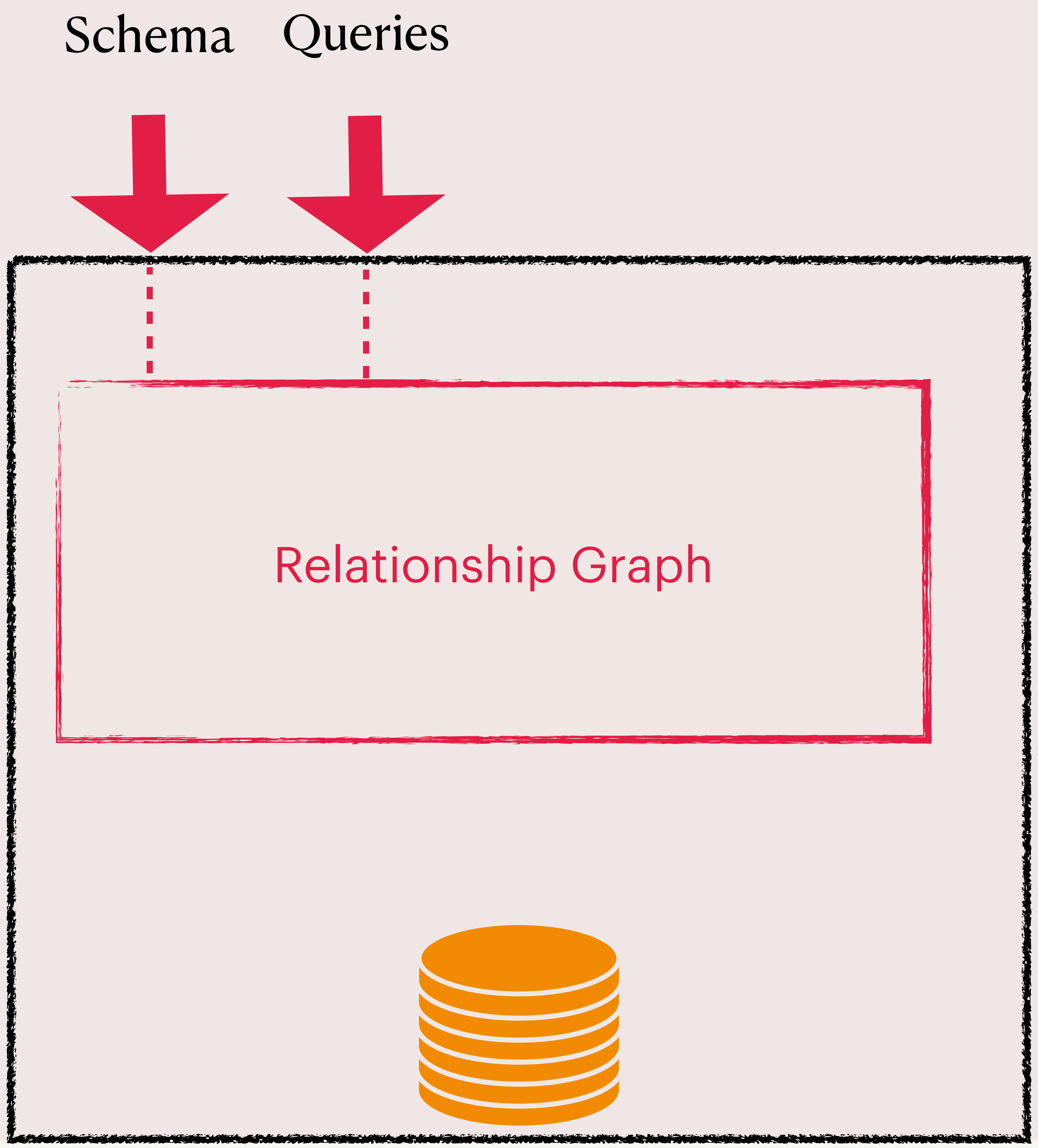
Joins in Queries

SELECT * FROM Paper, ContactInfo

WHERE

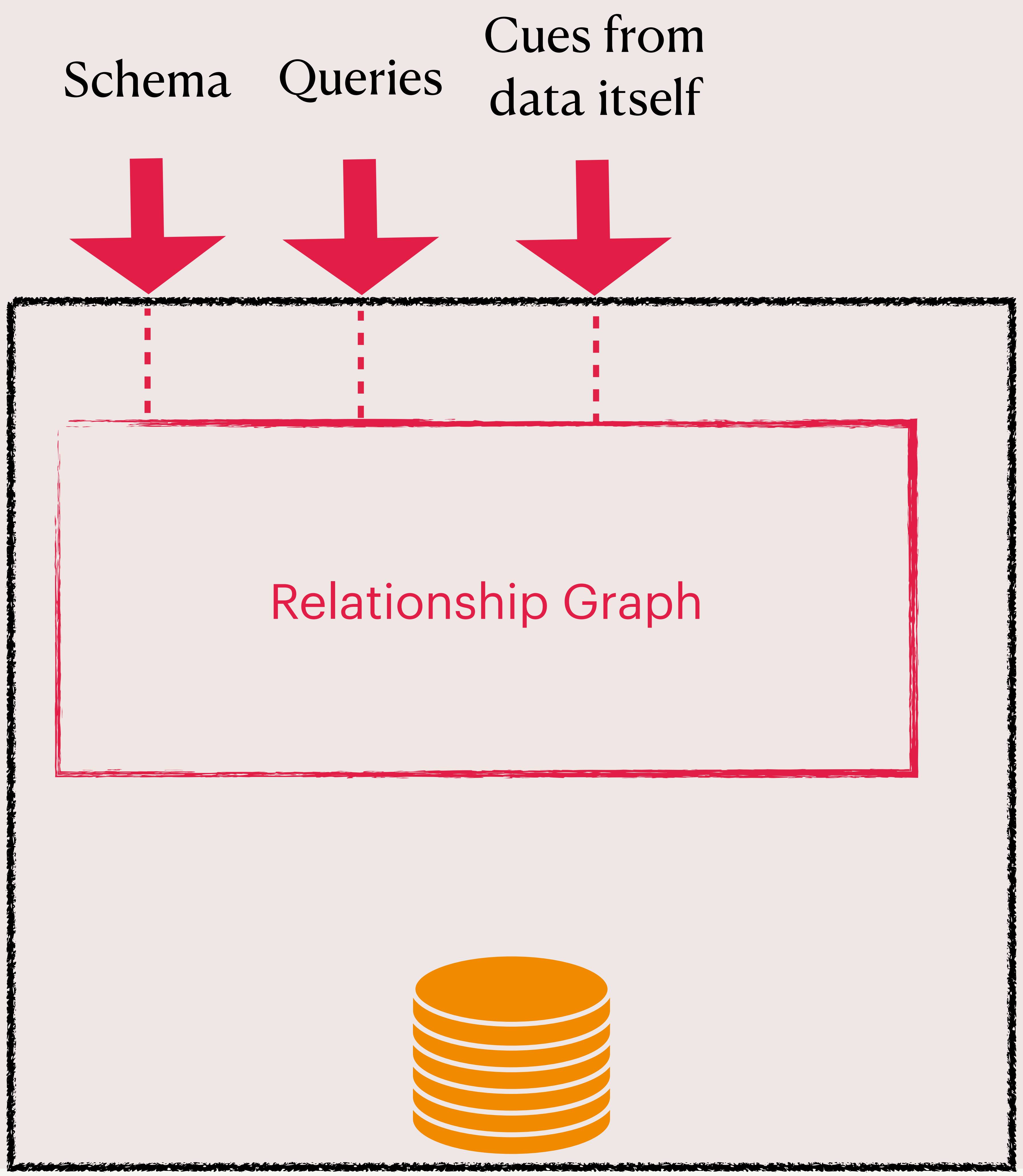
Paper.leadContactId = ContactInfo.contactId

