Adapting Differentially Private Synthetic Data to Relational Databases

Kaveh Alimohammadi[1], **Hao Wang**[2], **Ben Reilly**[3], **Akash Srivastava**[2], **Navid Azizan**[1]*
[1]MIT [2]MIT-IBM Watson AI Lab [3]IBM

Abstract

Existing differentially private (DP) synthetic data generation mechanisms typically assume a single-source table. In practice, data is often distributed across multiple tables with relationships. In this paper, we introduce the first-of-its-kind algorithm that can be combined with any existing DP mechanisms to generate synthetic relational databases. Our algorithm iteratively refines the relationship between individual synthetic tables to minimize their approximation errors in terms of low-order marginal distributions while maintaining referential integrity. Finally, we provide both DP and theoretical utility guarantees for our algorithm.

1 Introduction

Relational databases play a pivotal role in modern information systems and business operations due to their efficiency in managing structured data [SOAK+19]. According to a Kaggle survey [Kag17], 65.5% of users worked extensively with relational data. Additionally, the majority of leading database management systems (e.g., MySQL and Oracle) are built on relational database principles [Ran23]. These systems organize data into multiple tables, each representing a specific entity, and the relationships between tables delineate the connections between these entities. However, the widespread use of relational databases also carries a significant risk of privacy leakage. For example, if a single table suffers from a privacy breach, all other tables containing sensitive information can be exposed, as they are interconnected and share relationships. Moreover, deleting data in a relational database can be complex, and incomplete deletion can leave traces of data, which can potentially be accessed and reconstructed by attackers.

Today, differential privacy (DP) stands as the de facto standard for privacy protection. There is a growing body of research focused on applying DP to generate private synthetic data [see, e.g., RTMT21, MMS21, LVW21, Sma23]. This effort enables data curators to share (synthetic) data while ensuring the privacy of individuals' personal information within the original dataset. In return, they can harness advanced machine learning techniques employed by end-users trained on the synthetic data. Numerous studies have provided evidence that state-of-the-art (SOTA) DP synthetic data effectively preserves both the statistical properties of the original data and the high performance of downstream predictive models trained on these synthetic datasets when deployed on the original data [TMH+21, WSH+23]. However, all existing works assume a single-source database (with a few exceptions discussed in Related Work). It motivates the central question we aim to tackle in this paper:

Can we adapt existing DP synthetic data generation algorithms to relational databases while preserving their referential integrity?

The conventional approach—flattening relational databases into a single master table, generating a synthetic master table, and dividing it into separate databases—presents several challenges. To

[*] Emails: {mrz,azizan}@mit.edu, {hao,ben.reilly,akash.srivastava}@ibm.com. This is a working paper. Comments to improve this work are welcome.

illustrate, consider education data as an example, with two tables (student and teacher information) linked by students' enrollment in a teacher's course. First, flattening may lead to data integrity issues by introducing numerous null values. This makes it difficult to distinguish between missing and intentionally null data. For instance, if a teacher is not currently teaching, their students' information in the master table will be represented by null values. Consequently, the master synthetic table will contain a significant number of null values even if the original tables do not have any null values. Second, flattening introduces concerns related to data scalability and redundancy. Each record in the master table may have an extremely large number of features, which poses challenges for SOTA DP synthetic data generation mechanisms, as their running time escalates rapidly with the growing number of features [e.g., MMSM22, MMS21, VAA+22, ABK+21]. Additionally, certain records from the original tables might be duplicated across multiple entries in the master table, resulting in data redundancy and an increased demand for storage space. Finally, breaking down a synthetic master table may disrupt relationships, especially when there are records with shared feature values. For example, if the master table includes two courses with students sharing identical demographic information, it is unclear whether they are the same student or different students with matching demographic details.

Our goal is to introduce the first-of-its-kind algorithm that can generate synthetic relational databases while preserving privacy, statistical properties, and referential integrity of the original data. For this purpose, we first discuss DP in relational databases[2], extend the definition of k-way marginal queries [TUV12, RTMT21] to relational databases and articulate the requirements of referential integrity (Section 2). The main technical contribution is introducing an iterative algorithm that effectively learns the relationship between various tables (Section 3). At each iteration, the algorithm identifies a subset of k-way marginal queries with the highest approximation errors and then refines the relational synthetic database to minimize these errors. Notably, our algorithm avoids the need to flatten relational databases into a master table; instead, it only requires querying the relational databases to compute low-order marginal distributions, which can be done using SQL aggregate functions. Consequently, we can leverage *any* off-the-shelf DP mechanisms to generate synthetic data for individual tables and apply our algorithm to establish their inter-table relationship. We analyze our algorithm theoretically, establishing both DP and utility guarantees.

Note that existing literature on synthetic relational data generation, even without DP guarantees, is limited [MCM+22, PWV16, XGJ+23]. We hope our efforts can benefit the communities focused on DP synthetic data and inspire new research on this topic.

In summary, our main contributions are as follows.

- We present a pioneering study on privacy-preserving synthetic data generation in relational databases.
- We introduce an iterative algorithm designed to establish relationships between synthetic tables, preserving both statistical properties and referential integrity of the original data.
- Our algorithm is model-agnostic as it can seamlessly integrate with any existing DP synthetic data algorithms. Additionally, we establish both privacy and utility guarantees for our algorithm.

Related Work

Privacy-preserving synthetic data generation is an active research topic [see e.g., BLR13, UV20, BSV22, CXZX15, GMHI20, RLP+20, WSH+23]. For example, a line of work considered learning probabilistic graphical models [ZCP+17, MSM19, MMS21] or generative adversarial networks [XLW+18, BJWW+19, JYVDS19, TWB+19, NWD20, BKZ23] with DP guarantees and generating synthetic data by sampling from these models. Another line of work proposed to iteratively refined synthetic data to minimize its approximation error on a pre-selected set of workload queries [HLM12, GAH+14, MMSM22, LVW21, LVS+21, VTB+20, ABK+21, VAA+22, LTVW23]. They only focused on a single-source dataset without considering relational database. The only exception is [XGJ+23] that leveraged tools from random graph theory and representation learning to generate multiple tables with many-to-many relationships. However, their method required using DP-SGD

[2] We believe this step is crucial, yet it has been overlooked in some previous work [e.g., XGJ+23], given that relational databases consist of both tables and relationships between them. Depending on the definitions of neighboring datasets, there are bounded/unbounded DP [KM11] and edge/node DP [KNRS13]. When applying the composition theorem, these definitions may not always be compatible.

to optimize their generative models for each table generation, whereas our method can seamlessly integrate with *any* existing DP mechanisms for synthetic table generation. This versatility is essential, particularly as marginal-based and workload-based mechanisms often produce higher-quality synthetic tabular data than DP-SGD-based mechanisms [TMH+21, MMS21, WSH+23]. Finally, some research explored DP in relational database systems, primarily focusing on releasing statistical queries [see e.g., He21, JNS18, KTH+19]. In contrast, our emphasis is on releasing a synthetic copy of the relational database, extending the scope of privacy-preserving data generation methods to a more practical and general setting.

Another line of work focused on the release of DP synthetic graphs [LUZ23, ZNF21, EKKL20, Upa13, GRU12, BBDS12, SZW+11, HLMJ09] with the goal of maintaining essential graph properties (e.g., connectivity, degree distribution, and cut approximation). While a relational database can often be represented as a multipartite graph (see Section 2.1), applying these existing approaches to generate synthetic relational databases encounters various challenges. First, these methods often assume that the vertex set of both real and synthetic graphs is common, learning the edge set through DP mechanisms. In our scenario, however, the vertex sets (representing records) of real and synthetic databases differ, requiring privacy preservation for both the vertex set and the edge set. Additionally, existing approaches focus on preserving graph properties, whereas our objective extends to preserve both graph properties (i.e., relationships between different tables) and statistical properties of relational databases.

2 Preliminaries and Problem Formulation

We introduce notation, represent relational databases through bipartite graphs, review differential privacy and its properties, and provide an overview of our problem formulation.

2.1 Bipartite Graph and Relational Database

To simplify our presentation, we assume that the relational database $\mathcal{B}$ consists of only two tables $\mathcal{D}_1$ and $\mathcal{D}_2$ (e.g., student and teacher information), each having n_1 and n_2 rows. The relationship between tables is typically defined through primary and foreign keys. A primary key is a column in a table serving as a unique identifier for each row. A foreign key is a column that establishes a relationship between tables by referencing the primary key of a different table.

We represent the relational database as a bipartite graph $\mathcal{B} = (\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B})$ where each record corresponds to a node; the two tables define two distinct sets of nodes; and the relationships between the tables are depicted as edges connecting these nodes. We represent the edges using a bi-adjacency matrix, denoted as $\boldsymbol{B} \in \{0,1\}^{n_1 \times n_2}$, where $B_{i,j} = 1$ iff the i-th record from table $\mathcal{D}_1$ is connected to the j-th record from table $\mathcal{D}_2$ (e.g. if the i-th student is enrolled in the j-th teacher's course). For each record $\boldsymbol{x}$, we represent its degree as the total number of records from the other table connected to $\boldsymbol{x}$, denoted by $\deg(\boldsymbol{x})$. Finally, we assume the degree of each record is upper bounded by a constant $d_{\max}$.

