

Paper Review #2:  
PrivateSQL: A Differentially Private SQL Query Engine

*Summary 1:*

*PrivateSQL: A Differentially Private SQL Query Engine* is about a program that gives databases end-to-end differentially private system. It allows non-experts to specify privacy for their databases at different degrees. Using novel techniques, the program ensures that the privacy on the database matches the data owner's policy.

*Summary 2:*

*PrivateSQL: A Differentially Private SQL Query Engine* is about a program that makes it easier for non-experts to privatize their databases without experts. Usually, database owners need to assemble a team of privacy experts to make a database differently private, but with this program, they do not need to. It's a step closer to having differential privacy on all databases.

Question and Answer

What is the problem?

- The problem is that it's difficult for non-privacy-experts to use differential privacy and would often have to refer to experts create the appropriate algorithms.

Why is it interesting and important?

- It's important because differential privacy is a much-needed property for databases as the amount of information increases online. It ensures that data owners' privacy is kept from adversaries.

Why is it hard?

- It's hard to define privacy and create an algorithm that appropriately accommodates it. Especially when there are relational databases as related tables can have varying amount of private information.

Why hasn't it been solved before?

- There are other programs that help non-experts create private databases, but none other than PrivateSQL can support relational data. This is due to the algorithms being difficult to compute and differential privacy being a fairly new topic.

What are the key components of the proposed approach and results?

- The key idea for PrivateSQL is creating synopses for views, extending differential privacy to relational data, and view rewriting for policy-specific bounds.

## Paper Review #2: DJoin: Differentially Private Join Queries over Distributed Databases

### Summary 1:

*DJoin: Differentially Private Join Queries over Distributed Databases* is about a program that provides differential privacy. DJoin supports SQL queries, mainly joins for multiple databases if they can be expressed using the two novel primitives, BN-PSI-CA and MCR. The program is useful as it allows for correlations to be formed between two datasets with differential privacy.

### Summary 2:

*DJoin: Differentially Private Join Queries over Distributed Databases* is about a program that allows researchers to join two databases without learning private information from either database. By law, databases aren't allowed to share information together as they risk revealing private information about individuals. However, DJoin solves this problem and allows researchers to understand important information from two databases, safely and privately.

### Question and Answer

What is the problem?

- The problem is that there's no practical way to get data from two different databases to form correlations due to privacy concerns.

Why is it interesting and important?

- There is a lot of information that can be learned from two separate databases but is difficult to do without learning private information.

Why is it hard?

- Joining two separate databases to correlate individuals is hard to do without learning about their private information or without using a single trusted server.

Why hasn't it been solved before?

- A lot of research has mainly been focused on differentially private single data bases. Also, joining databases asks about individual rows but differential privacy was designed exactly to protect against that.

What are the key components of the proposed approach and results?

- The key idea behind DJoin is that queries can be expressed as intersections of sets or multisets, allowing the program to rewrite queries that don't require privacy to be leaked.