Relationship Graph



Encodes data dependencies across tables

Relationship Graph



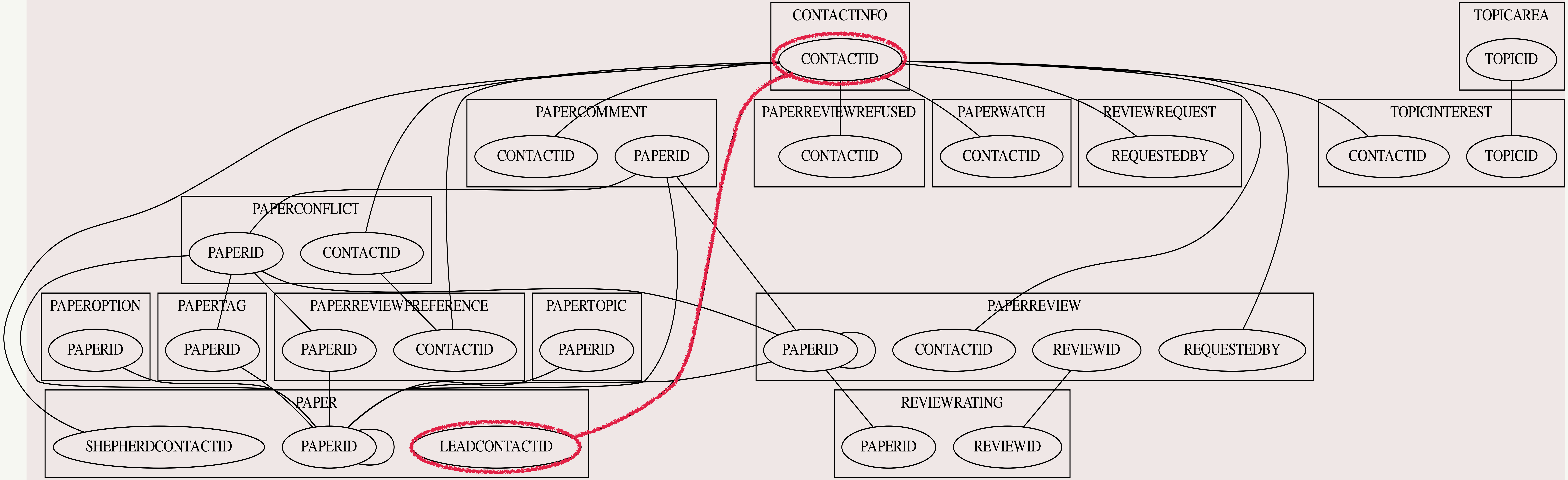
Encodes data dependencies across tables

Rich literature on identifying functional dependencies in data

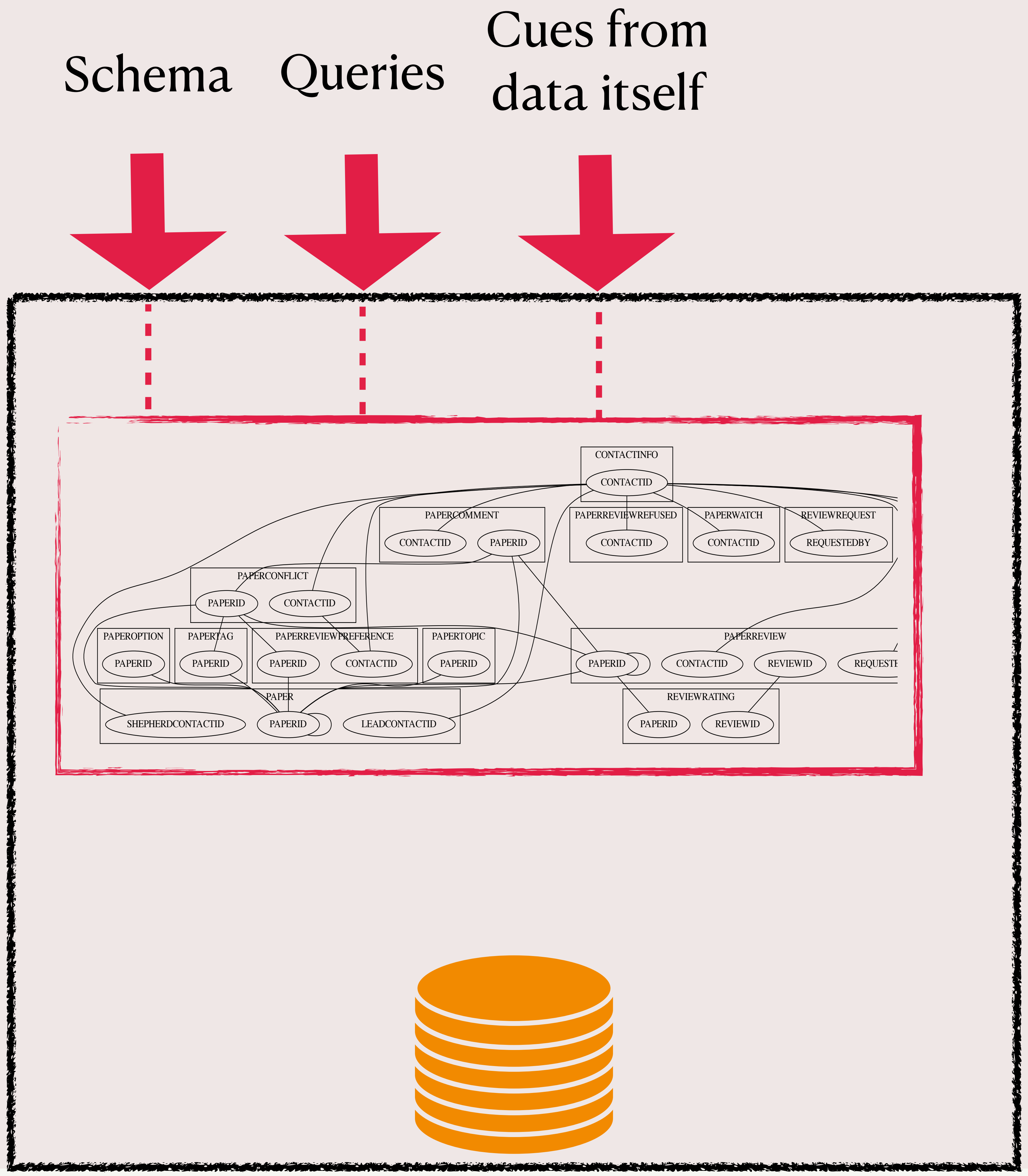
See survey by Abedjan et al., VLDB 2015

Relationship Graph of HotCRP

Using only the joins in queries



Service Data Access Request

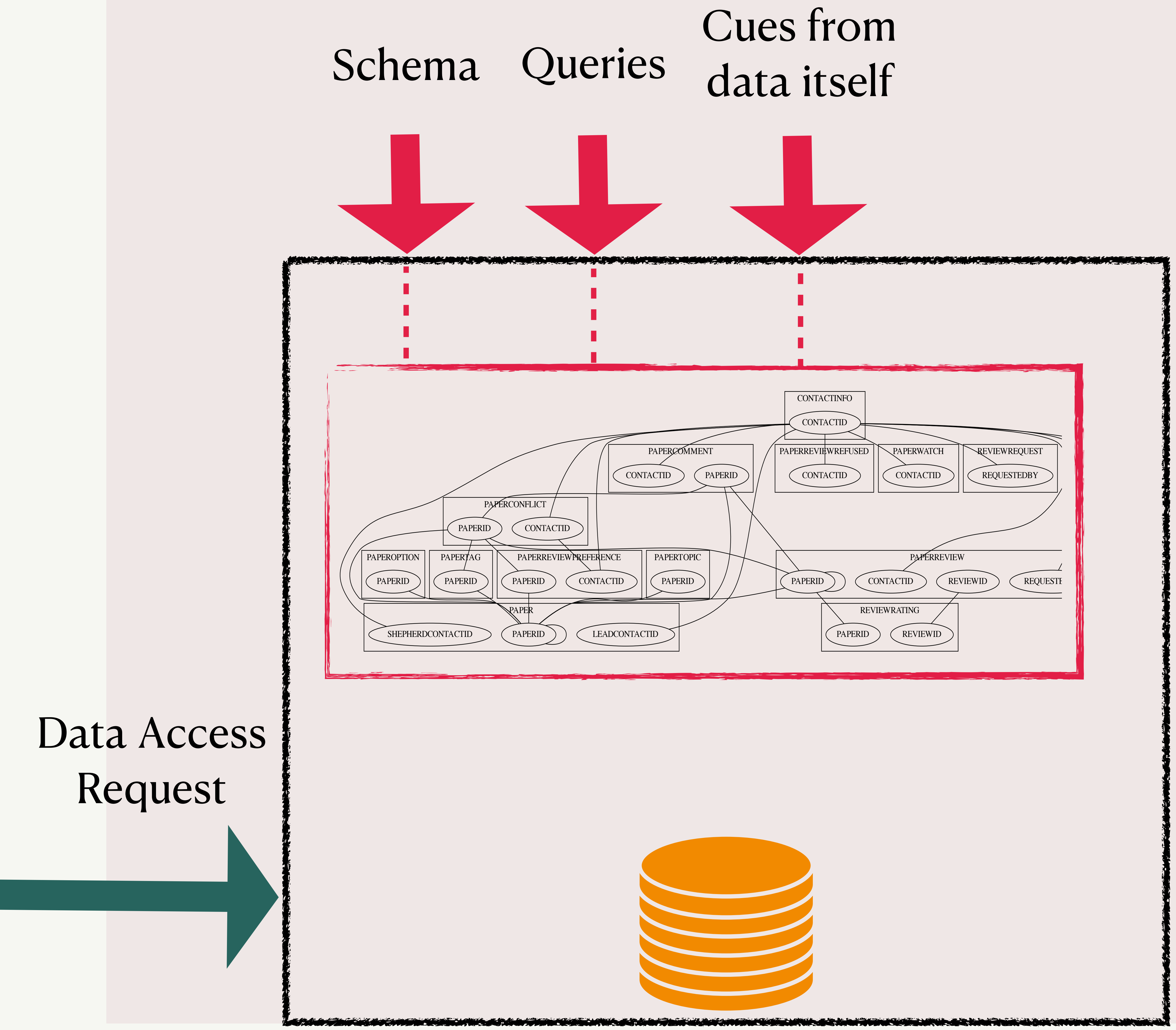


Schema

Queries

Cues from data itself

Service Data Access Request



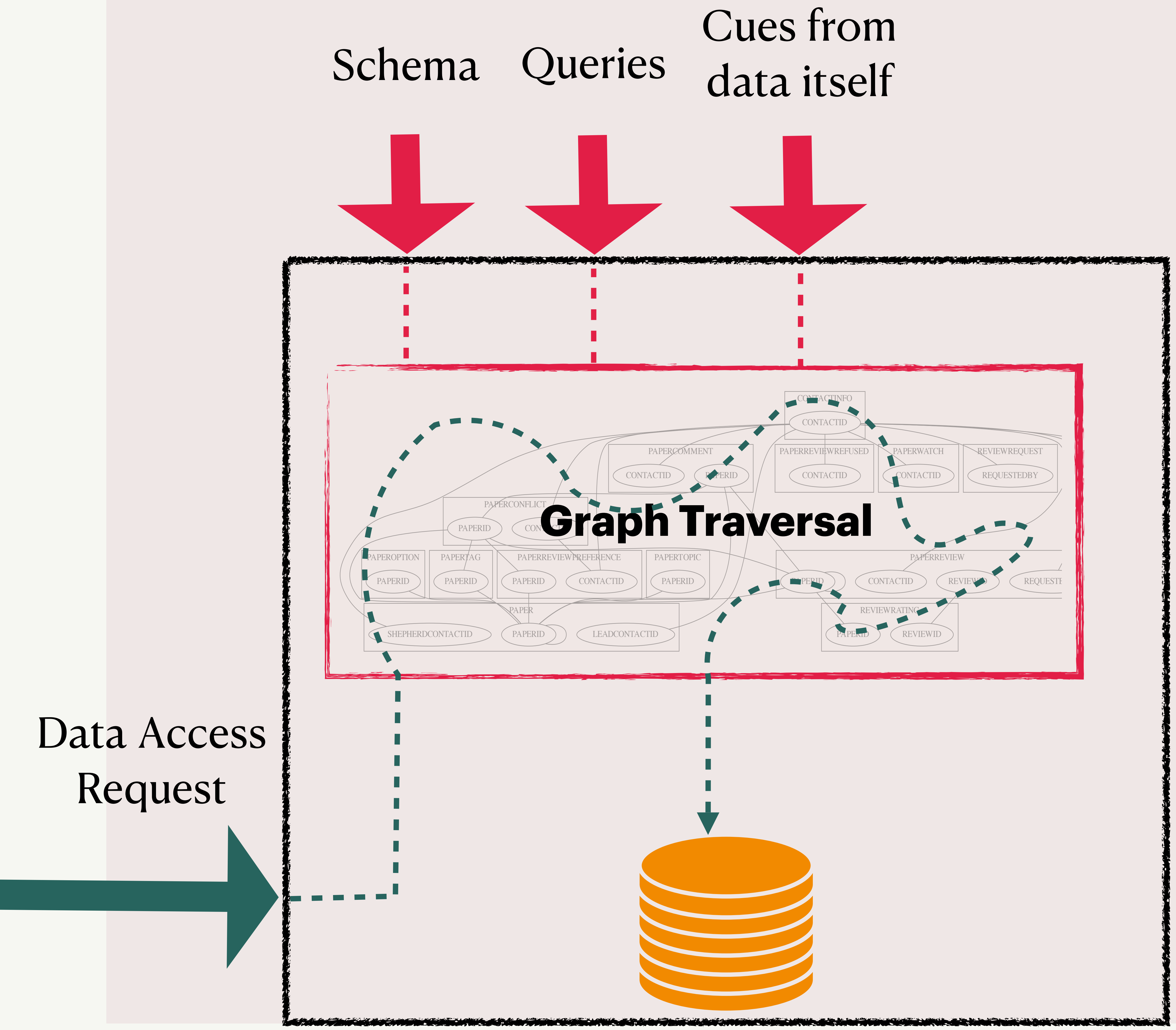
Schema

Queries

Cues from data itself

Data Access Request

Service Data Access Request



Schema

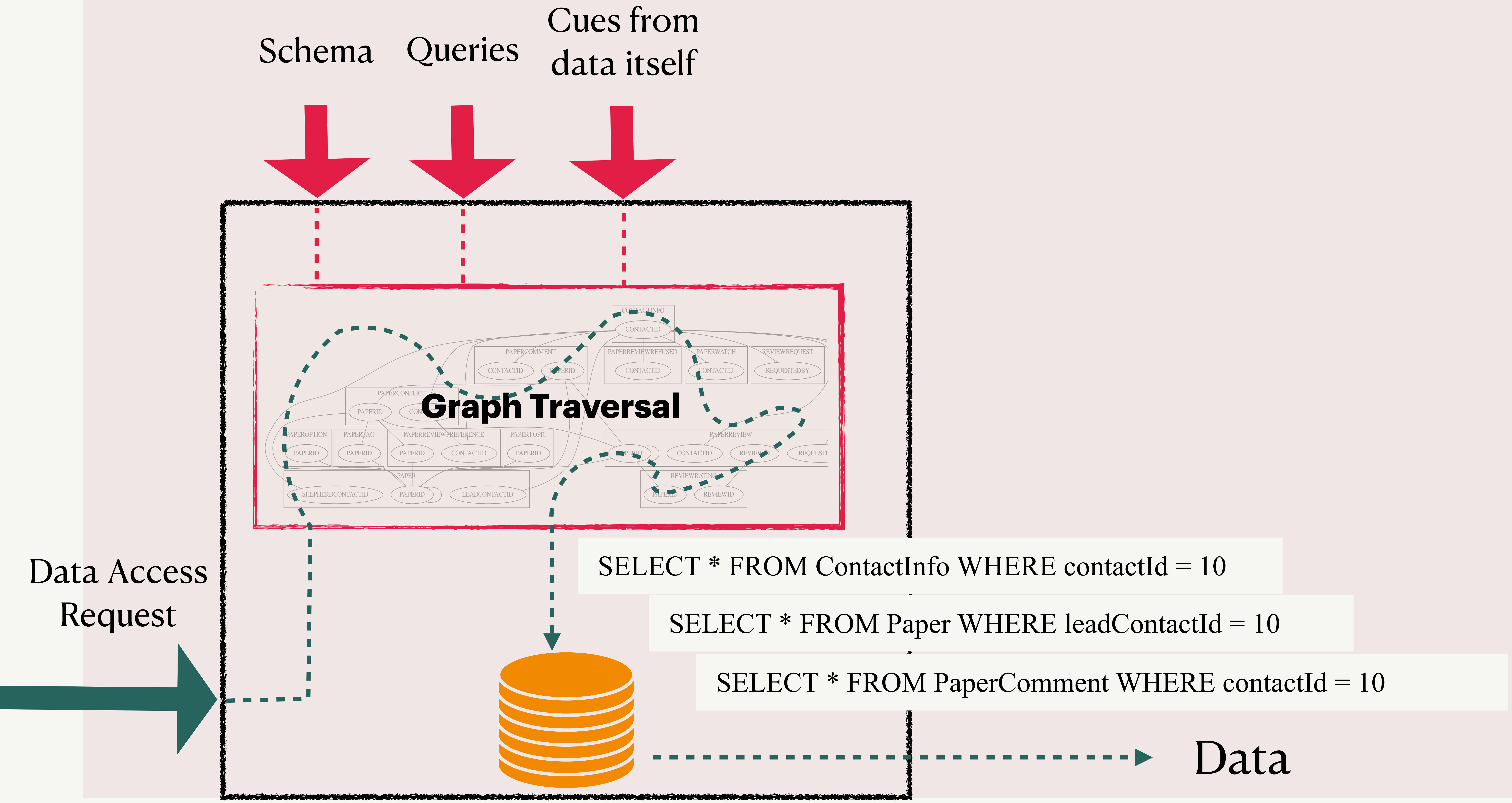
Queries

Cues from
data itself

Graph Traversal

Data Access
Request

Service Data Access Request



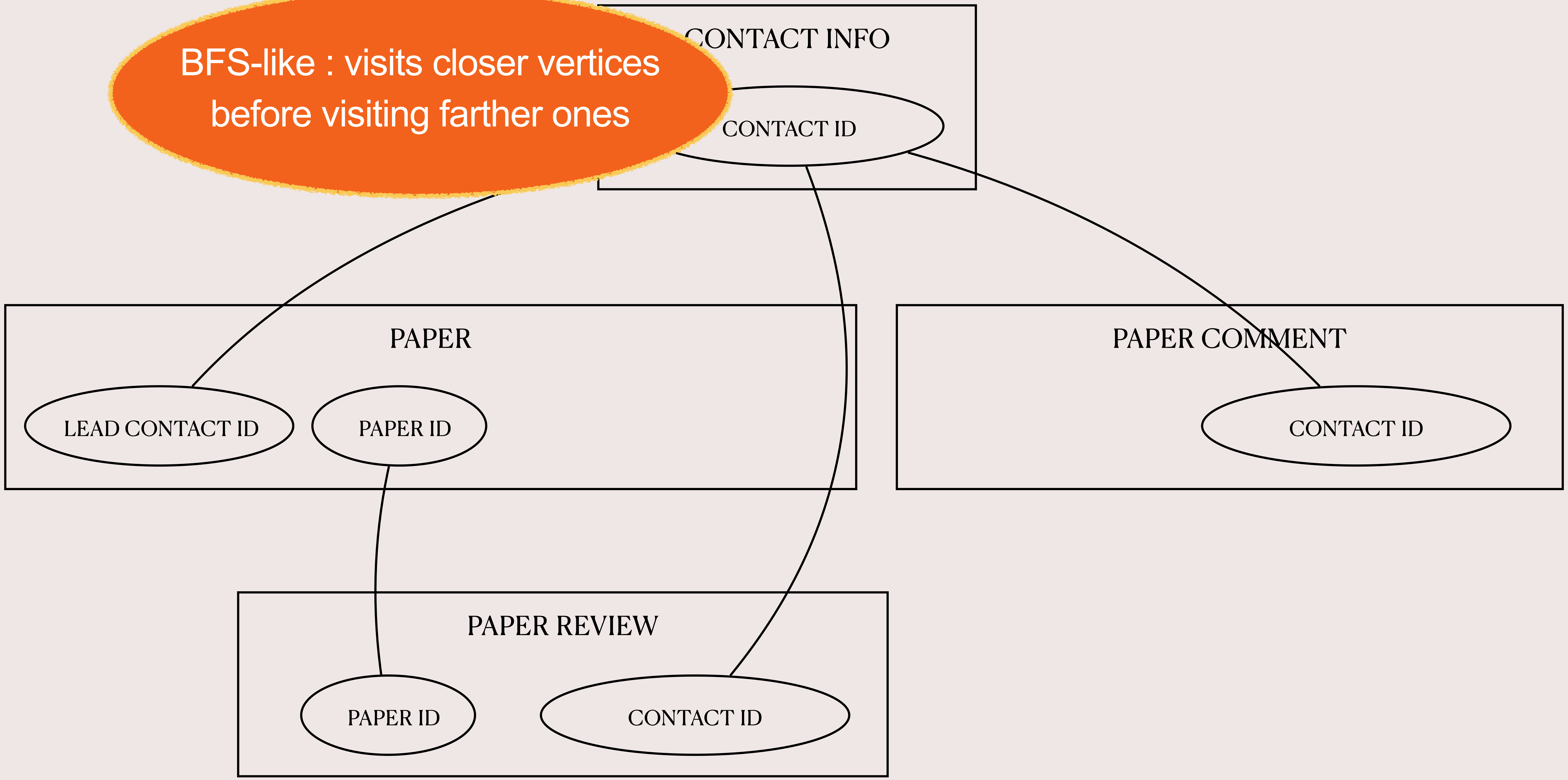
SELECT * FROM ContactInfo WHERE contactId = 10

SELECT * FROM Paper WHERE leadContactId = 10

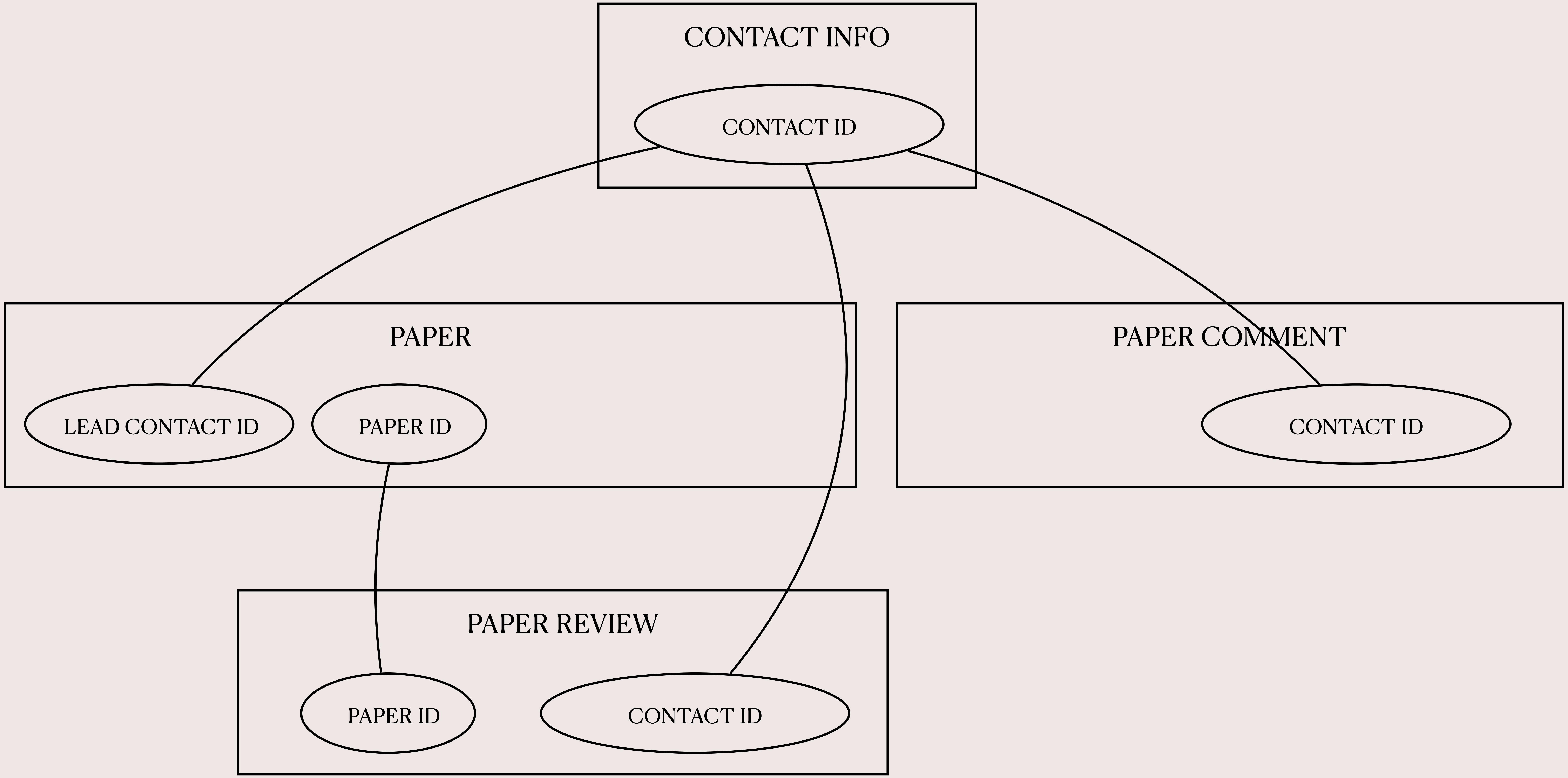
SELECT * FROM PaperComment WHERE contactId = 10