2.2 Differential Privacy

We first recall the definition of differential privacy (DP) [DR14].

Definition 1. A randomized mechanism $\mathcal{M}$ that takes a relational database as input and returns an output from a set $\mathcal{R}$ satisfies (ε, δ)-differential privacy, if for any adjacent relational databases $\mathcal{B}$ and $\mathcal{B}'$ and all possible subsets $\mathcal{O}$ of $\mathcal{R}$, we have

$$\Pr(\mathcal{M}(\mathcal{B}) \in \mathcal{O}) \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathcal{B}') \in \mathcal{O}) + \delta. \tag{1}$$

Following [JNS18], we consider two relational databases $\mathcal{B}$ and $\mathcal{B}'$ adjacent if $\mathcal{B}'$ can be obtained from $\mathcal{B}$ by selecting a table and changing a single row of this table.

DP plays a pivotal role in answering statistical queries, with two distinct categories to be considered for relational databases. The first class involves queries pertaining to a single table, say table $\mathcal{D}_1$. In the context of a student-teacher database, an example would be a query about whether a student is a freshman. If we denote the data domain of table $\mathcal{D}_1$ by $\mathcal{X}_1$, a statistical query can be represented by a function $q : \mathcal{X}_1 \to \{0,1\}$ and its average over a database is denoted as $q(\mathcal{B}) \triangleq \frac{1}{n_1} \sum_{\boldsymbol{x} \in \mathcal{D}_1} q(\boldsymbol{x})$. The second class includes cross-table queries, such as whether a teacher and a student belong to the same department. Analogously, these queries are represented by a function $q : \mathcal{X}_1 \times \mathcal{X}_2 \to \{0, 1\}$. We denote the average of their values over a database by $q(\mathcal{B}) \triangleq \frac{1}{\mathbf{1}^T \cdot \text{vec}(\boldsymbol{B})} \sum q(\boldsymbol{x}_i, \boldsymbol{x}_j)$ where the sum is over $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{D}_1 \times \mathcal{D}_2$ s.t. $B_{i,j} = 1$ and $\text{vec}(\cdot)$ converts a matrix into a vector. When $(\mathcal{D}_1, \mathcal{D}_2)$ are clear from the context, we also express query in terms of the bi-adjacency matrix $\boldsymbol{B}$.

2.3 Problem Formulation

Our goal is to generate a privacy-preserving synthetic database $\mathcal{B}_{\text{syn}} = (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \boldsymbol{B}^{\text{syn}})$ that preserves both statistical properties and referential integrity of the original relational database. In terms of statistical properties, our aim is to ensure that the marginal distributions of subsets of features in the synthetic data closely align with those of the real data. To achieve this, we revisit the definition of k-way marginal query and extend it to suit relational databases.

Definition 2. Suppose the data domain of table $\mathcal{D}_i$ has d_i categorical features: $\mathcal{X}_i = \mathcal{X}_{i,1} \times \cdots \mathcal{X}_{i,d_i}$. For a subset of k features $\mathcal{S} \subseteq [d_1]$ (or $\mathcal{S} \subseteq [d_2]$) with $|\mathcal{S}| = k$ and a reference value $\boldsymbol{y}$, a single-table k-way marginal query is defined as

$$q_{\mathcal{S},\boldsymbol{y}}(\boldsymbol{x}) \triangleq \prod_{j \in \mathcal{S}} \mathbb{I}(x_j = y_j) \quad \text{for } \boldsymbol{x} \in \mathcal{X}_1 \text{ (or } \mathcal{X}_2) \tag{2}$$

where $\mathbb{I}$ is an indicator function. Analogously, for $\mathcal{S}_1 \times \mathcal{S}_2 \subseteq [d_1] \times [d_2]$ with $|\mathcal{S}_1| + |\mathcal{S}_2| = k$, $0 < |\mathcal{S}_1| < k$, and a reference value $(\boldsymbol{y}_1, \boldsymbol{y}_2)$, a cross-table k-way marginal query is defined as

$$q_{(\mathcal{S}_1,\mathcal{S}_2),(\boldsymbol{y}_1,\boldsymbol{y}_2)}(\boldsymbol{x}_1, \boldsymbol{x}_2) \triangleq \prod_{(j_1,j_2) \in \mathcal{S}} \mathbb{I}((x_{1,j_1}, x_{2,j_2}) = (y_{1,j_1}, y_{2,j_2})) \quad \text{for } (\boldsymbol{x}_1, \boldsymbol{x}_2) \in \mathcal{X}_1 \times \mathcal{X}_2.$$

We denote the set of all single-table k-way marginal queries and cross-table k-way marginal queries by $\mathcal{Q}_{\text{single},k}$ and $\mathcal{Q}_{\text{cross},k}$, respectively.

To demonstrate the size of all k-way marginal queries, consider each categorical feature has a fixed number of distinct values: $|\mathcal{X}_{i,j}| = m$. Then we have

$$|\mathcal{Q}_{\text{single},k}| = \sum_{i \in \{1,2\}} \binom{d_i}{k} m^k, \quad |\mathcal{Q}_{\text{cross},k}| = \sum_{k_1=1}^{k-1} \binom{d_1}{k_1} \binom{d_2}{k_2} m^k.$$

In maintaining referential integrity, we use a linking table to establish relationships. It is designed to store the IDs of synthetic tables: the entry (i, j) is included in this table if $\boldsymbol{B}_{i,j}^{\text{syn}} = 1$. Moreover, when the original database exhibits a one-to-many relationship (i.e., a record in the parent table $\mathcal{D}_1$ can be related to one or more records in $\mathcal{D}_2$, but a record in the child $\mathcal{D}_2$ can be related to *only one* record in $\mathcal{D}_1$) or one-to-one relationship[3], we expect the synthetic database to preserve this relationship. Finally, in the case of a one-to-many relationship in the original database, we ensure there are no orphaned rows in the synthetic database. Specifically, we impose the requirement that each record in the child table $\mathcal{D}_2^{\text{syn}}$ connects to a record in the parent table $\mathcal{D}_1^{\text{syn}}$ (although records in the parent table are permitted to have no associated child records).

3 Main Results

We present our main algorithm (Algorithm 1) for generating privacy-preserving synthetic relational databases. This algorithm can be combined with any existing single-source DP synthetic data mechanisms, adapting them to the relational database context by establishing relationships among individual synthetic tables. Specifically, it learns a bi-adjacency matrix by minimizing approximation errors in terms of cross-table k-way marginal queries compared to the original data. Given the inherent large size of the query class, our algorithm employs an iterative approach to identify the queries with the highest approximation errors and refine the bi-adjacency matrix to reduce these errors. Notably, our algorithm is designed for efficiency and scalability, making it well-suited for

[3] We assume that information about the types of relationships in the original database is publicly available, but it can also be learned differentially privately by examining the degree distribution of each table.

high-dimensional data. It also ensures referential integrity in the generated synthetic relational data. We end this section by establishing rigorous utility and DP guarantees for our algorithm.

We first create individual synthetic tables $\mathcal{D}_i^{\text{syn}}$ by applying any DP mechanisms via black-box access. Our main technical contribution lies in establishing connections among these synthetic tables by constructing a bi-adjacency matrix $\boldsymbol{B}$ that links them. We learn this matrix by aligning the cross-table k-way marginal queries between the synthetic and original databases. The key observation is that each of these marginal queries can be written as a (fractional) linear function of the bi-adjacency matrix. We formalize this observation in the following lemma.

Lemma 1. *For a relational database* $(\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B})$ *and any k-way cross-table query* $q_{(\mathcal{S}_1,\mathcal{S}_2),(\boldsymbol{y}_1,\boldsymbol{y}_2)}$*, we denote* $\mathbf{1}_{(\mathcal{S}_1,\boldsymbol{y}_1)} \in \{0,1\}^{n_1}$ *as an indicator vector whose its i-th element equals 1 iff the i-th record in $\mathcal{D}_1$ satisfies $x_{i,j} = y_{1,j}$ for all $j \in \mathcal{S}_1$. Let* $\boldsymbol{q}_{(\mathcal{S}_1,\mathcal{S}_2),(\boldsymbol{y}_1,\boldsymbol{y}_2)} = \mathrm{vec}(\mathbf{1}_{(\mathcal{S}_1,\boldsymbol{y}_1)} \otimes \mathbf{1}_{(\mathcal{S}_2,\boldsymbol{y}_2)})$*. Then we have:*

$$q_{(\mathcal{S}_1,\mathcal{S}_2),(\boldsymbol{y}_1,\boldsymbol{y}_2)}(\boldsymbol{B}) = \frac{\boldsymbol{q}^T_{(\mathcal{S}_1,\mathcal{S}_2),(\boldsymbol{y}_1,\boldsymbol{y}_2)} \cdot \mathrm{vec}(\boldsymbol{B})}{\mathbf{1}^T \cdot \mathrm{vec}(\boldsymbol{B})}. \tag{3}$$

Based on the above lemma, we write the cross-table query and its corresponding vector $\boldsymbol{q}_{(\mathcal{S}_1,\mathcal{S}_2),(\boldsymbol{y}_1,\boldsymbol{y}_2)}$ interchangeably. We learn the bi-adjacency matrix of the synthetic database by solving the following optimization:

$$\min_{\substack{\boldsymbol{b}^{\text{syn}} \in \{0,1\}^{n_1^{\text{syn}} \cdot n_2^{\text{syn}}} \\ \mathbf{1}^T \boldsymbol{b}^{\text{syn}} = m^{\text{syn}}}} \max_{\boldsymbol{q} \in \mathcal{Q}_{\text{cross},k}} \left| \frac{1}{m^{\text{syn}}} \boldsymbol{q}^T \boldsymbol{b}^{\text{syn}} - a \right|. \tag{4}$$

where $\boldsymbol{b}^{\text{syn}} = \mathrm{vec}(\boldsymbol{B}^{\text{syn}})$ and a is the query value computed from the original real data. Since the minimization over $\boldsymbol{b}^{\text{syn}}$ in (4) is a combinatorial optimization, which is inherently challenging to solve, we solve a relaxed problem by allowing $\boldsymbol{b}^{\text{syn}} \in [0,1]^{n_1^{\text{syn}} \cdot n_2^{\text{syn}}}$ and use a randomized rounding algorithm to convert the obtained values back into integer values (see Appendix D for more details).

We present an iterative algorithm for solving the above optimization with DP guarantees. At each iteration, we apply $\texttt{RN}_K$ (i.e., report noisy top-K mechanism) [DR19] to identify cross-table queries that have not been reported before and have the highest approximation errors between synthetic and real databases. Then we add isotropic Gaussian noise to their query values and update the (vectorized) bi-adjacency matrix $\boldsymbol{b}^{\text{syn}}$ to reduce these approximation errors. Specifically, at iteration t, we stack all the queries reported so far into a matrix $\boldsymbol{Q}^t \in \{0,1\}^{tK \times n_1^{\text{syn}} n_2^{\text{syn}}}$ and let their noisy answers be a vector $\hat{\boldsymbol{a}}^t \in \mathbb{R}^{tK}$. Then we solve the following optimization to update $\boldsymbol{b}^{\text{syn}}$

$$\min_{\substack{\boldsymbol{b}^{\text{syn}} \in [0,1]^{n_1^{\text{syn}} \cdot n_2^{\text{syn}}} \\ \mathbf{1}^T \boldsymbol{b}^{\text{syn}} = m^{\text{syn}}}} \left| \frac{1}{m^{\text{syn}}} \boldsymbol{Q}^t \boldsymbol{b}^{\text{syn}} - \hat{\boldsymbol{a}}^t \right|. \tag{5}$$

We remark that iterative algorithms are a standard technique for query releasing in the DP literature [see GRU12, HLM12, LVW21, ABK+21, MMSM22, for examples in synthetic data/graph generation] and we extend its applications to establish relationships between synthetic tables. We end this section by establishing DP and utility guarantees for our algorithm.

Theorem 1. *Algorithm satisfies* $(\epsilon_1 + \epsilon_2 + \epsilon_{rel}, \delta_1 + \delta_2 + \delta_{rel})$*-DP.*

Theorem 2. *For synthetic tables* $\mathcal{D}_1^{syn}, \mathcal{D}_2^{syn}$ *generated by any DP mechanisms and* $\varepsilon_{rel}, \delta_{rel} > 0$*, we denote the output of our algorithm when* $T = 1$ *and* $K = |\mathcal{Q}_{cross,k}|$ *by* $\hat{\boldsymbol{b}}$ *and the optimal solution of the relaxed optimization problem without any privacy constraints by* $\boldsymbol{b}^*$*. Therefore,* $e_{inh} = \sqrt{\frac{1}{|\mathcal{Q}_{cross,k}|} \|\boldsymbol{Q}\boldsymbol{b} - \hat{\boldsymbol{Q}}\boldsymbol{b}^*\|_2^2}$ *is the error we face without any privacy constraints for learning the*

Algorithm 1 Adapting DP mechanisms to generate relational synthetic data.

```
Input: relational database $\mathcal{B} = (\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B})$; privacy budgets $(\varepsilon_1, \delta_1)$, $(\varepsilon_2, \delta_2)$, $(\varepsilon_{\text{rel}}, \delta_{\text{rel}})$; queries per iteration $K$; number of iterations $T \le \frac{|\mathcal{Q}_{\text{cross, k}}|}{K}$; maximum degree $d_{\text{max}}$; synthetic dataset parameters $n_1^{\text{syn}}, n_2^{\text{syn}}, m^{\text{syn}}$
initialize $\rho_{\text{rel}} = (\sqrt{\varepsilon_{\text{rel}} + \log \frac{1}{\delta_{\text{rel}}}} - \sqrt{\log \frac{1}{\delta_{\text{rel}}}})^2, \boldsymbol{Q} = \emptyset, \hat{\boldsymbol{a}} = \emptyset, \boldsymbol{b}^{\text{syn}} \in [0,1]^{n_1^{\text{syn}} \times n_2^{\text{syn}}}$
$\mathcal{D}_i^{\text{syn}} = \text{SyntheticTableGenration}(\mathcal{D}_i, (\varepsilon_i, \delta_i))$ for $i \in \{1, 2\}$
for $t = 1, \cdots, T$ do
    $\boldsymbol{q}_1, \cdots, \boldsymbol{q}_K = \text{RN}_\text{K}(\mathcal{B}, (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \boldsymbol{b}^{\text{syn}}), \frac{\rho}{2 \cdot T \cdot d_{\text{max}}})$
    for $i = 1, \cdots, K$ do
        $\hat{\boldsymbol{a}}$.append(GaussMech($q_i(\mathcal{B}), \frac{\rho}{2 \cdot T \cdot K \cdot d_{\text{max}}}$)))
        $\boldsymbol{Q}$.append($\hat{\boldsymbol{q}}_i$)
    end for
    $\boldsymbol{b}^{\text{syn}} = \arg\min_{\substack{\boldsymbol{b} \in [0,1]^{n_1^{\text{syn}} \cdot n_2^{\text{syn}}} \\ \boldsymbol{1}^T \boldsymbol{b} = m^{\text{syn}}}} \|\boldsymbol{Q}\boldsymbol{b} - \hat{\boldsymbol{a}}\|_2^2$
    $\boldsymbol{B}^{\text{syn}} = \text{RandomRounding}(\boldsymbol{b}^{\text{syn}})$.reshape
end for
Output: $\mathcal{B}^{\text{syn}} = (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \boldsymbol{B}^{\text{syn}})$
```

relationship between two tables. By this definition, with probability at least $1 - \beta$, we have:

$$
\begin{aligned}
\sqrt{\frac{1}{|\mathcal{Q}_{cross,k}|} \|\hat{\boldsymbol{Q}}\hat{\boldsymbol{b}} - \boldsymbol{Q}\boldsymbol{b}\|_2^2} &\le 2e_{inh} + \frac{2 \cdot (d_{max} \cdot \log(2n_1 n_2 + \log(\frac{1}{\beta}))^{\frac{1}{4}}}{\rho_{rel}{}^{\frac{1}{4}} \cdot \sqrt{n_1, n_2} \cdot m^{\frac{k}{4}}} \\
&\le 2e_{inh} + \frac{2 \cdot (d_{max} \cdot \log(2n_1 n_2 + \log(\frac{1}{\beta}))^{\frac{1}{4}}}{\sqrt{(\sqrt{\varepsilon_{rel} + \log \frac{1}{\delta_{rel}}} - \sqrt{\log \frac{1}{\delta_{rel}}}) \cdot n_1 n_2} \cdot m^{\frac{k}{4}}} \\
&\le 2e_{inh} + O(\frac{d_{max}^{\frac{1}{4}}}{\sqrt{n_1 n_2}})
\end{aligned}
$$

Using the Gaussian mechanism for answering m queries with L_2 sensitivity s leads to a mean squared error of $O(s \cdot \sqrt{m})$ [DKM+06]. Hence, we allocate the privacy budget in proportion to $\frac{1}{n_i}\sqrt{\mathcal{Q}^i_{\text{single},k}}$ for generating synthetic tables and $\frac{d_{\text{max}}}{n_1 n_2}\sqrt{\mathcal{Q}_{\text{cross},k}}$ for learning their relationship.

4 Conclusion and Limitations

In this paper, we investigated synthetic relational database generation with DP guarantees. We proposed an iterative algorithm that can be combined with any preexisting single-table generation mechanisms to maintain cross-table statistical properties and referential integrity. We derived rigorous utility guarantees for our algorithm. Furthermore, we conducted comprehensive experiments, introduced new benchmark datasets and evaluation metrics to assess the performance and scalability of our algorithm. We hope our effort can inspire new research and push the frontiers of DP synthetic data towards more practical scenarios.