Graph Traversal: Access Request for contactID = 10

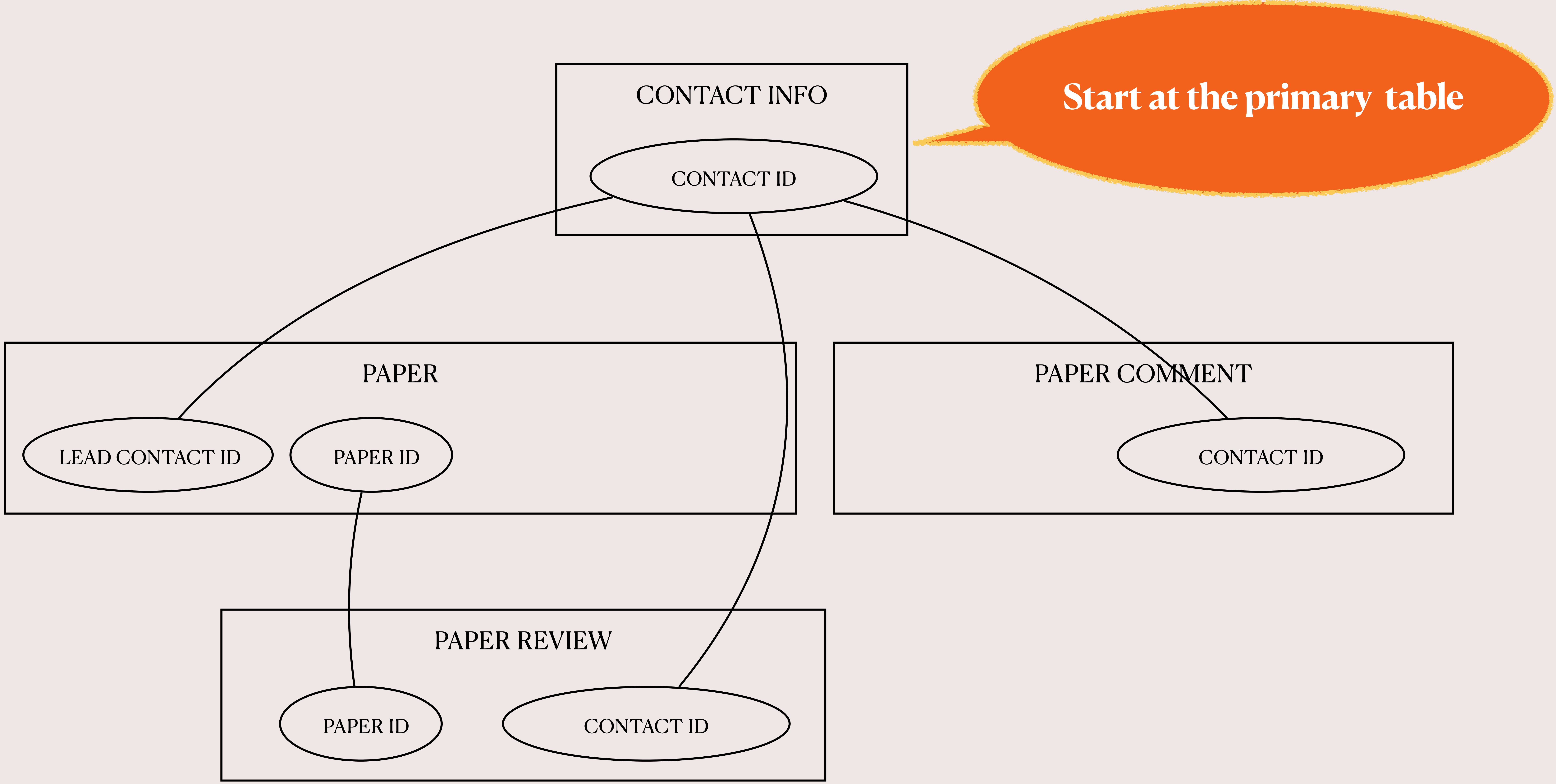
BFS-like : visits closer vertices before visiting farther ones



Graph Traversal: Access Request for contactID = 10



Graph Traversal: Access Request for contactID = 10

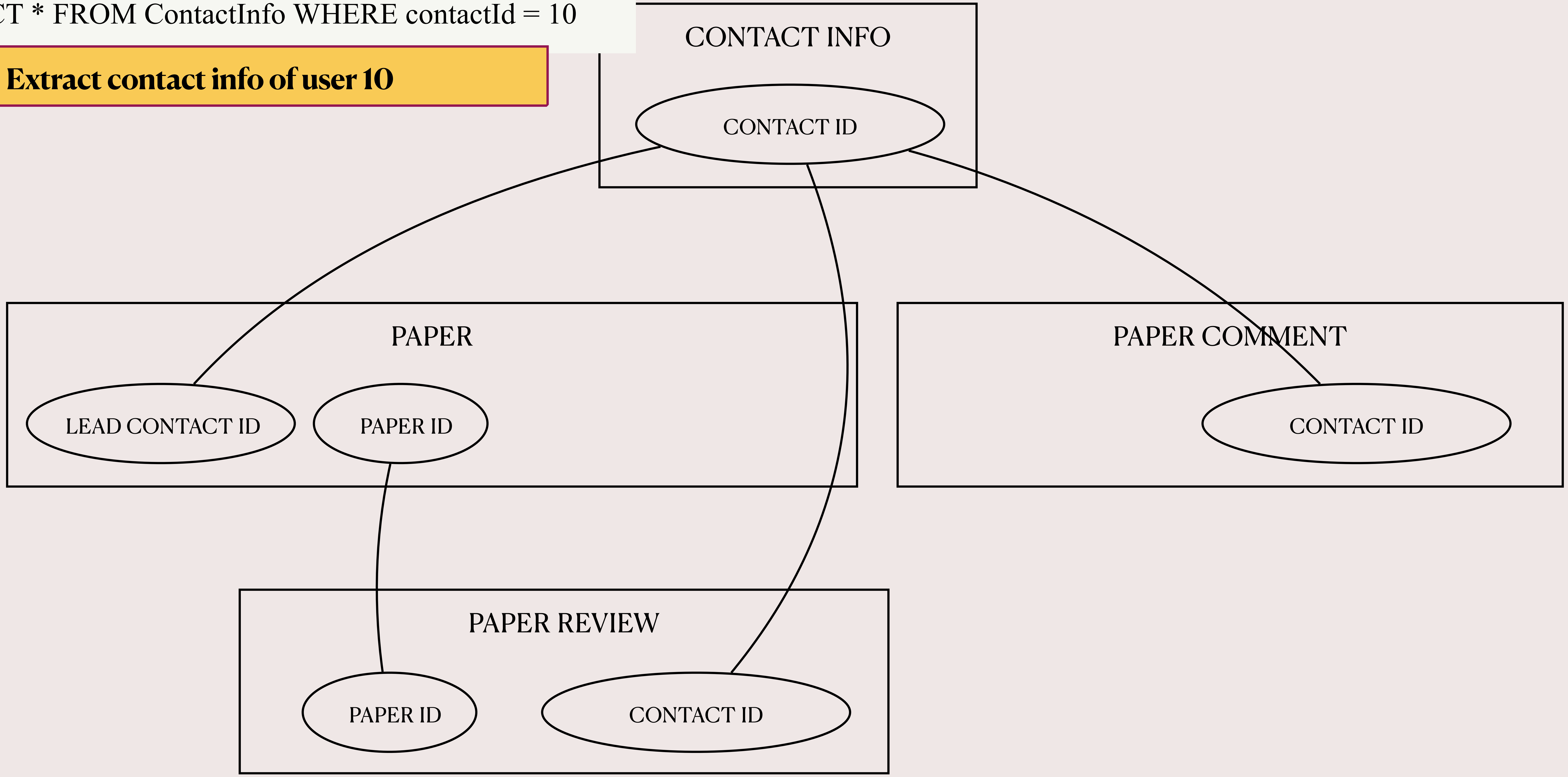


Start at the primary table

Graph Traversal: Access Request for contactID = 10

SELECT * FROM ContactInfo WHERE contactId = 10

Q1: Extract contact info of user 10



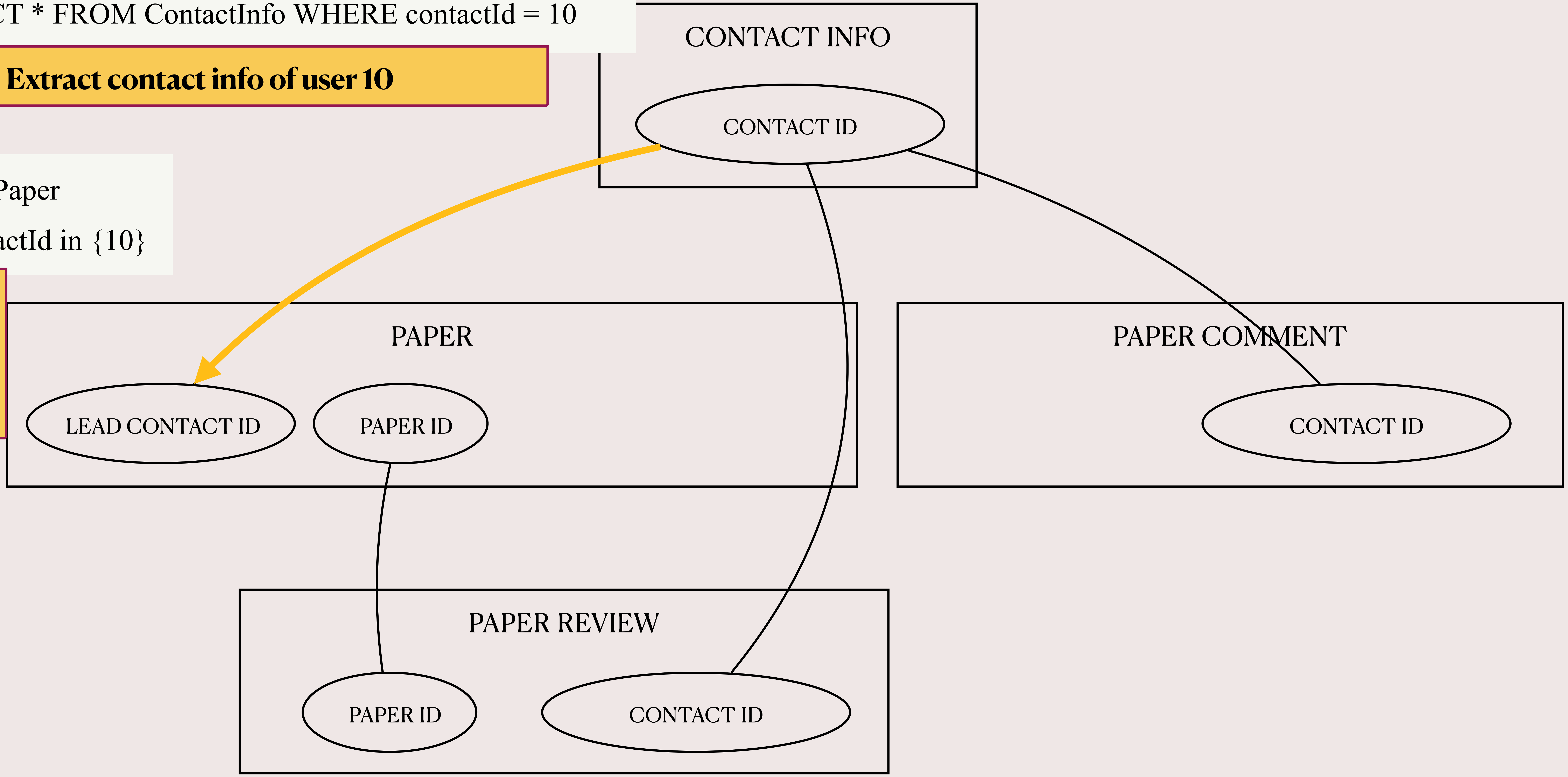
Graph Traversal: Access Request for contactID = 10