There are several intriguing directions that deserve further exploration. First, we assumed any record in the relational database must be kept private. It would be interesting to explore scenarios in which only specific tables contain personal private information. For example, in the context of education data, it may be pertinent to examine situations where only teacher and student information requires privacy-preserving, while course and department information, being often publicly available, does not. Second, our algorithm generates individual synthetic tables and uses iterative algorithm to establish their relationship. One extension would be integrating synthetic table generation into the iterative algorithm, learning both single-table and cross-table queries simultaneously. However, achieving this may require white-box access to the generative models. Finally, while we generate synthetic relational database with the goal of preserving low-order marginal distributions, there are other criteria worthy of exploration, such as logical consistency, temporal dynamics, and user-defined constraints.

5 Impact Statements

This paper focuses on privacy-preserving synthetic data generation, building upon prior efforts in DP synthetic data. We extend their application to a more practical scenario where data is stored in a relational database. This line of research may significantly impact critical domains, such as finance, healthcare, and government, where safeguarding data privacy is paramount. The deployment of DP synthetic data can lead to more inclusive research practices, allowing organizations to share data without compromising the privacy of individuals' personal information. This, in turn, facilitates collaboration and propels advancements in research and business applications reliant on data-driven insights.

It is crucial to acknowledge potential challenges and ethical considerations associated with the deployment of synthetic data. Synthetic data proves to be a suitable substitute for tasks that do not necessitate absolute precision, such as data visualization, software testing, and initial model development. Nevertheless, there is often a trade-off between preserving the essential statistical properties of the original data and meeting privacy requirements. For applications with individual-level consequences, any methods or insights derived from synthetic data need to be carefully evaluated to avoid unintended consequences and biases in downstream applications.

In summary, while our research advances the field of DP synthetic data generation, we recognize the importance of ethical considerations and the potential societal impact. By mitigating privacy concerns in data sharing, this work aspires to play a pivotal role in fostering the responsible and reliable development of advanced machine learning technologies.

References

[ABK$^+$21] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*, pages 457–467. PMLR, 2021.

[BBDS12] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The Johnson-Lindenstrauss transform itself preserves differential privacy. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 410–419. IEEE, 2012.

[BJWW$^+$19] Brett K Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P Bhavnani, James Brian Byrd, and Casey S Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7):e005122, 2019.

[BKZ23] Alex Bie, Gautam Kamath, and Guojun Zhang. Private GANs, revisited. *Transactions on Machine Learning Research*, 2023.

[BLR13] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):1–25, 2013.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part I*, pages 635–658. Springer, 2016.

[BSV22] March Boedihardjo, Thomas Strohmer, and Roman Vershynin. Privacy of synthetic data: A statistical framework. *IEEE Transactions on Information Theory*, 69(1):520–527, 2022.

[CLS21] Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM (JACM)*, 68(1):1–39, 2021.

[CXZX15] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. Differentially private high-dimensional data publication via sampling-based inference. In *ACM SIGKDD international conference on knowledge discovery and data mining*, pages 129–138, 2015.

[DKM$^+$06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[DR19] David Durfee and Ryan M Rogers. Practical differentially private top-k selection with pay-what-you-get composition. *Advances in Neural Information Processing Systems*, 32, 2019.

[EKKL20] Marek Eliáš, Michael Kapralov, Janardhan Kulkarni, and Yin Tat Lee. Differentially private release of synthetic graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–578. SIAM, 2020.

[GAH$^+$14] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In *International Conference on Machine Learning*, pages 1170–1178. PMLR, 2014.

[GMHI20] Chang Ge, Shubhankar Mohapatra, Xi He, and Ihab F Ilyas. Kamino: Constraint-aware differentially private data synthesis. *arXiv preprint arXiv:2012.15713*, 2020.

[GRU12] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *Theory of Cryptography: 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings 9*, pages 339–356. Springer, 2012.

[He21] Xi He. Differential privacy for complex data: Answering queries across multiple data tables. https://www.nist.gov/blogs/cybersecurity-insights/differential-privacy-complex-data-answering-queries-across-multiple, 2021.

[HLM12] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems*, 25, 2012.

[HLMJ09] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *2009 Ninth IEEE International Conference on Data Mining*, pages 169–178. IEEE, 2009.

[JNS18] Noah Johnson, Joseph P Near, and Dawn Song. Towards practical differential privacy for sql queries. *Proceedings of the VLDB Endowment*, 11(5):526–539, 2018.

[JYVDS19] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2019.

[Kag17] Kaggle. Kaggle 2017 survey results. https://www.kaggle.com/code/amberthomas/kaggle-2017-survey-results, 2017.

[KM11] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011.

[KNRS13] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Theory of Cryptography: 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*, pages 457–476. Springer, 2013.

[KTH$^+$19] Ios Kotsogiannis, Yuchao Tao, Xi He, Maryam Fanaeepour, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. Privatesql: a differentially private sql query engine. *Proceedings of the VLDB Endowment*, 12(11):1371–1384, 2019.

[LTVW23] Terrance Liu, Jingwu Tang, Giuseppe Vietri, and Steven Wu. Generating private synthetic data with genetic algorithms. In *International Conference on Machine Learning*, pages 22009–22027. PMLR, 2023.

[LUZ23] Jingcheng Liu, Jalaj Upadhyay, and Zongrui Zou. Optimal bounds on private graph approximation. *arXiv preprint arXiv:2309.17330*, 2023.

[LVS⁺21] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. Leveraging public data for practical private query release. In *International Conference on Machine Learning*, pages 6968–6977. PMLR, 2021.

[LVW21] Terrance Liu, Giuseppe Vietri, and Steven Z Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 34:690–702, 2021.

[MCM⁺22] Ciro Antonio Mami, Andrea Coser, Eric Medvet, Alexander TP Boudewijn, Marco Volpe, Michael Whitworth, Borut Svara, Gabriele Sgroi, Daniele Panfilo, and Sebastiano Saccani. Generating realistic synthetic relational data through graph variational autoencoders. *arXiv preprint arXiv:2211.16889*, 2022.

[MMS21] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021.

[MMSM22] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: An adaptive and iterative mechanism for differentially private synthetic data. *arXiv preprint arXiv:2201.12677*, 2022.

[MSM19] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pages 4435–4444. PMLR, 2019.

[NTZ13] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 351–360, 2013.

[NWD20] Marcel Neunhoeffer, Zhiwei Steven Wu, and Cynthia Dwork. Private post-gan boosting. *arXiv preprint arXiv:2007.11934*, 2020.

[PWV16] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410. IEEE, 2016.

[Ran23] DB-Engines Ranking. DB-engines ranking - popularity ranking of database management systems. https://db-engines.com/en/ranking, 2023.

[RLP⁺20] Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. Differentially private synthetic data: Applied evaluations and enhancements. *arXiv preprint arXiv:2011.05537*, 2020.

[RTMT21] Diane Ridgeway, Mary F Theofanos, Terese W Manley, and Christine Task. Challenge design and lessons learned from the 2018 differential privacy challenges. *NIST Technical Note 2151*, 2021.

[Sch13] Rolf Schneider. *Convex Bodies: The Brunn–Minkowski Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 2013.

[Sma23] SmartNoise. Smartnoise sdk: Tools for differential privacy on tabular data. https://github.com/opendp/smartnoise-sdk, 2023.

[SOAK⁺19] Maximilian Schleich, Dan Olteanu, Mahmoud Abo-Khamis, Hung Q Ngo, and XuanLong Nguyen. Learning models over relational data: A brief tutorial. In *Scalable Uncertainty Management: 13th International Conference, SUM 2019, Compiègne, France, December 16–18, 2019, Proceedings 13*, pages 423–432. Springer, 2019.

[SZW$^+$11] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 81–98, 2011.

[TMH$^+$21] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238*, 2021.

[TUV12] Justin Thaler, Jonathan Ullman, and Salil Vadhan. Faster algorithms for privately releasing marginals. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2012.

[TWB$^+$19] Uthaipon Tantipongpipat, Chris Waites, Digvijay Boob, Amaresh Ankit Siva, and Rachel Cummings. Differentially private mixed-type data generation for unsupervised learning. *arXiv preprint arXiv:1912.03250*, 1:13, 2019.

[Upa13] Jalaj Upadhyay. Random projections, graph sparsification, and differential privacy. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 276–295. Springer, 2013.

[UV20] Jonathan Ullman and Salil Vadhan. PCPs and the hardness of generating synthetic data. *Journal of Cryptology*, 33(4):2078–2112, 2020.

[VAA$^+$22] Giuseppe Vietri, Cedric Archambeau, Sergul Aydore, William Brown, Michael Kearns, Aaron Roth, Ankit Siva, Shuai Tang, and Steven Z Wu. Private synthetic data for multitask learning and marginal queries. *Advances in Neural Information Processing Systems*, 35:18282–18295, 2022.

[VTB$^+$20] Giuseppe Vietri, Grace Tian, Mark Bun, Thomas Steinke, and Steven Wu. New oracle-efficient algorithms for private synthetic data release. In *International Conference on Machine Learning*, pages 9765–9774. PMLR, 2020.

[WSH$^+$23] Hao Wang, Shivchander Sudalairaj, John Henning, Kristjan Greenewald, and Akash Srivastava. Post-processing private synthetic data for improving utility on selected measures. In *Conference on Neural Information Processing Systems*, 2023.

[XGJ$^+$23] Kai Xu, Georgi Ganev, Emile Joubert, Rees Davison, Olivier Van Acker, and Luke Robinson. Synthetic data generation of many-to-many datasets via random graph generation. In *International Conference on Learning Representations*, 2023.

[XLW$^+$18] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

[ZCP$^+$17] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. PrivBayes: Private data release via Bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

[ZNF21] Sen Zhang, Weiwei Ni, and Nan Fu. Differentially private graph publishing with degree distribution preservation. *Computers & Security*, 106:102285, 2021.

A General Framework

Our framework, in its most generic form, can be represented as follows:

Algorithm 2 relational database, generic.

```
Input: relational database $\mathcal{B} = (\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B})$; set of queries $\mathcal{Q}$; privacy budgets $(\varepsilon_1, \delta_1)$, $(\varepsilon_2, \delta_2)$, $(\varepsilon_{\text{rel}}, \delta_{\text{rel}})$; queries per iteration $K$; number of iterations $T \leq \frac{|\mathcal{Q}|}{K}$; maximum degree $d_{\text{max}}$; synthetic dataset parameters $n_1^{\text{syn}}, n_2^{\text{syn}}$; linear constraints $\boldsymbol{C}, \boldsymbol{d}$
Let $\mathcal{D}_i^{\text{syn}} = \text{SyntheticTableGenration}(\mathcal{D}_i, (\varepsilon_i, \delta_i)))$ for $i \in \{1, 2\}$
initialize $\rho_{\text{rel}} = (\sqrt{\varepsilon_{\text{rel}}/2 + \log \frac{2}{\delta_{\text{rel}}}} - \sqrt{\log \frac{2}{\delta_{\text{rel}}}})^2$, $\boldsymbol{Q}$, $\hat{\boldsymbol{a}} = \emptyset$, $\boldsymbol{b}^{\text{syn}} \in [0,1]^{n_1^{\text{syn}} \times n_2^{\text{syn}}}$
for $t = 1, \cdots, T$ do
    Let $q_1, \cdots, q_K = RN_K((\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B}), (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \hat{\boldsymbol{B}}_{t-1}), q^{\mathcal{Q}/\hat{\boldsymbol{Q}}_m}, \frac{\rho_{\text{rel}}}{2 \cdot T \cdot d_{\text{max}}})$
    for $i = 1, \cdots, K$ do
        Let $\hat{a}_i = \text{DPM}(q_i(\mathcal{B}), \frac{1}{2 \cdot T \cdot K \cdot d_{\text{max}}} \cdot (\varepsilon_{\text{rel}}, \delta_{\text{rel}})))$
        Let $\hat{\boldsymbol{Q}}_m = \hat{\boldsymbol{Q}}_m \cup \{\hat{\boldsymbol{q}}_i\}$, Where $\hat{\boldsymbol{q}}_i$ is the vectorized of query $q_i$ w.r.t $(\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2)$
    end for
    Let $\hat{\boldsymbol{b}}_t^r = \arg\min_{\substack{\hat{\boldsymbol{b}} \in [0,1]^{n_1 \cdot n_2} \\ \boldsymbol{C}\hat{\boldsymbol{b}} = \boldsymbol{d}}} \mathcal{L}(\hat{\boldsymbol{Q}}_m \hat{\boldsymbol{b}} - \boldsymbol{a})$
    Let $\hat{\boldsymbol{B}}_t = \text{Rounding}(\hat{\boldsymbol{b}}_t^r)$
end for
Output: $\mathcal{B} = (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \hat{\boldsymbol{B}}_T)$
```

In this general framework,

- $\mathcal{Q}$ can be any set of queries that can be represented as a statistical query over edges, i.e., each query corresponds to a set of edges and counts how many of them appear in the bi-adjacency matrix. Cross-table k-way marginal queries are a specific example, but there are many others, such as degree and joint-degree queries. (What else?)
- We can use any differential private mechanism, $\text{DPM}(\text{x}, (\varepsilon, \delta)$, that outputs x with (ε, δ)-DP guarantee. We used the Gaussian mechanism in the main algorithm and will use the Laplace mechanism in a sequel for another instance of our algorithm. In Theorem 3, we will show that the algorithm would satisfy differential privacy for any choice of DP mechanism.
- Loss function $\mathcal{L}(.)$ can be any loss function. In Theorem 4, we will show that this problem would be a convex optimization for any convex loss function and thus can be solved efficiently.
- One can use any Rounding function to convert the weighted bi-adjacency matrix according to the problem's context. We used a randomized rounding technique for the main algorithm on many-to-many databases of the paper, and in Appendix D, we will discuss another technique for one-to-many databases.

Theorem 3. *The above algorithm satisfies (ε, δ)-DP*

Theorem 4. *For any convex loss function, the projection step is a convex optimization problem*

A.1 Manhattan Norm

In this section, we present a variation of our primary algorithm, delineated in 3, specifically designed to minimize the l_1 error, diverging from the Mean Squared Error (MSE) utilized in the original version. Opting for the loss function $\mathcal{L}(\cdot) = \| \cdot \|_1$, the corresponding projection challenge transforms into a linear programming problem that will be proved in the forthcoming theorem 6. This formulation enables highly efficient optimization problem solving, as detailed in [CLS21]. Moreover, within the context of the l_1 norm, introducing Laplace noise is imperative to effectively minimize the l_1 error, in contrast to the Gaussian noise applied in the case of MSE.

Theorem 5. *Algorithm 3 satisfies (ε, δ)-DP.*

Theorem 6. *The projection step is linear programming for l_1 norm as the loss function.*

Algorithm 3 relational database, l_1 norm.

```
Input: relational database $\mathcal{B} = (\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B})$; set of queries $\mathcal{Q}$; privacy budgets $(\varepsilon_1, \delta_1)$, $(\varepsilon_2, \delta_2)$, $(\varepsilon_{\text{rel}}, \delta_{\text{rel}})$; queries per iteration $K$; number of iterations $T \leq \frac{|\mathcal{Q}|}{K}$; maximum degree $d_{\text{max}}$; synthetic dataset parameters $n_1^{\text{syn}}, n_2^{\text{syn}}, m^{\text{syn}}$
initialize $\hat{\boldsymbol{Q}}_m = \emptyset$, $\hat{\boldsymbol{B}}_0 \in [0,1]^{n_1^{\text{syn}} \times n_2^{\text{syn}}}$
Let $\mathcal{D}_i^{\text{syn}} = \text{SyntheticTableGenration}(\mathcal{D}_i, (\varepsilon_i, \delta_i))$ for $i \in \{1, 2\}$
for $t = 1, \cdots, T$ do
    Let $q_1, \cdots, q_K = \text{RN}_\text{K}((\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B}), (\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2, \hat{\boldsymbol{B}}_{t-1}), q^{\mathcal{Q}/\hat{\boldsymbol{Q}}_m}, \frac{1}{2 \cdot T \cdot d_{\text{max}}} \cdot (\varepsilon_{\text{rel}}, \delta_{\text{rel}}))$
    for $i = 1, \cdots, K$ do
        Let $\hat{a}_i = \text{LM}(q_i(\mathcal{B}), \frac{1}{2 \cdot T \cdot K \cdot d_{\text{max}}} \cdot (\varepsilon_{\text{rel}}, \delta_{\text{rel}}))$
        Let $\hat{\boldsymbol{Q}}_m = \hat{\boldsymbol{Q}}_m \cup \{\hat{\boldsymbol{q}}_i\}$, Where $\hat{\boldsymbol{q}}_i$ is the vectorized of query $q_i$ w.r.t $(\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}})$
    end for
    Let $\hat{\boldsymbol{b}}_t^r = \arg\min_{\substack{\hat{\boldsymbol{b}} \in [0,1]^{n_1^{\text{syn}} \cdot n_2^{\text{syn}}} \\ \mathbf{1}^T \hat{\boldsymbol{b}} = m'}} \|\hat{\boldsymbol{Q}}_m \hat{\boldsymbol{b}} - \boldsymbol{a}\|_1$
    Let $\hat{\boldsymbol{B}}_t = \text{RandomRounding}(\hat{\boldsymbol{b}}_t^r)$
end for
Output: $\mathcal{B} = (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \hat{\boldsymbol{B}}_T)$
```

Proof. Let $N = n_1^{\text{syn}} \cdot n_2^{\text{syn}}$, and $M = |\boldsymbol{Q}|$ be number of queries. Projection problem for l_1 is:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{Q}\boldsymbol{b} - \boldsymbol{a}\|_1 \\ \text{subject to} \quad & \sum_{i=1}^{N} \boldsymbol{b}_i = m \\ & 0 \leq \boldsymbol{b}_i \leq 1\,, \quad \forall i = 1, \cdots, N \end{aligned}$$