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

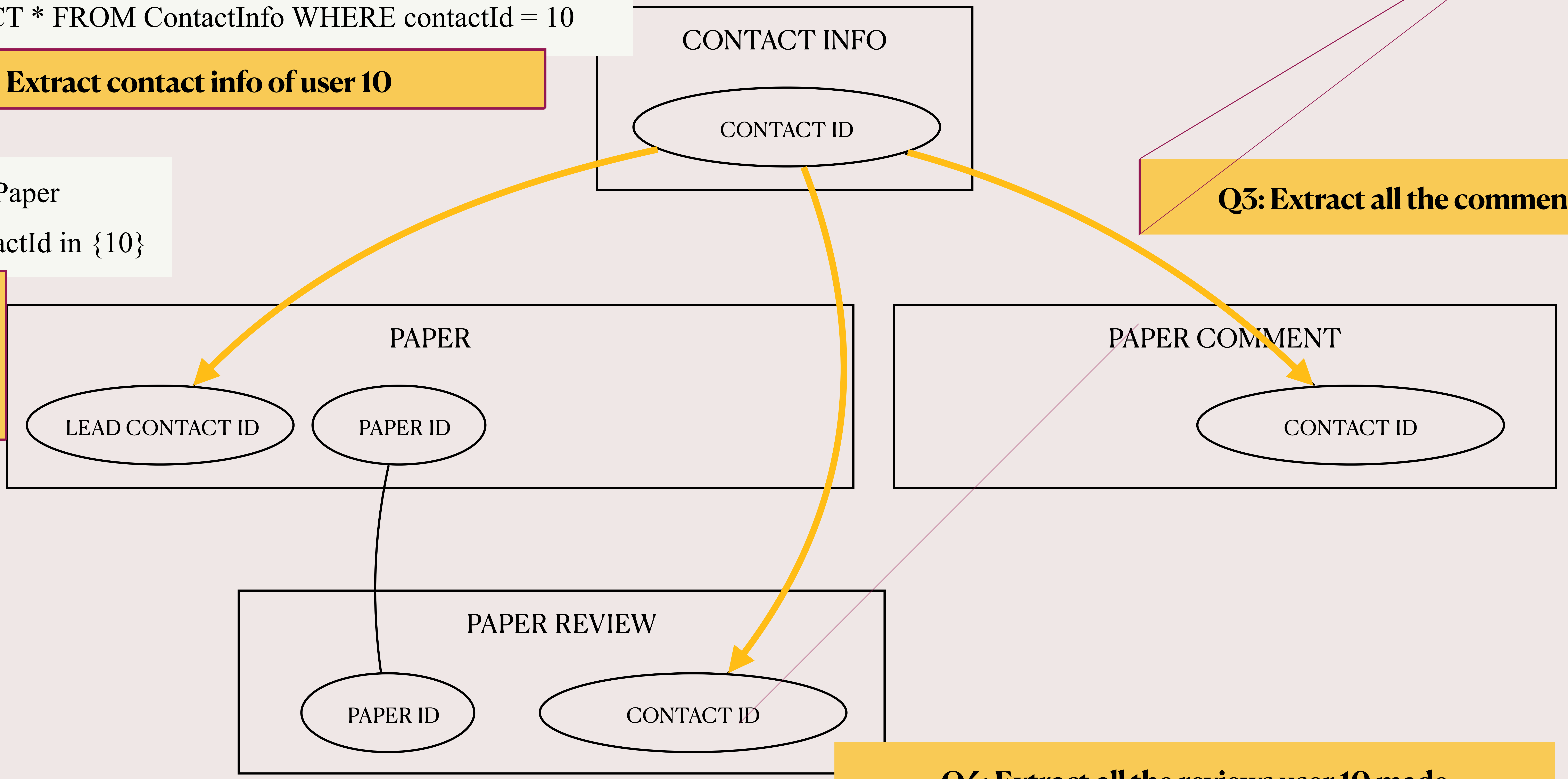
Q1: Extract contact info of user 10

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

Q2: Extract all the papers user 10 wrote



Graph Traversal: Access Request for contactID = 10



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

Q1: Extract contact info of user 10

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

Q2: Extract all the papers user 10 wrote

Q3: Extract all the comments of user 10

Q4: Extract all the reviews user 10 made

CONTACT INFO

CONTACT ID

PAPER

LEAD CONTACT ID

PAPER ID

PAPER COMMENT

CONTACT ID

PAPER REVIEW

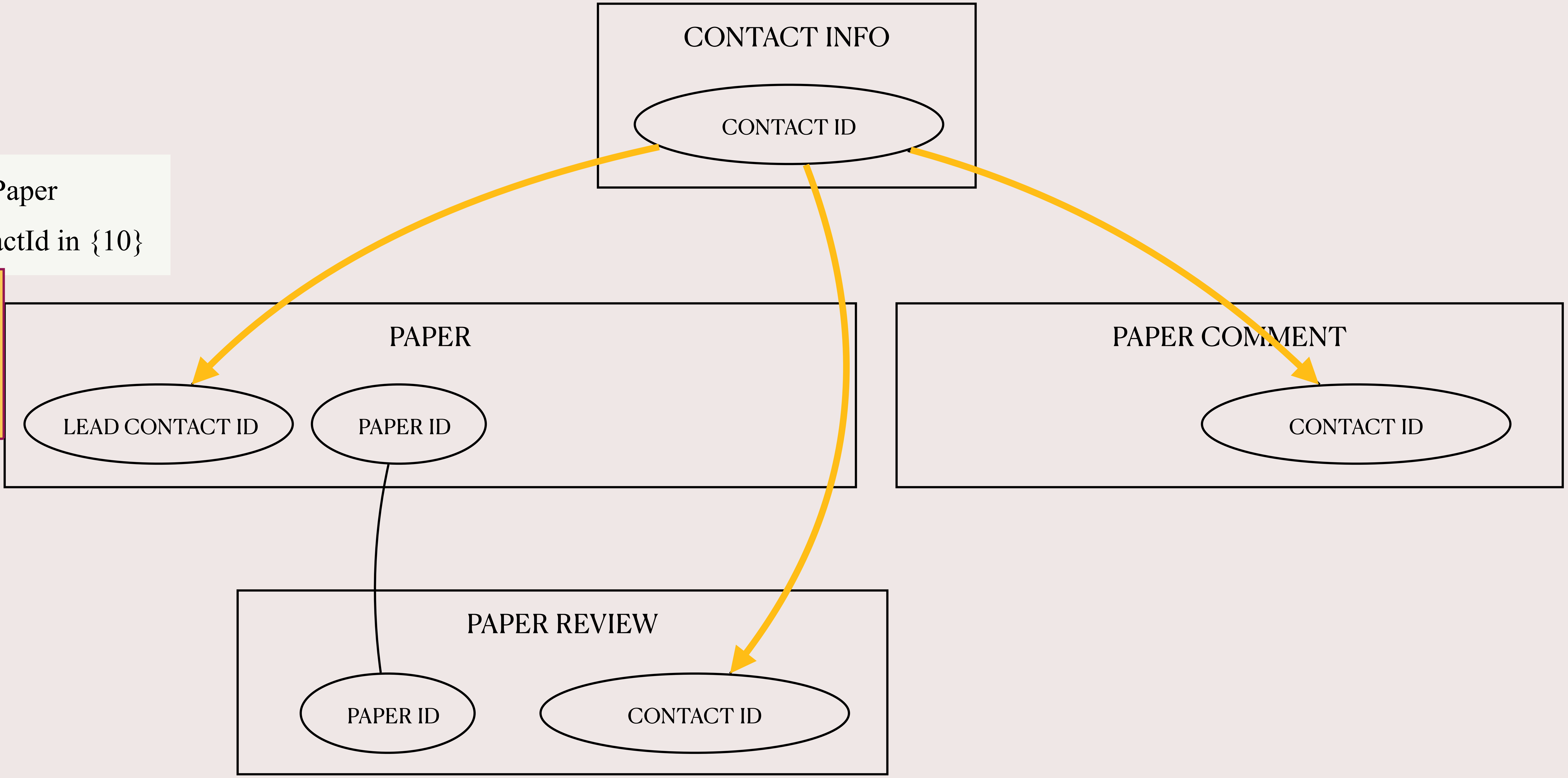
PAPER ID

CONTACT ID

Graph Traversal: Access Request for contactID = 10

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

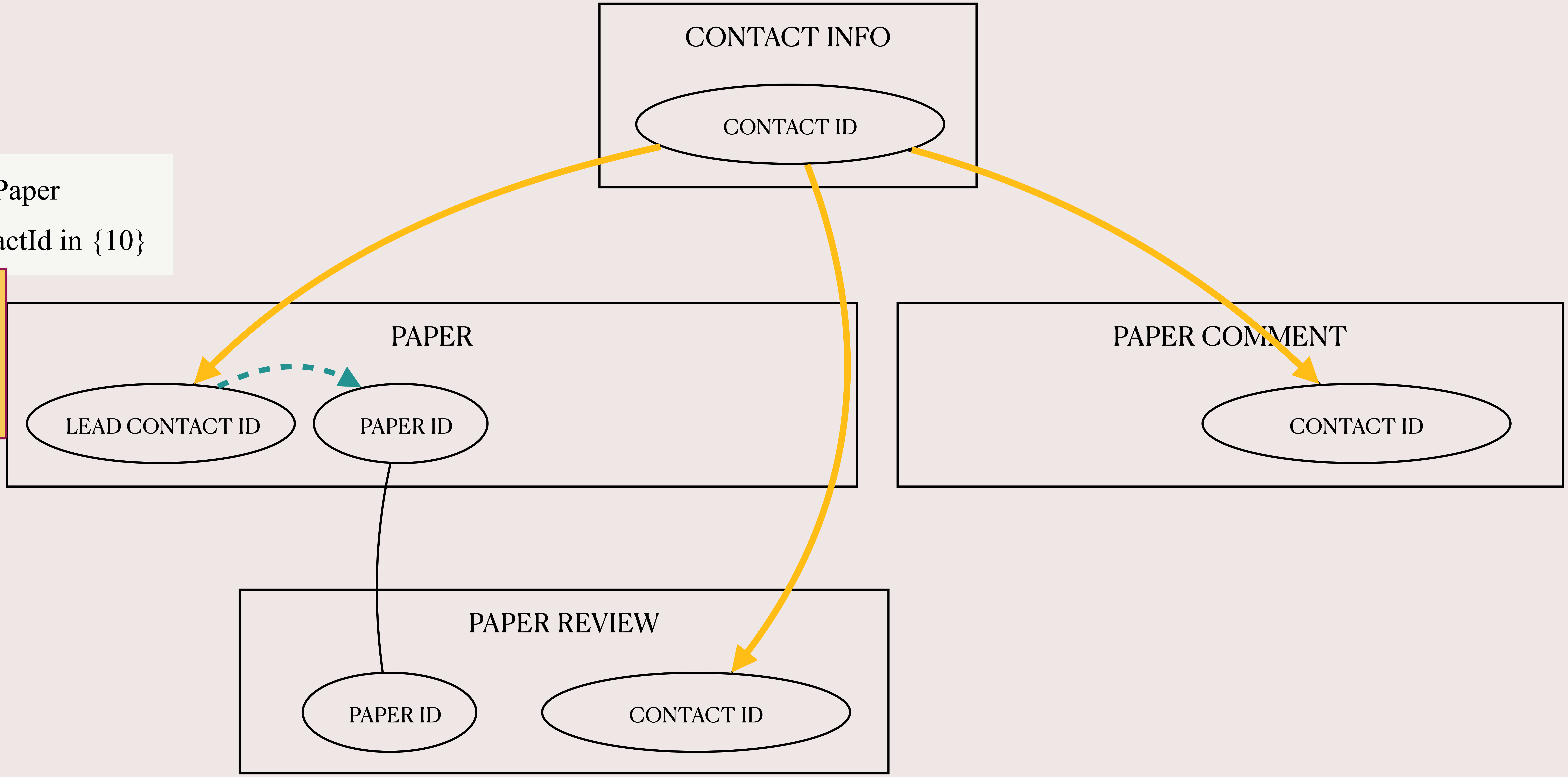
Q2: Extract all the papers user 10 wrote



Graph Traversal: Access Request for contactID = 10

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

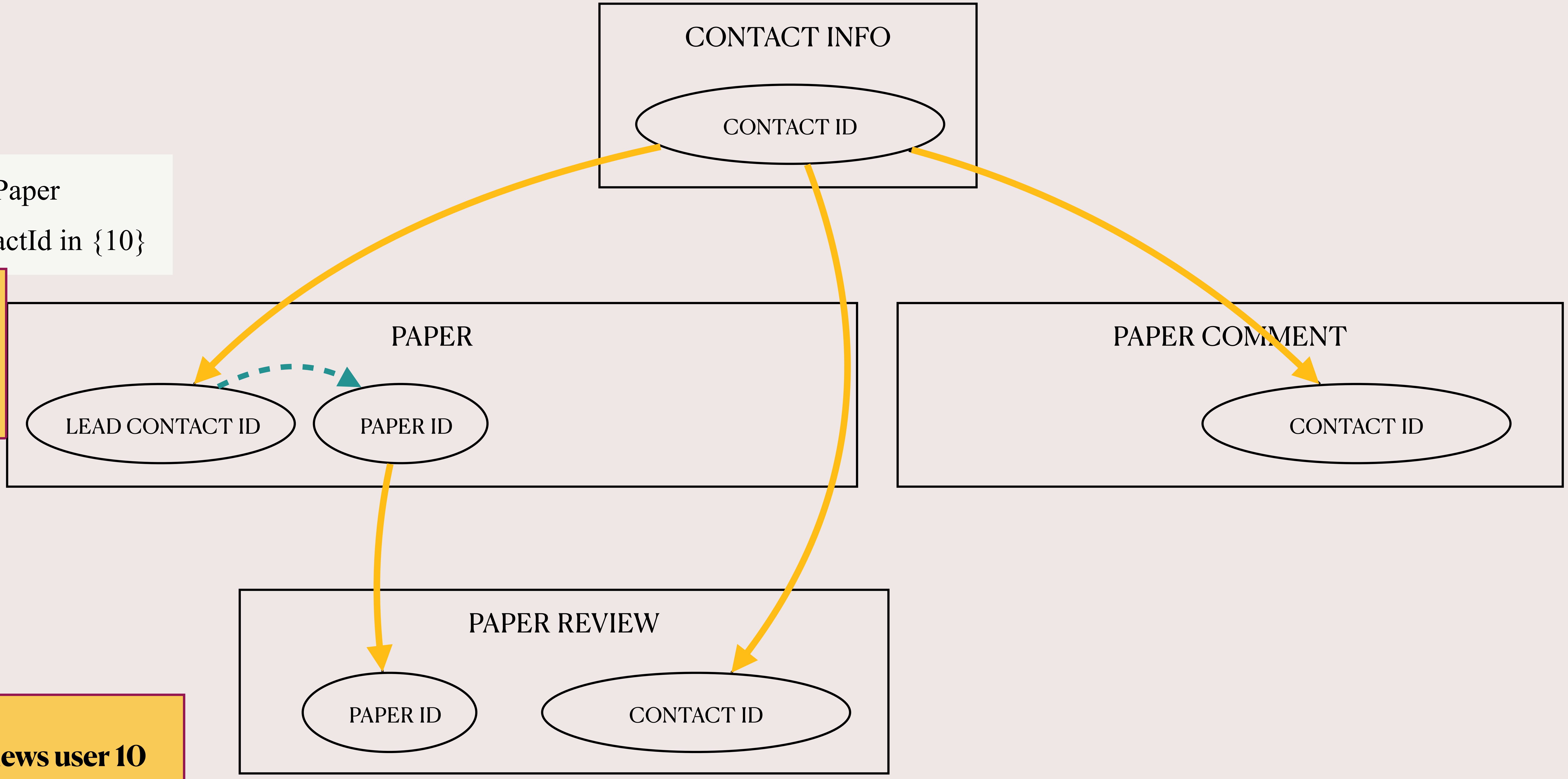
Q2: Extract all the papers user 10 wrote



Graph Traversal: Access Request for contactID = 10

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

Q2: Extract all the papers user 10 wrote



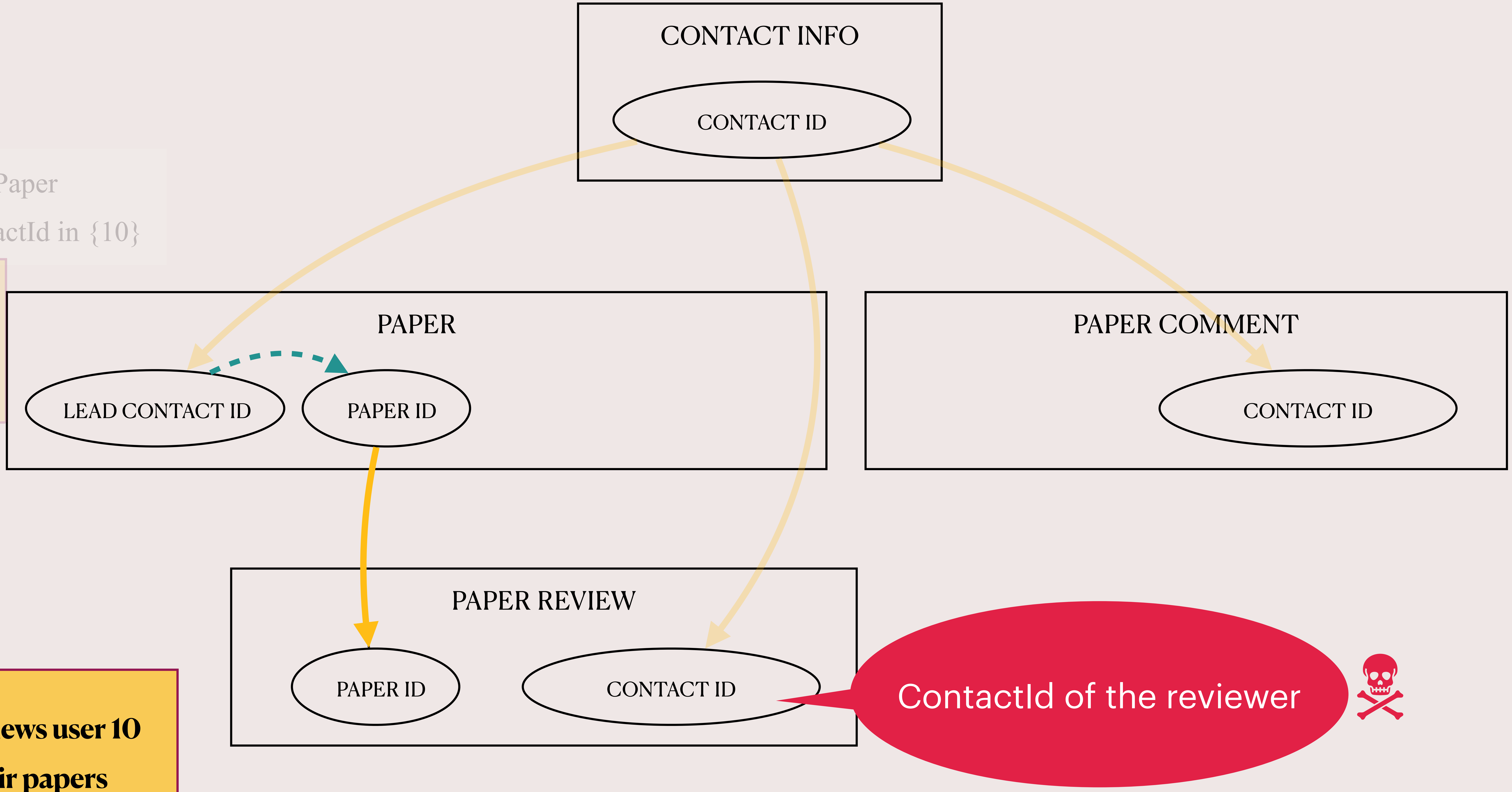
Extract all the reviews user 10 received on their papers

Graph Traversal: Access Request for contactID = 10

SELECT * FROM Paper

WHERE LeadContactId in {10}

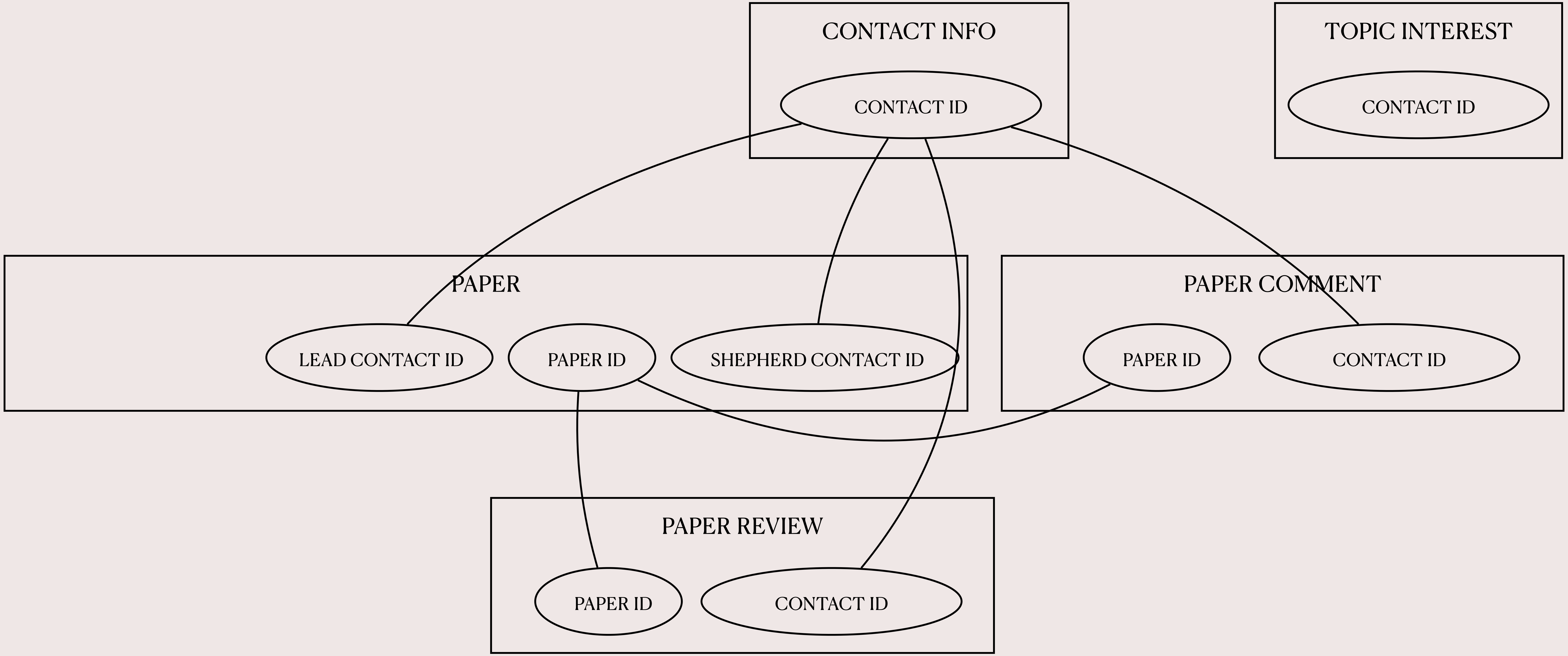
Q2: Extract all the papers user 10 wrote