Now consider the following minimization problem with variables $x \in \mathbb{R}, \boldsymbol{s} \in \mathbb{R}^M, \boldsymbol{b} \in \mathbb{R}^N$. First, note that the objective function is linear, and also, all the constraints are linear in variables; therefore, in fact, this is a linear programming problem with $N + M + 1$ variables and $2M + 2N + 2$ constraints, and hence, it is solvable in $O^*((M+N)^{2+\frac{1}{6}})$ [CLS21]. We argue that $\boldsymbol{s}_j^* = |\boldsymbol{q}_j^T \boldsymbol{b}^* - \boldsymbol{a}_j|$ and therefore the objective function in the optimal point would be $\sum |\boldsymbol{q}_j^T \boldsymbol{b}^* - \boldsymbol{a}_j| = \|\boldsymbol{Q}\boldsymbol{b} - \boldsymbol{a}\|_1$. Note that because of the first set of inequalities, we have that and $\forall \boldsymbol{b}, x, \forall j : \boldsymbol{s}_j \geq |\boldsymbol{q}_j^T \boldsymbol{b} - \boldsymbol{a}_j|$. Also note $\forall \boldsymbol{b} : \boldsymbol{s}_j^* = |\boldsymbol{q}_j^T \boldsymbol{b} - \boldsymbol{a}_j\|_\infty$ satisfies the inequalities, and we conclude that this problem is minimizing the Manhattan norm of vector $\boldsymbol{Q}_m \boldsymbol{b} - a$ while satisfying the constraint on $\boldsymbol{b}$, and is equivalent to the former formulation of the problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{M} \boldsymbol{s}_j \\ \text{subject to} \quad & -\boldsymbol{s}_j \leq \hat{\boldsymbol{q}}_j^T \cdot \boldsymbol{b} - \boldsymbol{a}_j \leq \boldsymbol{s}_j, \quad \forall j = 1, \cdots, M \\ & \sum_{i=1}^{N} \boldsymbol{b}_i = m \\ & 0 \leq \boldsymbol{b}_i \leq 1 \qquad\qquad\quad , \quad \forall i = 1, \cdots, N \end{aligned}$$

□

A.2 Max Norm

As stated in Equation 4, we are interested in solving the min-max problem; one natural approach that can be integrated into our framework is to use the Max norm for projection. In this section, we will introduce our algorithm's max norm variation. Further, we will show that, like the projection concerning the Manhattan norm, this projection is also linear programming and, thus, can be solved efficiently.

Theorem 7. *The projection step is linear programming for l_∞ norm as the loss function.*

Algorithm 4 l_∞ variation.

```
Input:
Input: relational database $\mathcal{B} = (\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B})$; set of queries $\mathcal{Q}$; privacy budgets $(\varepsilon_1, \delta_1)$, $(\varepsilon_2, \delta_2)$, $(\varepsilon_{\text{rel}}, \delta_{\text{rel}})$; queries per iteration $K$; number of iterations $T \leq \frac{|\mathcal{Q}|}{K}$; maximum degree $d_{\max}$; synthetic dataset parameters $n_1^{\text{syn}}, n_2^{\text{syn}}, m^{\text{syn}}$
initialize $\hat{\boldsymbol{Q}}_m = \emptyset$, $\hat{\boldsymbol{B}}_0 \in [0,1]^{n_1^{\text{syn}} \times n_2^{\text{syn}}}$
$\mathcal{D}_i^{\text{syn}} = \text{SyntheticTableGenration}(\mathcal{D}_i, (\varepsilon_i, \delta_i))$ for $i \in \{1, 2\}$
for $t = 1, \cdots, T$ do
    $q_1, \cdots, q_K = \text{RN}_{\text{K}}((\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B}), (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \boldsymbol{B}_{t-1}^{\text{syn}}), q^{\mathcal{Q}/\hat{\boldsymbol{Q}}_m}, \frac{\rho}{2 \cdot T \cdot d_{\max}})$
    for $i = 1, \cdots, K$ do
        $\hat{a}_i = \text{DPM}(q_i(\mathcal{B}), \frac{\rho}{2 \cdot T \cdot K \cdot d_{\max}}))$
        $\hat{\boldsymbol{Q}}_m = \hat{\boldsymbol{Q}}_m \cup \{\hat{\boldsymbol{q}}_i\}$, where $\hat{\boldsymbol{q}}_i$ is the vectorized of query $q_i$ w.r.t $(\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}})$
    end for
    $\hat{\boldsymbol{b}}_t^r = \arg\min_{\substack{\hat{\boldsymbol{b}} \in [0,1]^{n_1^{\text{syn}} \cdot n_2^{\text{syn}}} \\ \mathbf{1}^T \hat{\boldsymbol{b}} = m^{\text{syn}}}} \|\hat{\boldsymbol{Q}}_m \hat{\boldsymbol{b}} - \boldsymbol{a}\|_\infty$
    $\hat{\boldsymbol{B}}_t = \text{RandomRounding}(\hat{\boldsymbol{b}}_t^r)$
end for
Output: $\mathcal{B}^{\text{syn}} = (\mathcal{D}_1^{\text{syn}}, \mathcal{D}_2^{\text{syn}}, \boldsymbol{B}_T^{\text{syn}})$
```

Proof. Let $N = n_1^{\text{syn}} \cdot n_2^{\text{syn}}$, and $M = |\boldsymbol{Q}|$ be number of queries. Projection problem for l_∞ is:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{Q}\boldsymbol{b} - \boldsymbol{a}\|_\infty \\ \text{subject to} \quad & \sum_{i=1}^{N} \boldsymbol{b}_i = m \\ & 0 \leq \boldsymbol{b}_i \leq 1 \quad , \quad \forall i = 1, \cdots, N \end{aligned}$$

Now consider the following minimization problem with variables $x \in \mathbb{R}, \boldsymbol{b} \in \mathbb{R}^N$. First, note that the objective function is linear, and also, all the constraints are linear in variables; therefore, in fact, this is a linear programming problem with $N+1$ variables and $2M + 2N + 2$ constraints, and hence, it solvable in $O^*((M+N)^{2+\frac{1}{6}})$ [CLS21]. We argue that $x^* = \|\boldsymbol{Q}\boldsymbol{b}^* - \boldsymbol{a}\|_\infty$. Note that because of the first set of inequalities, we have that and $\forall \boldsymbol{b}, x, \forall j : x \geq \|\hat{\boldsymbol{q}}_j \boldsymbol{b} - \boldsymbol{a}_j\|_\infty$ and therefore $\forall \boldsymbol{b}, x : x \geq \|\boldsymbol{Q}\boldsymbol{b} - \boldsymbol{a}\|_\infty$. Also note $\forall \boldsymbol{b} : x^* = \|\boldsymbol{Q}\boldsymbol{b} - \boldsymbol{a}\|_\infty$ satisfies the inequalities, and we conclude that this problem is minimizing the max norm of vector $\boldsymbol{Q}\boldsymbol{b} - \boldsymbol{a}$ while satisfying the constraint on $\boldsymbol{b}$, and is equivalent to the former formulation of the problem.

$$\begin{aligned} \text{minimize} \quad & x \\ \text{subject to} \quad & -x \leq \boldsymbol{q}_j^T \cdot \boldsymbol{b} - \boldsymbol{a}_j \leq x, \quad \forall j = 1, \cdots, M \\ & \sum_{i=1}^{N} \boldsymbol{b}_i = m \\ & 0 \leq \boldsymbol{b}_i \leq 1 \qquad\qquad , \quad \forall i = 1, \cdots, N \end{aligned}$$

□

B Projection Utility Theorem

B.1 Proof of Theorem 2

Lemma 2. *(contractive property of convex sets projection [Sch13]) Let $\mathcal{K}$ be a non-empty closed convex subset of $\mathbb{R}^d$, then for any $\boldsymbol{x}_1, \boldsymbol{x}_2$ if we name their projections $\boldsymbol{y}_1, \boldsymbol{y}_2$ respectively, i.e. $\boldsymbol{y}_i = \arg\min_{\boldsymbol{y} \in \mathcal{K}} \|\boldsymbol{x}_i - \boldsymbol{y}\|_2^2$, then we have*

$$\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2^2 \leq \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2$$

Proof. Let $\boldsymbol{v} := \boldsymbol{y}_2 - \boldsymbol{y}_2 \neq o$. The function f defined by $f(t) := |\boldsymbol{x}_1 - (\boldsymbol{y}_1 + t\boldsymbol{v})|^2$ for $t \in [0, 1]$ has a minimum at $t = 0$, hence $f'(0) \geq 0$. This gives $\langle \boldsymbol{x}_1 - \boldsymbol{y}_1, \boldsymbol{v} \rangle \leq 0$. Similarly we obtain $\langle \boldsymbol{x}_2 - \boldsymbol{y}_2, \boldsymbol{v} \rangle \geq 0$. Thus, the segment $[\boldsymbol{x}_1, \boldsymbol{x}_2]$ meets the two hyperplanes that are orthogonal to $\boldsymbol{v}$ and that go through $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$, respectively. □