Extract all the reviews user 10 received on their papers

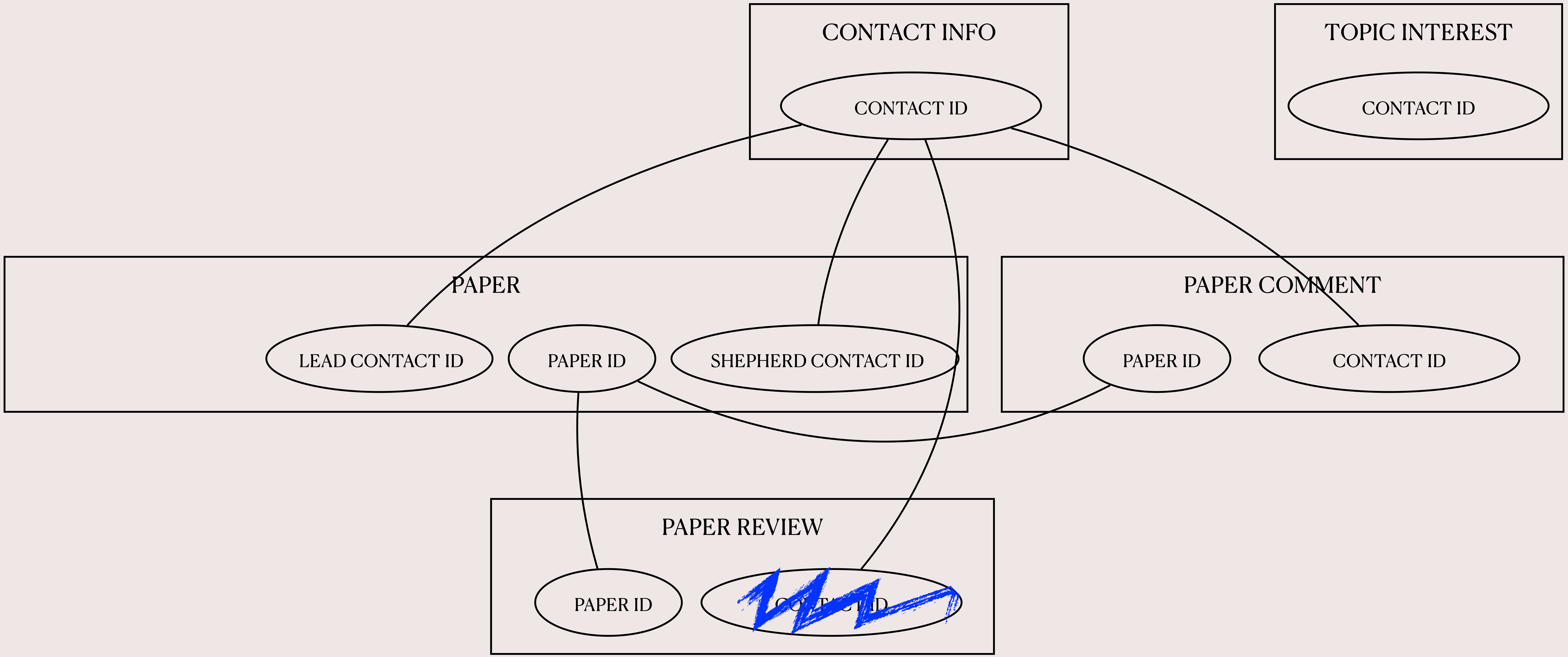
Output 1

Customizations



Customizations

Output 1 Column Filtering



Customizations

Output

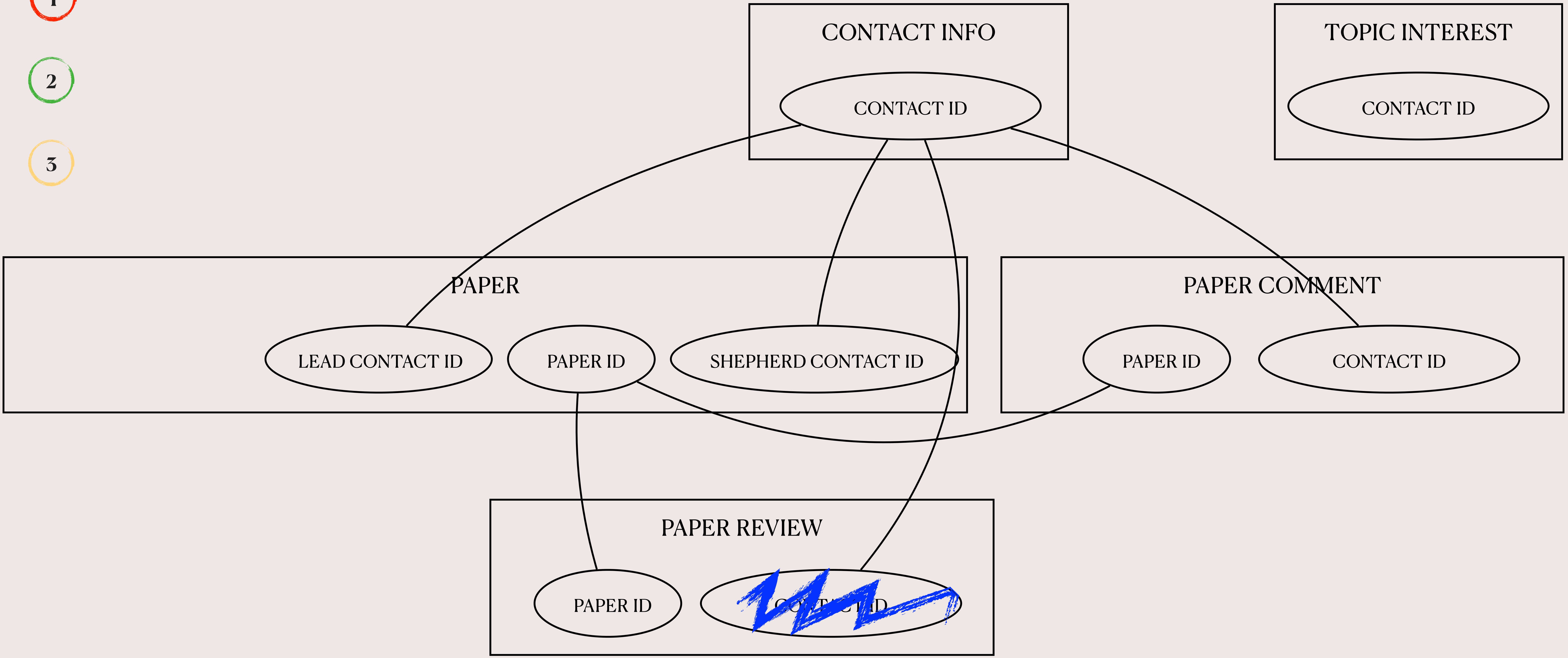
1 Column Filtering

1

Graph

2

3



Customizations

Output

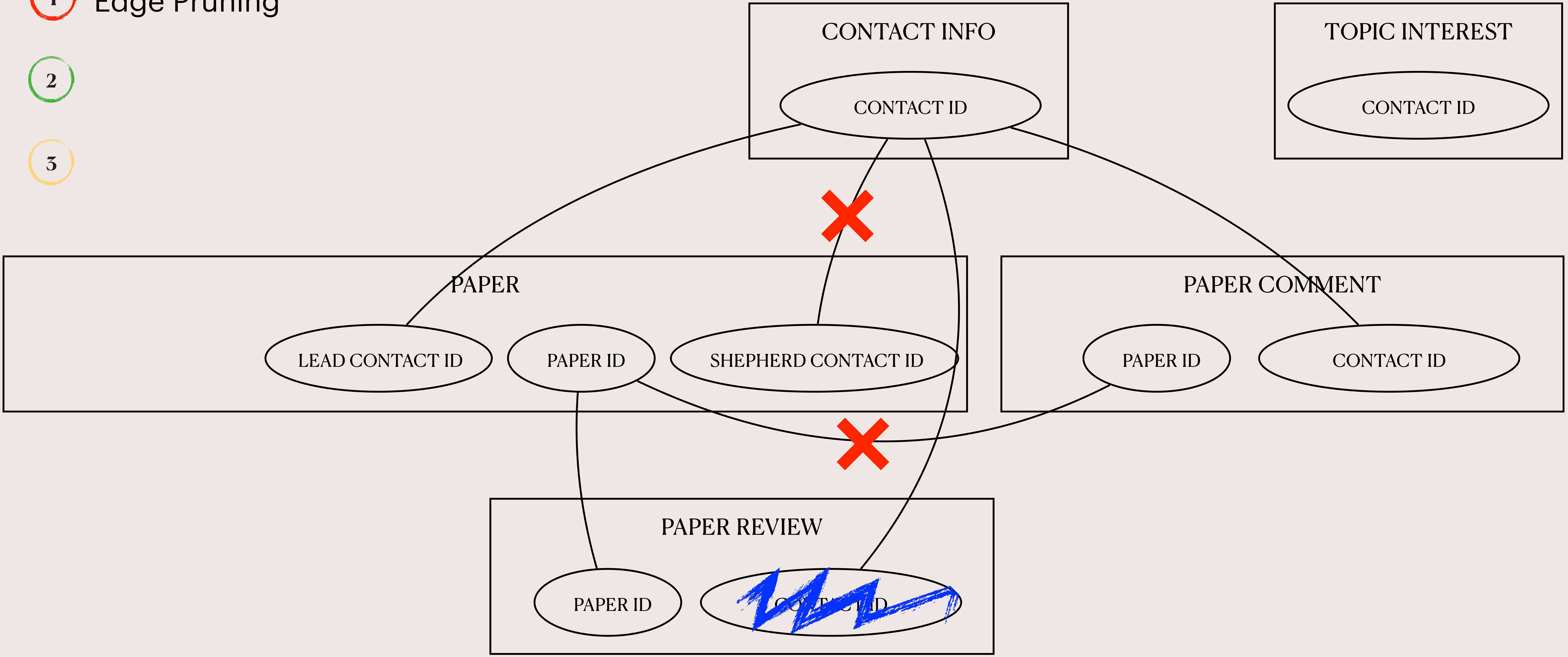
1 Column Filtering

1 Edge Pruning

Graph

2

3



Customizations

Output

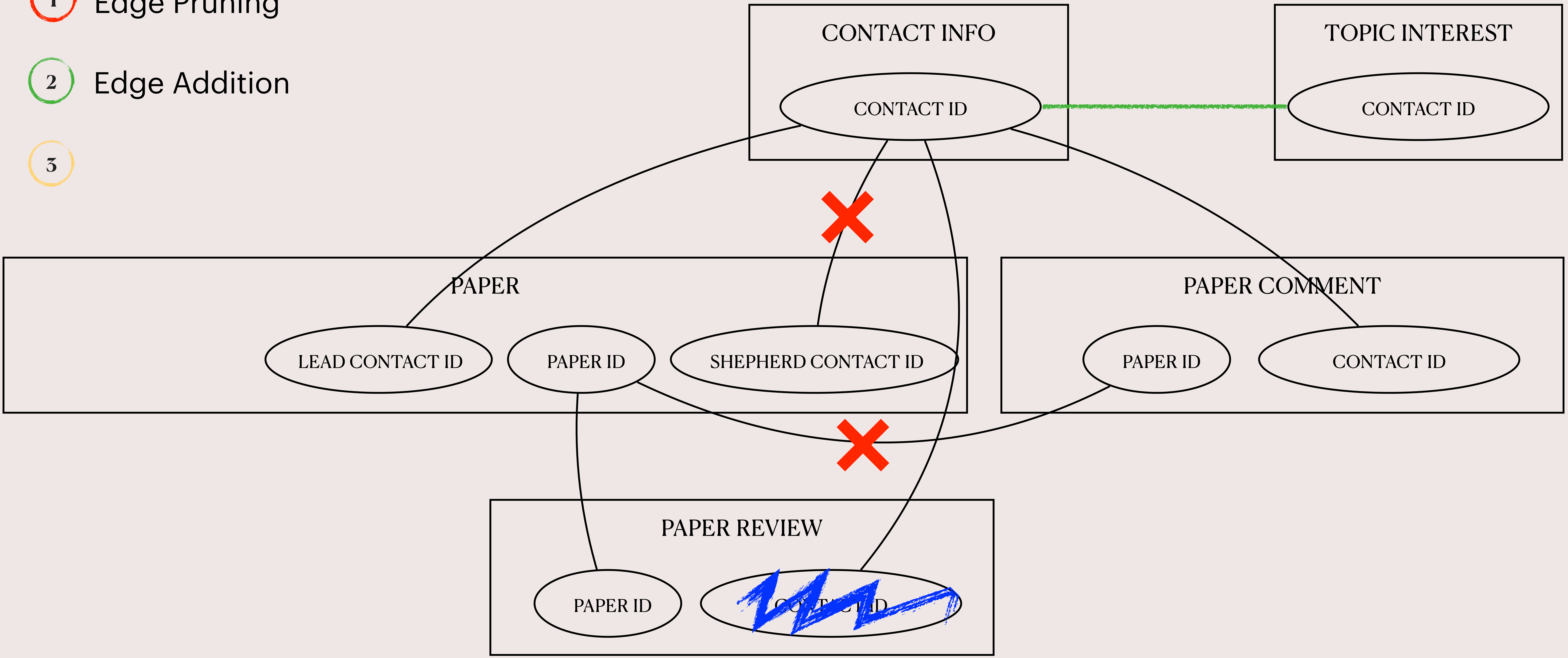
1 Column Filtering

Graph

1 Edge Pruning

2 Edge Addition

3



Customizations

Output

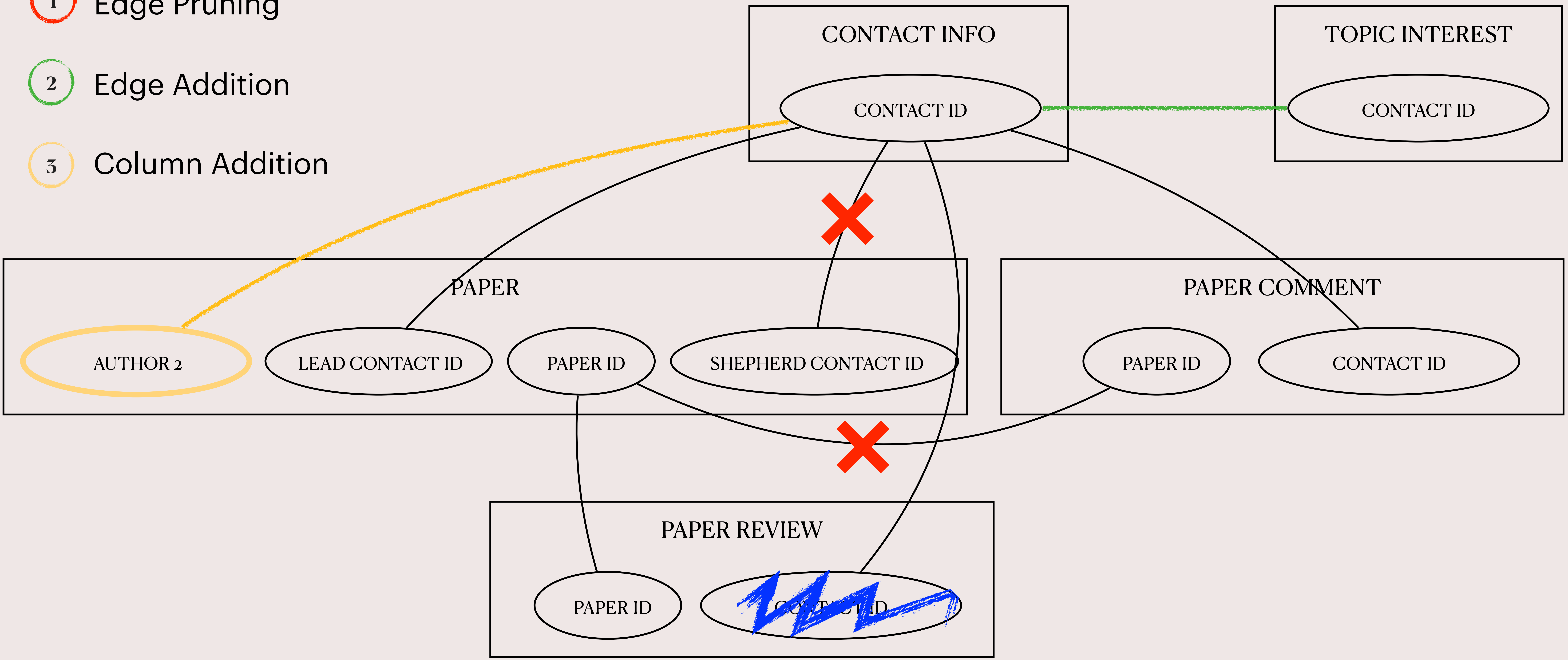
1 Column Filtering

Graph

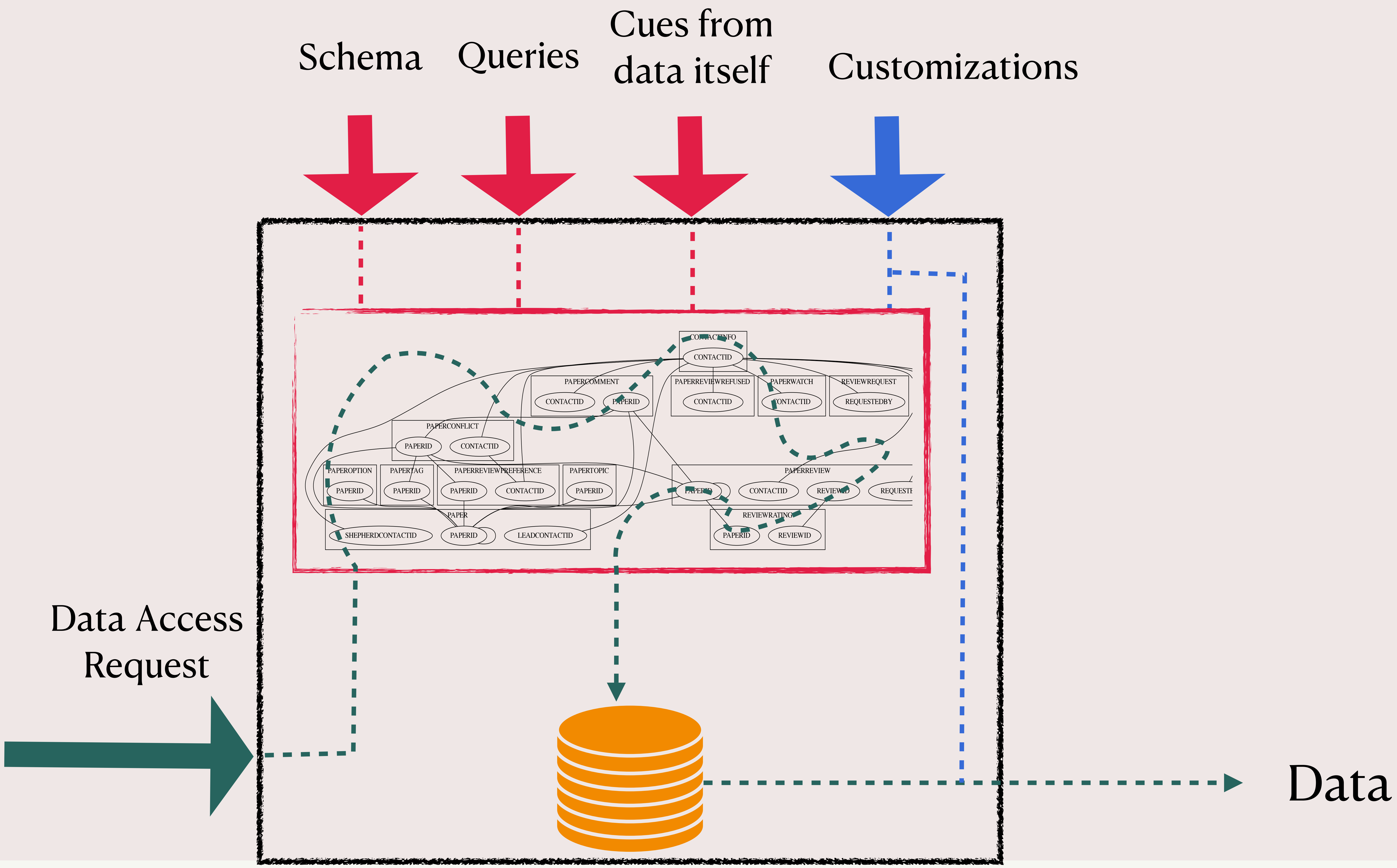
1 Edge Pruning

2 Edge Addition

3 Column Addition



GDPRizer: Architecture



Schema