Lemma 3. *Assume $\mathcal{K}$ is a symmetric convex body, let $\boldsymbol{b}^* = \arg\min_{\boldsymbol{b}\in\mathcal{K}} \|\hat{\boldsymbol{Q}}\boldsymbol{b} - \boldsymbol{a}\|_2^2$, $\hat{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}\in\mathcal{K}} \|\hat{\boldsymbol{Q}}\boldsymbol{b} - (\boldsymbol{a} + \boldsymbol{w})\|_2^2$, and $\widetilde{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}\in\mathcal{K}} \|\hat{\boldsymbol{Q}}\boldsymbol{b} - (\hat{\boldsymbol{Q}}\boldsymbol{b}^* + \boldsymbol{w})\|_2^2$, then*

$$\|\hat{\boldsymbol{Q}}\hat{\boldsymbol{b}} - \hat{\boldsymbol{Q}}\boldsymbol{b}^*\|_2^2 \leq \|\boldsymbol{w}\|_2^2$$

Proof. Use triangle inequality and then lemma 2 □

Lemma 4. *(Lemma 1 in [NTZ13]) Assume $\mathcal{K}$ is a symmetric convex body, let $\boldsymbol{b}^* = \arg\min_{\boldsymbol{b}\in\mathcal{K}} \|\hat{\boldsymbol{Q}}\boldsymbol{b} - \boldsymbol{a}\|_2^2$, and $\widetilde{\boldsymbol{b}} = \arg\min_{\boldsymbol{b}\in\mathcal{K}} \|\hat{\boldsymbol{Q}}\boldsymbol{b} - (\hat{\boldsymbol{Q}}\boldsymbol{b}^* + \boldsymbol{w})\|_2^2$ then $\|\hat{\boldsymbol{Q}}\widetilde{\boldsymbol{b}} - \hat{\boldsymbol{Q}}\boldsymbol{b}^*\|_2^2 \leq 4\|\boldsymbol{w}\|_{\mathcal{K}^\circ}$, where $\|\boldsymbol{w}\|_{\mathcal{K}^\circ} = \max_{\boldsymbol{x}\in\mathcal{K}}\langle \boldsymbol{x}, \boldsymbol{w}\rangle$*

Lemma 5. *Let $X_1, \ldots, X_n$ be $\mathcal{N}(0, \sigma^2)$ normal random variables with mean zero. Then $\mathbb{P}\left\{\max_{1\leq i\leq n} X_i \geq \sqrt{2\sigma^2(\log n + t)}\right\} \leq e^{-t}$*

Proof. Let $u := \sqrt{2\sigma^2(\log n + t)}$. We have

$$\begin{aligned}
\mathbb{P}\left\{\max_{1\leq i\leq n} X_i \geq u\right\} &= \mathbb{P}\left\{\exists i, X_i \geq u\right\} \\
&\leq \sum_{i=1}^{n} \mathbb{P}\left\{X_i \geq u\right\} \\
&\leq n e^{-\frac{u^2}{2\sigma^2}} = e^{-t}
\end{aligned}$$

□

Proof. of Theorem 2

Let $N = n_1 \cdot n_2$ and $\mathcal{K} = \hat{Q}B_\infty^N$ where $B_\infty^N \triangleq \left\{x \in \mathbb{R}^N : \|x\|_\infty \leq 1\right\}$ be the N-dimensional ℓ_∞ ball. Also we denote j-th column of $\hat{\boldsymbol{Q}}$ by $\boldsymbol{p}_j$ for $j = 1, \cdots, N$. By combining Lemma 3 and 4 we have:

$$\begin{aligned}
\|\boldsymbol{Q}\boldsymbol{b} - \hat{\boldsymbol{Q}}\hat{\boldsymbol{b}}\|_2^2 &\leq \|\boldsymbol{Q}\boldsymbol{b} - \hat{\boldsymbol{Q}}\boldsymbol{b}^*\|_2^2 + \|\hat{\boldsymbol{Q}}\boldsymbol{b}^* - \hat{\boldsymbol{Q}}\hat{\boldsymbol{b}}\|_2^2 \\
&\leq 2\|\boldsymbol{Q}\boldsymbol{b} - \hat{\boldsymbol{Q}}\boldsymbol{b}^*\|_2^2 + 4\|\boldsymbol{w}\|_{\mathcal{K}^\circ} \qquad\qquad 2
\end{aligned}$$

Furthermore, we can show that

$$\begin{aligned}
\|\boldsymbol{w}\|_{\mathcal{K}^\circ} &= \max_{\boldsymbol{x}\in\mathcal{K}}\langle \boldsymbol{x}, \boldsymbol{w}\rangle \\
&\leq \max_{j=1}^{N} |\langle \boldsymbol{p}_j, \boldsymbol{w}\rangle| \\
&\leq \max_{j=1}^{N} \langle \pm\boldsymbol{p}_j, w\rangle
\end{aligned}$$

Where the first equality is the definition of the dual norm, and inequality follows from the fact that K is a polytope and maximum occurs at its vertices, which are $\{\pm p_j\}_{j=1}^N$. Now let

$$X_j \triangleq \langle p_j, w\rangle, X_{N+j} \triangleq \langle -p_j, w\rangle, \text{for } j = 1, \cdots, N$$

Since each of these random variables is a linear combination of independent zero mean Gaussian random variables, they are also zero mean Gaussian random variables as well, and we have

$$\begin{aligned}
\mathbf{Var}(X_j) &= \mathbf{Var}\Big(\sum_{i=1}^{|Q_{\text{cross, k}}|} w_i \mathbb{I}(e_j \in q_i)\Big) \\
&= \left(\binom{d_1 + d_2}{k} - \binom{d_1}{k} - \binom{d_2}{k}\right)\mathbf{Var}(w_i) \\
&= \frac{|Q_{\text{cross, k}}|}{m^k} \cdot \frac{d_{\text{max}} \cdot |Q_{\text{cross, k}}|}{2N^2 \rho_{\text{rel}}}
\end{aligned}$$

Now Lemma 5 implies that with probability at least $1-\beta$, we have

$$\begin{aligned}\frac{1}{|Q_{\text{cross, k}}|}\|\hat{Q}\widetilde{b}-\hat{Q}b^*\|_2^2 &\le \frac{1}{|Q_{\text{cross, k}}|}\cdot 4\max_{j=1}^{2N} X_j \\ &\le \frac{4\sqrt{d_{\max}}}{\sqrt{\rho}\cdot n_1 n_2\cdot m^{\frac{k}{2}}}\cdot\sqrt{\log(2n_1n_2+\log(\frac{1}{\beta}))}\end{aligned}$$

□

C Background and DP Theorems

We recall the concept of (zero) concentrated differential privacy. It will be used in the proof of Theorem 1 for establishing DP guarantees for our algorithm.

Definition 3. (Zero Concentrated Differential Privacy (zCDP) [BS16]). An algorithm $\mathcal{M}:\mathcal{X}^n\to R$ satisfies ρ-zero Concentrated Differential Privacy (zCDP) if for all pairs of neighboring datasets $D,D'\in\mathcal{X}^n$, and for all $\alpha\in(0,\infty)$:

$$\mathbb{D}_\alpha\left(\mathcal{M}(D),\mathcal{M}\left(D'\right)\right)\le\rho\alpha$$

where $\mathbb{D}_\alpha\left(\mathcal{M}(D),\mathcal{M}\left(D'\right)\right)$ denotes the α-Renyi divergence between the distributions $\mathcal{M}(D)$ and $\mathcal{M}\left(D'\right)$.

zCDP enjoys clean composition and postprocessing properties. Moreover, zCDP is tailored to the Gaussian noise that we are using in our algorithm:

Lemma 6. *(Composition [BS16]). Let $\mathcal{M}:\mathcal{X}^n\to\mathcal{Y}$ and $\mathcal{M}':\mathcal{X}^n\to\mathcal{Z}$ be randomized algorithms. Suppose $\mathcal{M}$ satisfies ρ-zCDP and $\mathcal{M}'$ satisfies ρ'-zCDP. Define $\mathcal{M}'':\mathcal{X}^n\to\mathcal{Y}\times\mathcal{Z}$ by $\mathcal{M}''(x)=(M(x),\mathcal{M}'(x))$. Then $\mathcal{M}''$ satisfies $(\rho+\rho')$-zCDP.*

Lemma 7. *(Postprocessing [BS16]). Let $\mathcal{M}:\mathcal{X}^n\to\mathcal{Y}$ and $f:\mathcal{Y}\to\mathcal{Z}$ be randomized algorithms. Suppose $\mathcal{M}$ satisfies ρ-zCDP. Define $M':\mathcal{X}^n\to\mathcal{Z}$ by $\mathcal{M}'(x)=f(\mathcal{M}(x))$. Then $\mathcal{M}'$ satisfies ρ-zCDP.*

Lemma 8. *(Relation between zCDP and DP [BS16].) If $\mathcal{M}$ provides ρ-zCDP, then $\mathcal{M}$ is $(\rho+2\sqrt{\rho\log(1/\delta)},\delta)$-DP for any $\delta>0$.*

Definition 4. (Gaussian Mechanism). The Gaussian mechanism $\mathcal{G}(D,q,\rho)$ takes as input a dataset $D\in\mathcal{X}^*$, a statistical query $q:\mathcal{X}^*\to[0,1]$, and a zCDP parameter ρ. It outputs noisy answer $\hat{a}=q(D)+Z$, where $Z\sim\mathcal{N}\left(0,\frac{1}{2n^2\rho}\right)$, where n is the number of rows in D.