Queries

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

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 ?

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

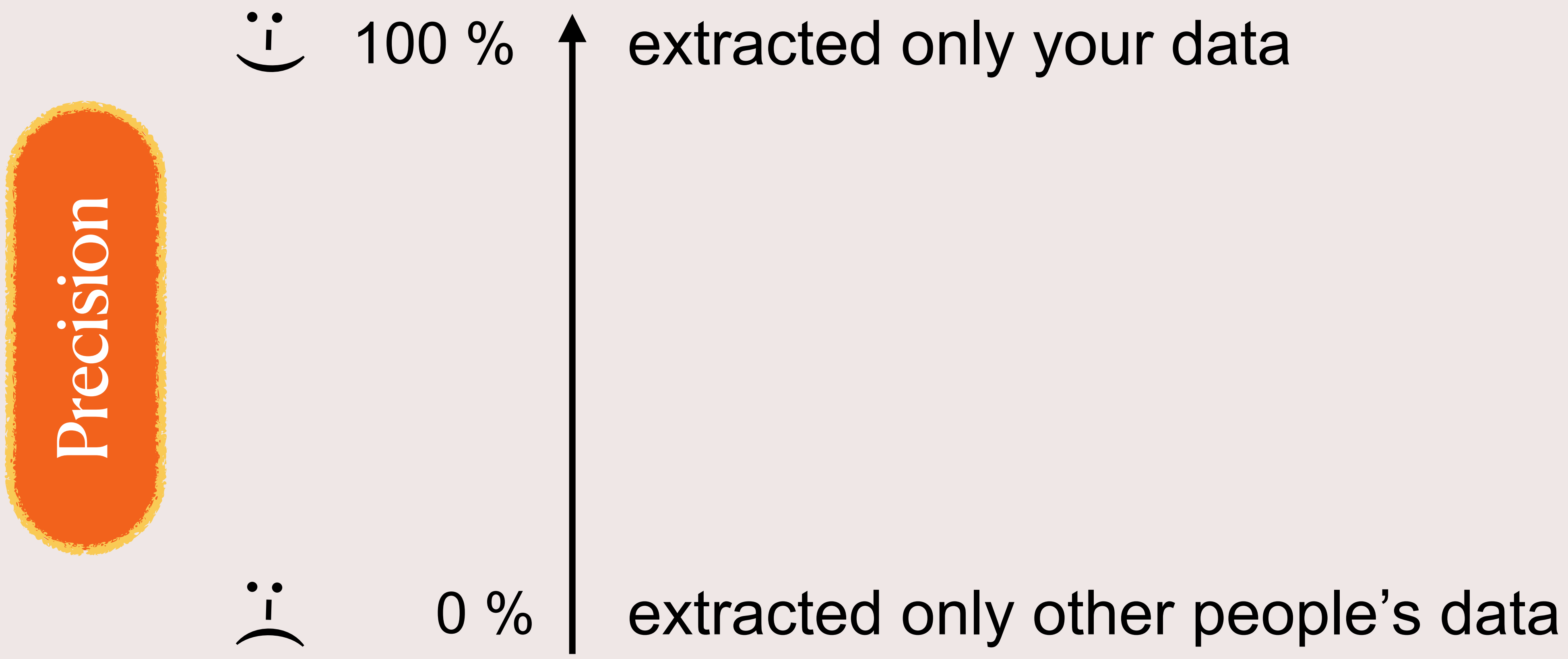
Q1: Does GDPRizer correctly identify user-data ?

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**:



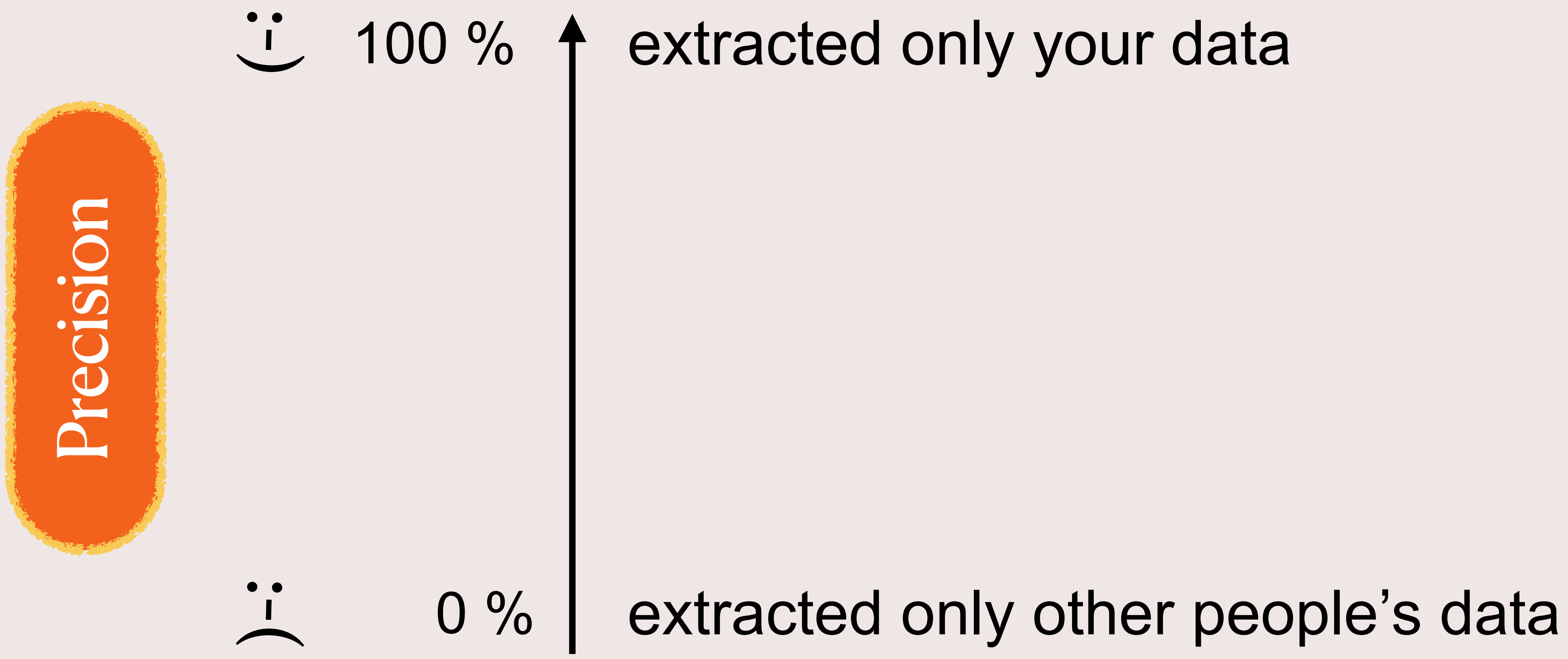
Precision

100 % extracted only your data

0 % extracted only other people's data

Q1: Does GDPRizer correctly identify user-data ?

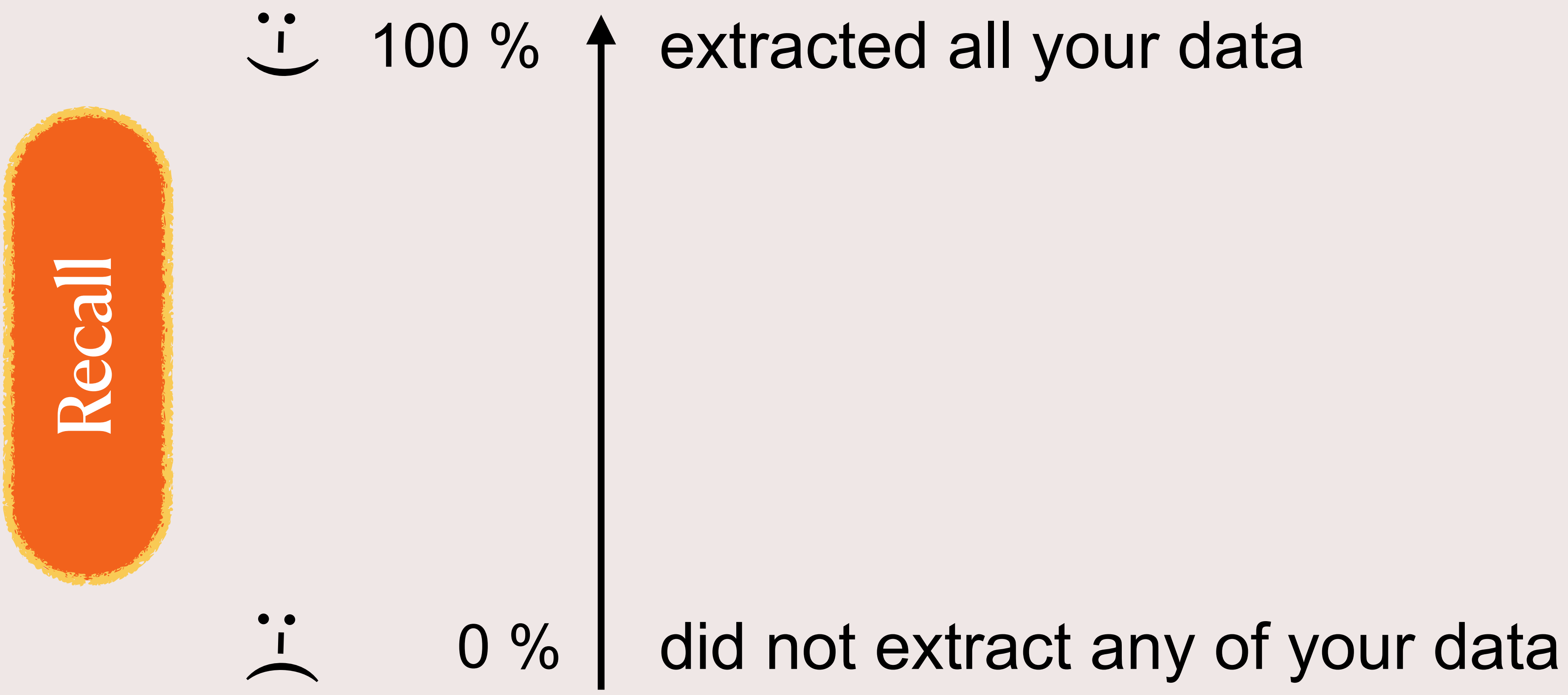
- **Precision**: Measures what fraction of what GDPRizer extracted was actually user-data
- **Recall**: Measures what fraction of the user-data did GDPRizer manage to extract
- **F1-Score**:



Precision

☺ 100 % extracted only your data

☹ 0 % extracted only other people’s data



Recall

☺ 100 % extracted all your data

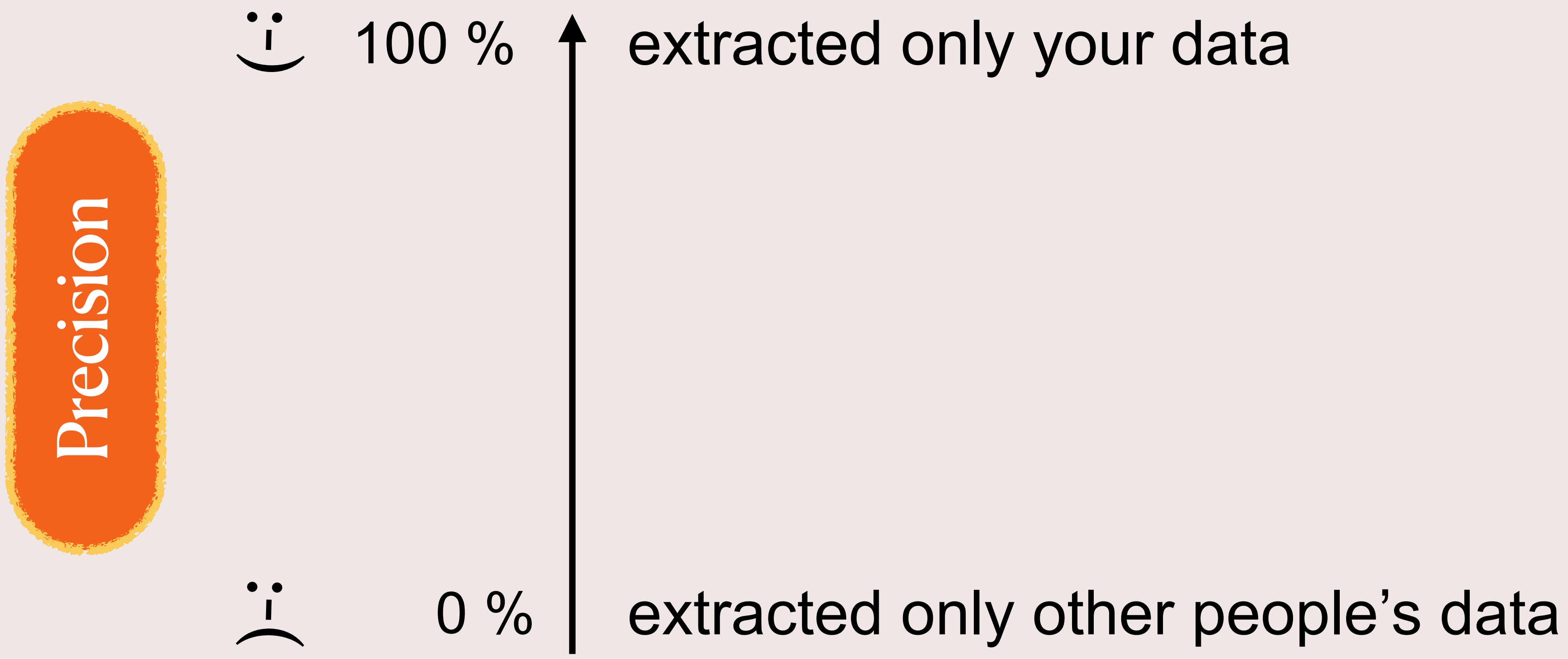
☹ 0 % did not extract any of your data

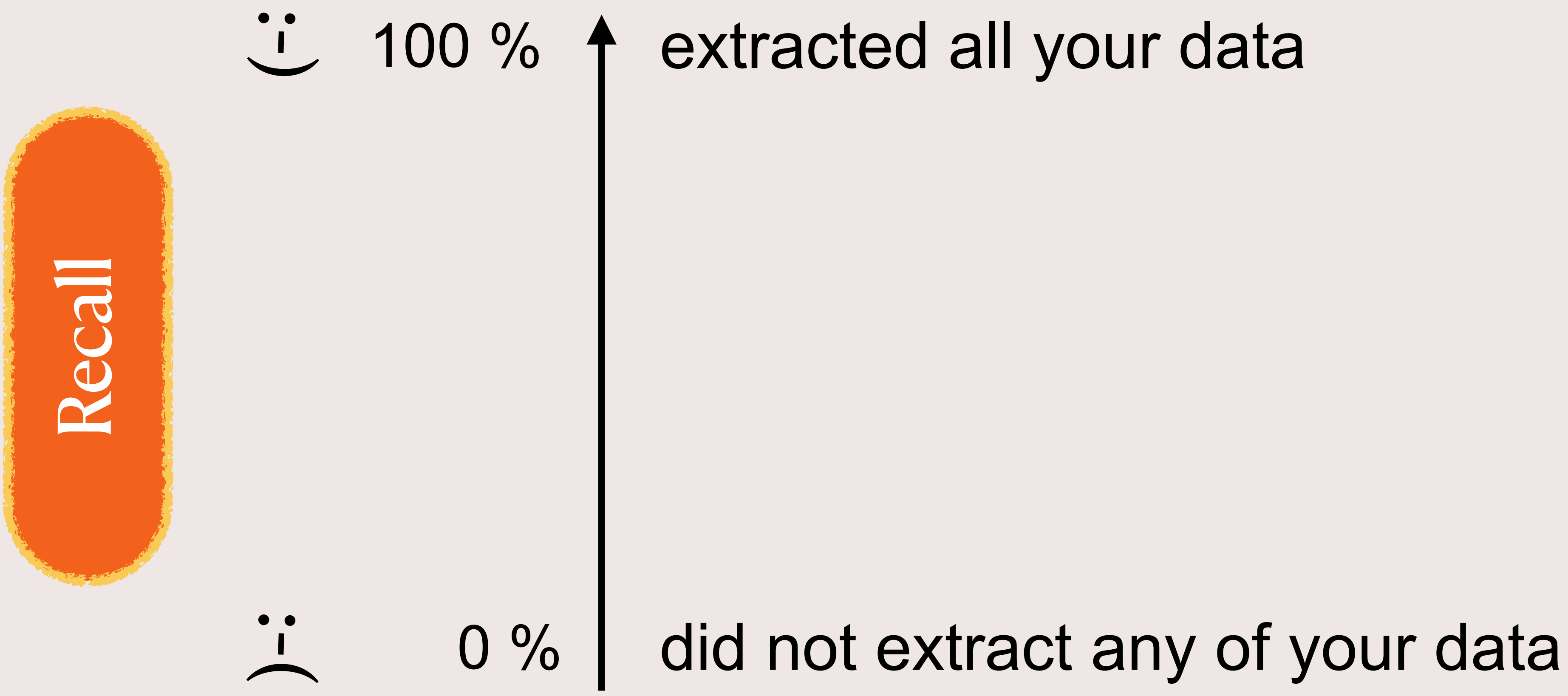
Q1: Does GDPRizer correctly identify user-data ?

- **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

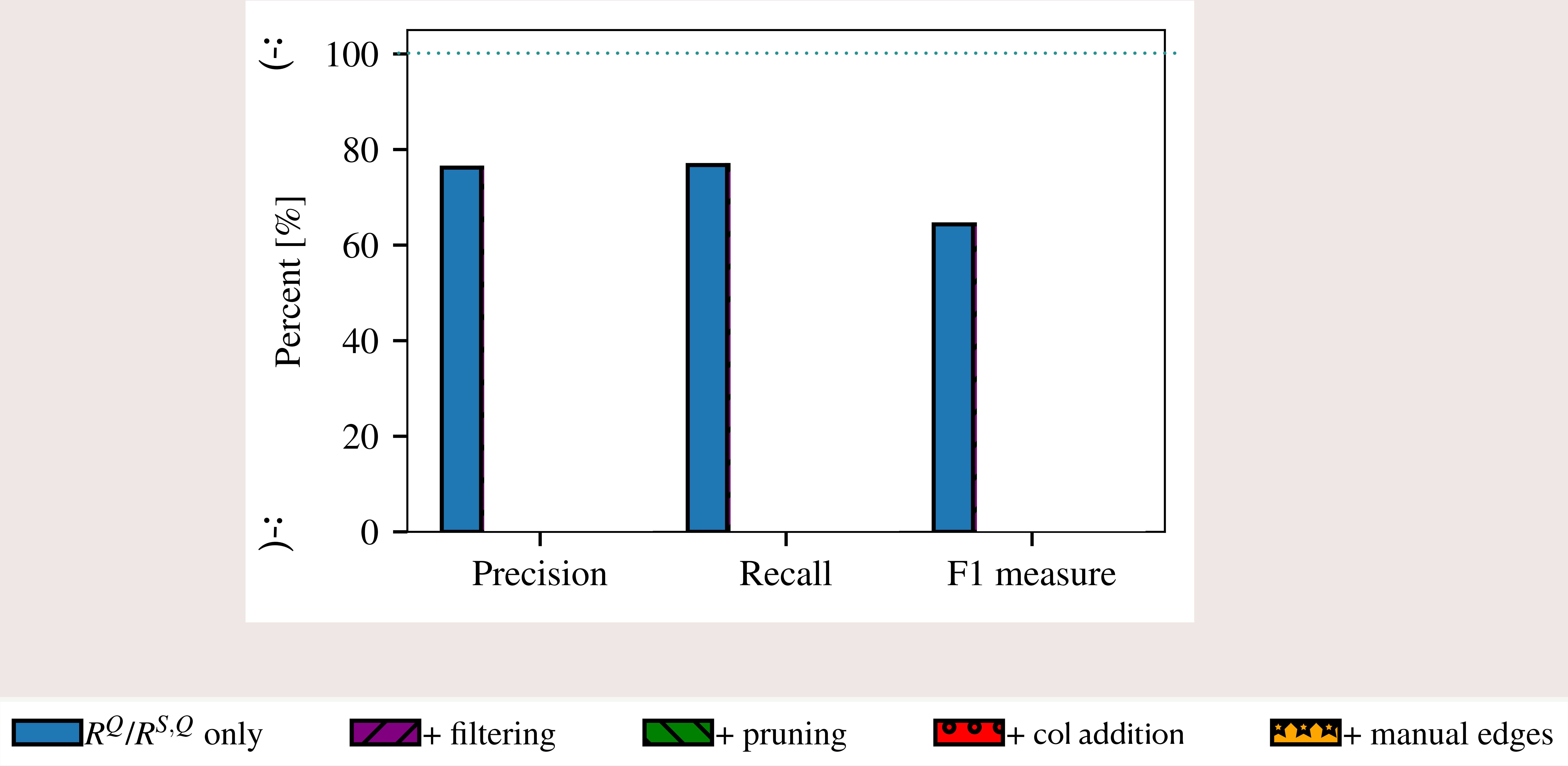
- **F1-Score**: Combination of precision and recall





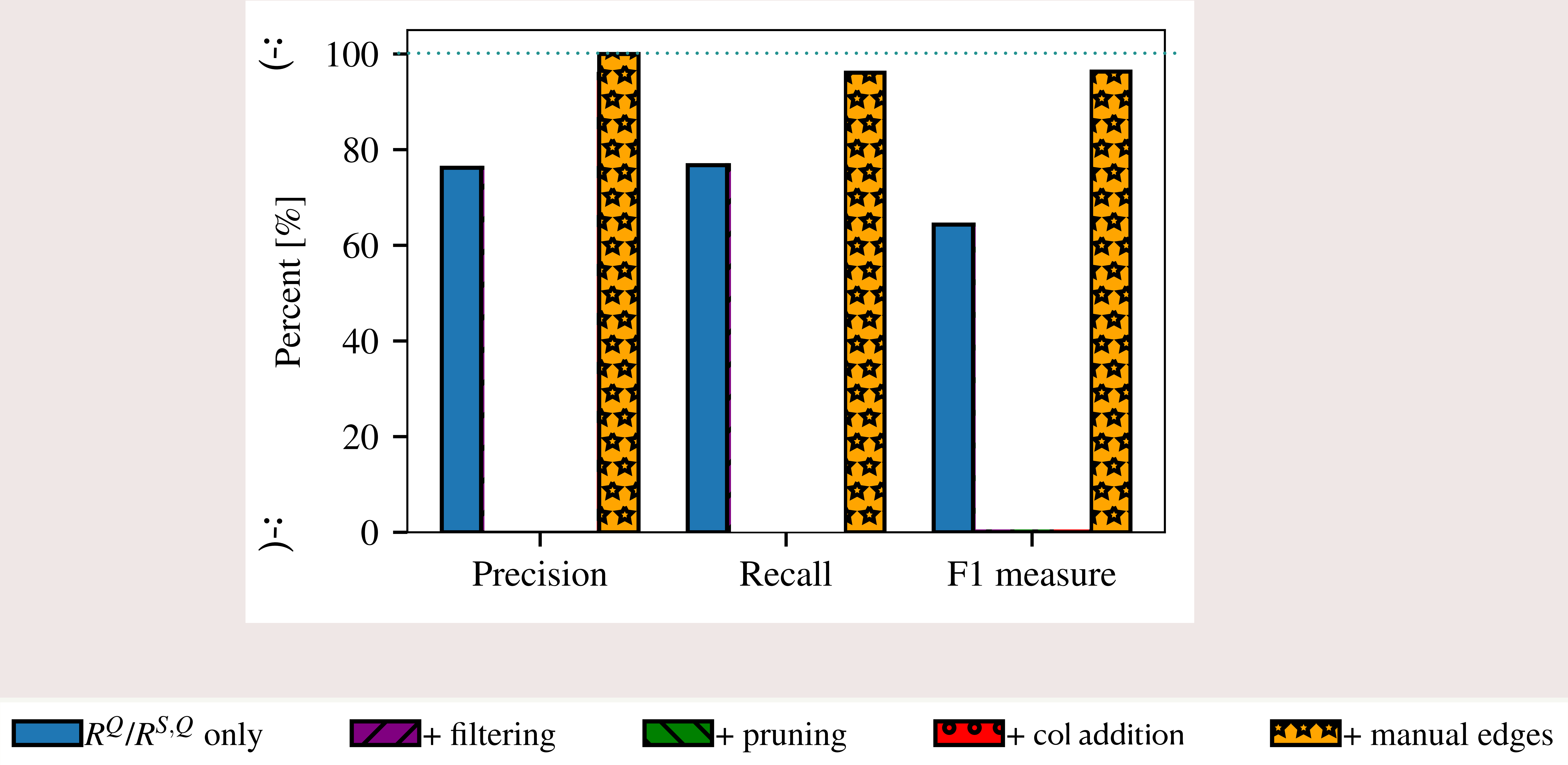
Q1: Does GDPRizer correctly identify user-data ?

HotCRP



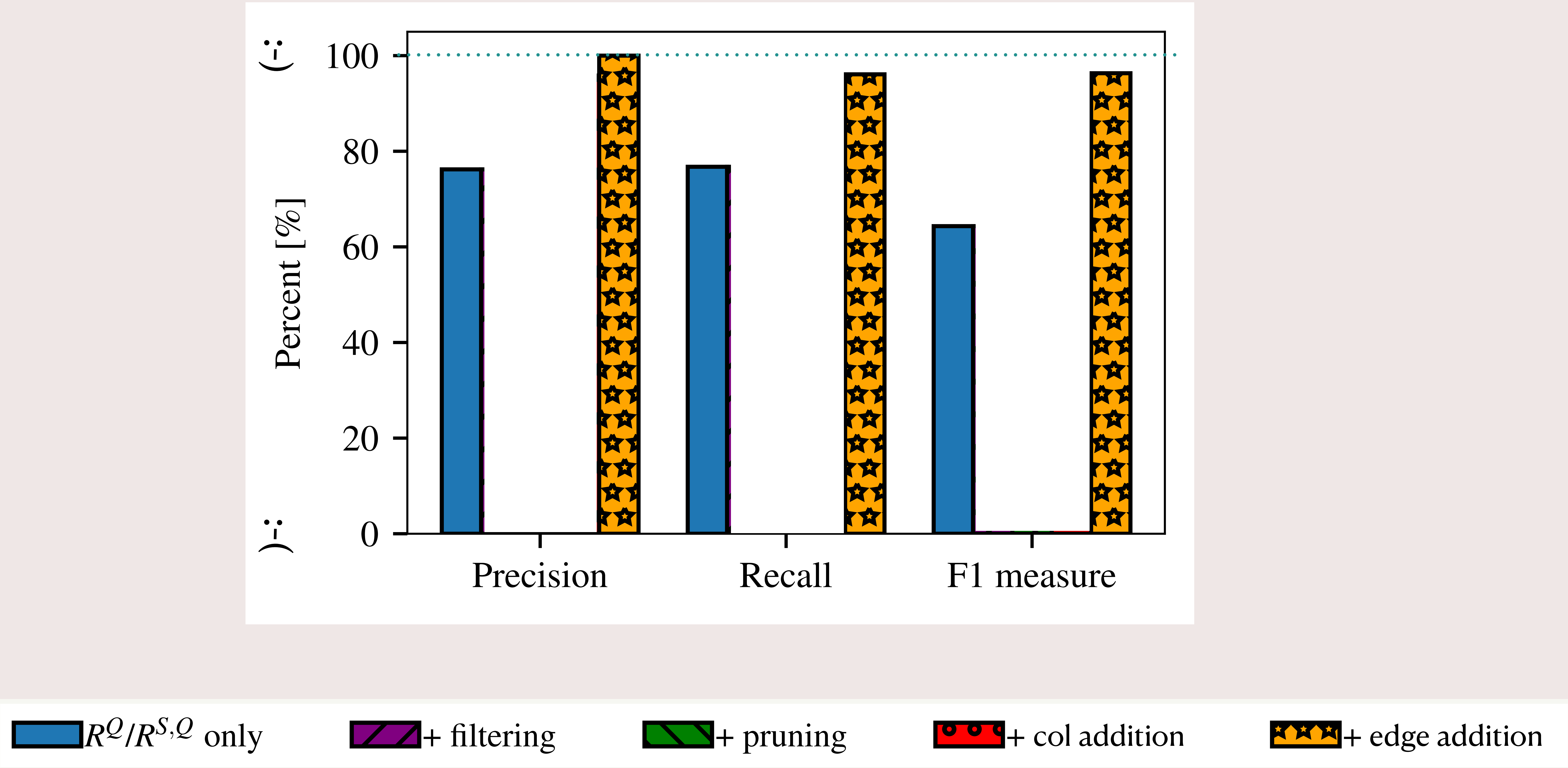
Q1: Does GDPRizer correctly identify user-data ?

HotCRP



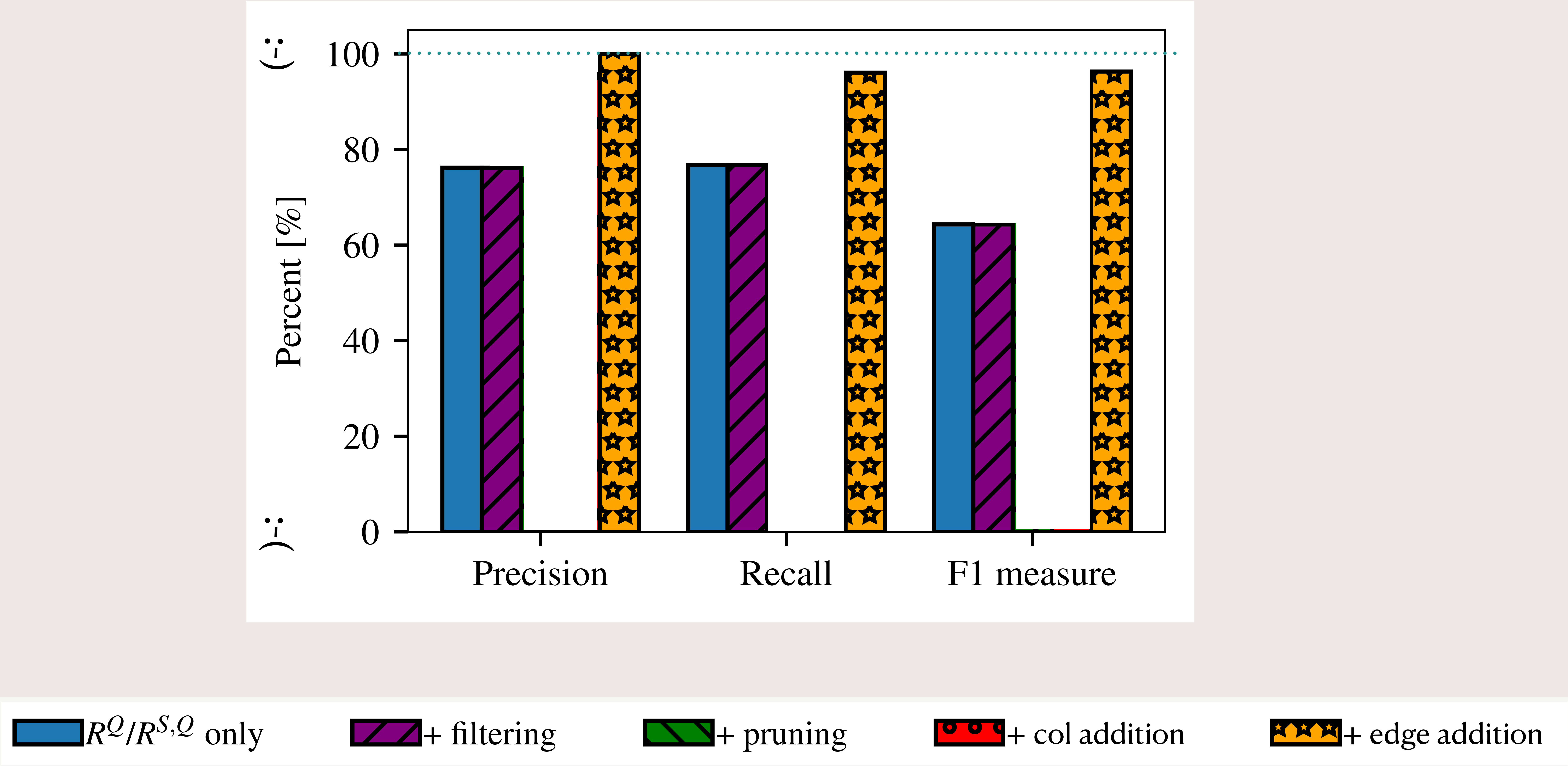
Q2: What is the impact of customizations ?

HotCRP



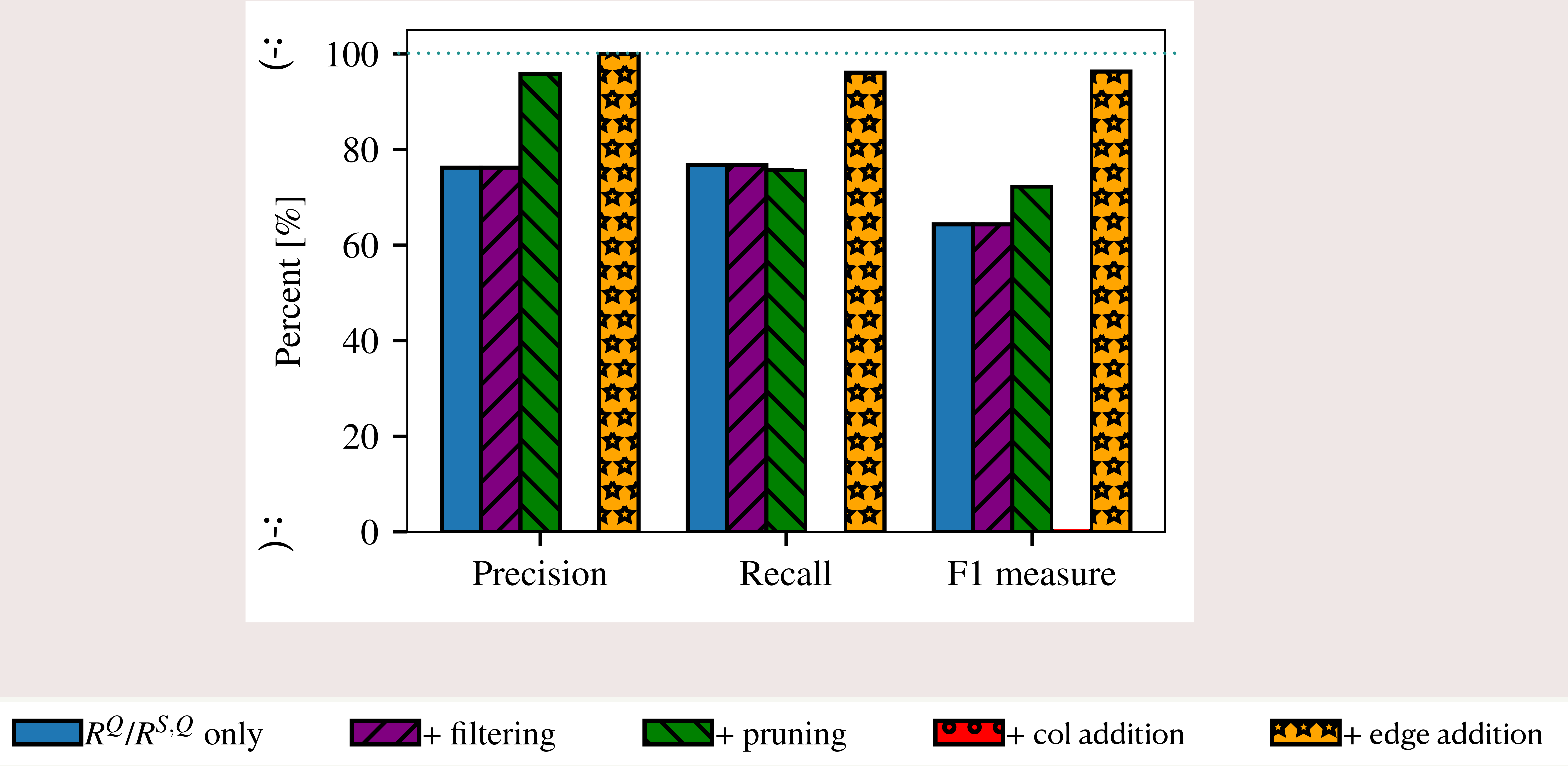
Q2: What is the impact of customizations ?

HotCRP



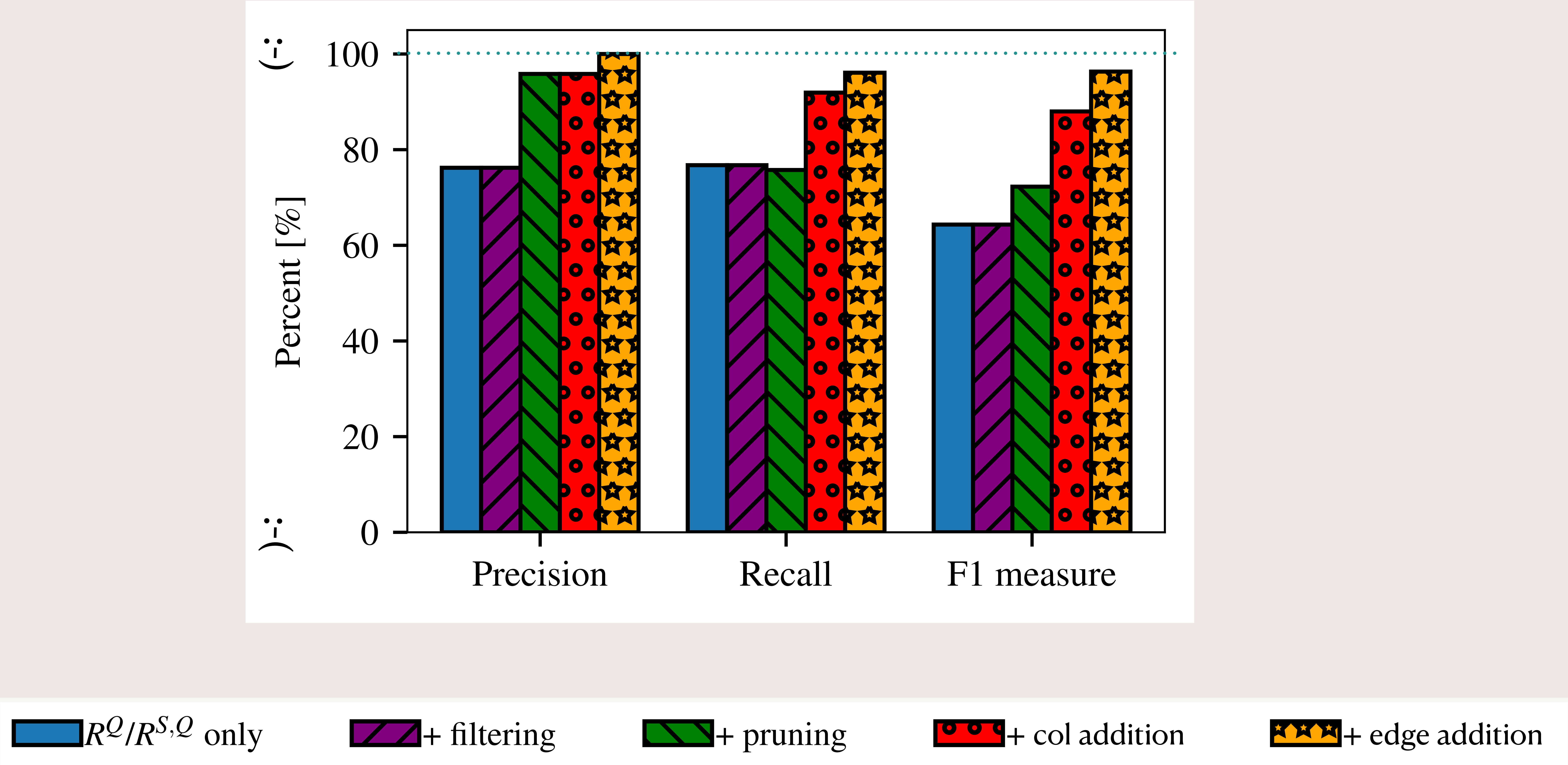
Q2: What is the impact of customizations ?

HotCRP



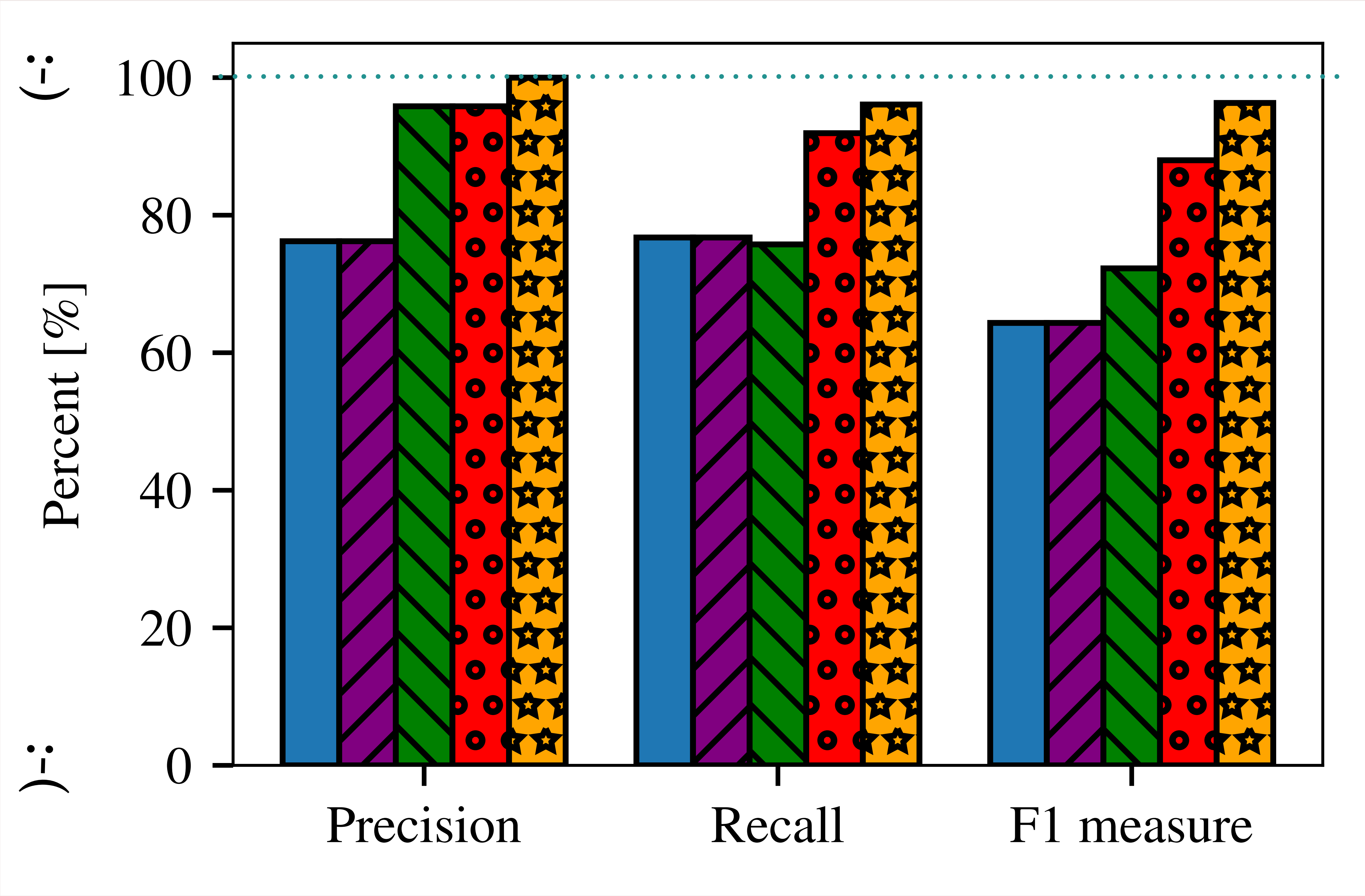
Q2: What is the impact of customizations ?

HotCRP



Q2: What is the impact of customizations ?

HotCRP



Similar results for all the other applications

Q3: How many customizations are needed ?

	Total number of customizations
TPC-H (customer)	4
TPC-H (supplier)	7
HotCRP	31
Lobsters	16
WordPress	4
WordPress (w/ plugins)	12

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

Conclusion by glyph.faisalovers from Noun Project

Conclusion

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

Conclusion by glyph.faisalovers from Noun Project

Conclusion

- GDPRizer : a tool for user-data extraction in legacy databases
- A fully-automated, general solution for legacy systems is unlikely

Conclusion by glyph.faisalovers from Noun Project

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

Conclusion by glyph.faisalovers from Noun Project

Questions?