Lemma 9. *(Gaussian Mechanism Privacy [BS16]). Let $q:\mathcal{X}^n\to\mathbb{R}$ be a sensitivity-Δ query. Consider the mechanism $\mathcal{M}:\mathcal{X}^n\to\mathbb{R}$ that on input x, releases a sample from $\mathcal{N}\left(q(x),\sigma^2\right)$. Then $\mathcal{M}$ satisfies $\left(\Delta^2/2\sigma^2\right)$-zCDP.*

Definition 5. (One-shot Report Noisy Top-K with Gumbel Noise [DR19]). The "Report Noisy Top-K" mechanism $\mathrm{RN}_K(D,\widehat{D},Q,\rho)$, takes as input a dataset $D\in\mathcal{X}^n$ with n rows, a synthetic dataset $\widehat{D}\in\mathcal{X}^*$, a set of m statistical queries $Q=\{q_1,\ldots,q_m\}$, and a zCDP parameter ρ. First, it adds Gumbel noise to the error of each $q_i\in Q$:

$$\hat{y}_i=\left|q_i(D)-q_i(\widehat{D})\right|+Z_i,\ \text{where } Z_i\sim\text{Gumbel}(K/\sqrt{2\rho}n),$$

Let $i_{(1)},\ldots,i_{(m)}$ be an ordered set of indices such that $\hat{y}_{i_{(1)}}\ge,\ldots,\ge\hat{y}_{i_{(m)}}$. The algorithm outputs the top-K indices $\left\{i_{(1)},\ldots,i_{(K)}\right\}$ corresponding to the K queries where the answers between D and $\hat{D}$ differ most.

Lemma 10. *(zCDP guarantees of RN_K [ABK+21]) For a dataset D, a synthetic dataset $\widehat{D}$, a set of statistical queries Q, and zCDP parameter ρ, $\mathrm{RN}_K(D,\widehat{D},Q,\rho)$ satisfies ρ-zCDP.*

Lemma 11. *(Basic Composition Theorem) Suppose $M=(M_1,\ldots,M_k)$ is a sequence of algorithms, where M_i is (ε_i,δ_i) differentially private, and the M_i's are potentially chosen sequentially and adaptively. Then M is $\left(\sum_{i=1}^k\varepsilon_i,\sum_{i=1}^k\delta_i\right)$-differentially private.*

Now, we are going to prove Theorem 1

Proof. The privacy of the Algorithm, as stated in Theorem 1, follows from the tools we introduced in this section. Each iteration makes one call to report noisy Top-K and K separate calls to the Gaussian mechanism. By construction and by Lemmas 9 and 10, each of these calls satisfies $\frac{\rho}{2K}$-zCDP, and together by the composition Lemma 6, satisfy $\frac{\rho}{K}$-zCDP. The algorithm then makes a call to the projection algorithm and rounding algorithm, which is a post-processing of the composition of the Gaussian mechanism with report noisy TopK, and so does not increase the zCDP parameter by Lemma7. The loop runs $T \cdot K$ times, and so the entire algorithm satisfies ρ-zCDP by the composition Lemma 6. Lemma 8, and choosing $\rho = (\sqrt{\varepsilon_{\text{rel}} + \log \frac{1}{\delta_{\text{rel}}}} - \sqrt{\log \frac{1}{\delta_{\text{rel}}}})^2$ ensures that our algorithm satisfies $(\varepsilon_{\text{rel}}, \delta_{\text{rel}})$ differential privacy as desired.

□

Next, we will provide a privacy guarantee for Theorem 3

Proof. The proof follows from the same procedure as the previous proof and follows from the tools we introduced in this section; the only difference is that we are considering similar lemmas for approximate differential privacy, as ρ-zCDP is designed to be tailored to the Gaussian mechanism. Each iteration makes one call to report noisy Top-K and K separate calls to the Gaussian mechanism.

□

D Proper Discretization for Referential Integrity

First, we will delve deeper into the randomized algorithm used in the main algorithm. We want to sample m edges out of all possible edges. We use an algorithm called "Rejection Sampling". At each iteration, we sample with the given distribution; if the output is not in the set of edges, we add that edge. Otherwise, we draw a new random number again.

Algorithm 5 Randomized Rounding for Many-to-Many databases.

```
Input: relaxed bi-adjacency $\boldsymbol{b}^r \in [0,1]^{n_1,n_2} \mathbf{1}^T \boldsymbol{b}^r = m$
$m^u = 0, \mathcal{E}^u = \emptyset$
while $m^u < m$ do
    Let $\boldsymbol{x}$ be a random variable that takes value $j$ with probability $\frac{\boldsymbol{b}^r_j}{m}$
    if $\boldsymbol{x} \notin \mathcal{E}^u$ then
        $\mathcal{E}^u = \mathcal{E}^u \cup \{\boldsymbol{x}\}, \quad m^u = m^u + 1$
    end if
end while
$\boldsymbol{B}$ = Reshape $\mathcal{E}^u$ to a $n_1 \times n_2$ matrix
Output: unweighted bi-adjacency matrix with $m$ edges $\boldsymbol{B}$
```

This sampling can be done using another algorithm that is essentially similar to this algorithm but is much more efficient and is based on properties of exponential random variables.

Algorithm 6 Exponential Rounding for Many-to-Many databases.

```
Input: relaxed bi-adjacency $\boldsymbol{b}^r \in [0,1]^{n_1 \cdot n_2} \mathbf{1}^T \boldsymbol{b}^r = m$
Generate $N = n_1 \cdot n_2$, independent random numbers, $X_1, \cdots, X_N$, where $X_i$ is generated from exp($\boldsymbol{b}^r_i$)
Sort $X_1, \cdots, X_N$ from smallest to largest
Let $\mathcal{E}$ be set of indices of the $m$ smallest numbers.
$\boldsymbol{B}$ = Reshape $\mathcal{E}^u$ to a $n_1 \times n_2$ matrix
Output: unweighted bi-adjacency matrix with $m$ edges $\boldsymbol{B}$
```

The time complexity of this method is $O(N \log N)$ whereas the simple-and-reject approach presented in Algorithm 5 takes at least $\Omega\left(N^2\right)$ time.

Theorem 8. *Algorithm 6 and 5 are equivalent.*

Proof. To prove that these two procedures are equivalent, we must first remember the following properties of exponential random variables.

1. If X is exponential with parameter λ, and Y is independently exponential with parameter μ, then $\min(X, Y)$ is exponential with parameter $\lambda + \mu$.
2. With the same assumptions as the last point, $P(X < Y) = \frac{\lambda}{\lambda+\mu}$.
3. The exponential distribution is memoryless, in the sense that $P(X \leq s + t \mid X > t) = P(X \leq s)$ for all $s, t \geq 0$.

This implies that, for example, the probability that the first sampled element is X_1 is

$$P(X_1 < \min(X_2, X_3, \ldots, X_N)) = \frac{\hat{\boldsymbol{b}}_1^r}{\hat{\boldsymbol{b}}_1^r + \left(\hat{\boldsymbol{b}}_2^r + \cdots + \hat{\boldsymbol{b}}_N^r\right)}$$

Then, for subsequent samples, you can do a similar calculation where you leave $\hat{\boldsymbol{b}}_1^r$ out (because of memorylessness) and conclude that the probability of each potential outcome happening is similar in both algorithms. □

Now, we turn our attention to an adapted version of the primary algorithm tailored for databases where each entry in the first table is linked to just one entry in the second table, known as one-to-many databases, as outlined in Algorithm 7. This adaptation introduces two notable distinctions from the main algorithm. Firstly, we modify the projection algorithm to seek solutions that maintain a fixed degree of one for each entry in the first dataset. We introduce a linear equation for each entry into the projection problem to accomplish this, as elaborated in Algorithm 2. It's worth mentioning that this adjusted projection step and the additional linear constraints remain computationally manageable and can be solved using mirror descent. Secondly, in Algorithm 8, we adopt a strategy that selects precisely one edge from the set of edges connected to each entry in the initial dataset. This approach yields an unbiased estimator for each query.

Algorithm 7 Adapting DP mechanisms to generate one-to-many relational synthetic data

```
Input: relational database $\mathcal{B} = (\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B})$; privacy budgets $(\varepsilon_1, \delta_1)$, $(\varepsilon_2, \delta_2)$, $(\varepsilon_{\text{rel}}, \delta_{\text{rel}})$; queries per iteration $K$; number of iterations $T \leq \frac{|\mathcal{Q}_{\text{cross, k}}|}{K}$; maximum degree $d_{\max}$; synthetic dataset parameters $n'_1, n'_2, m'$
$\hat{\mathcal{D}}_i = \text{SyntheticTableGenration}(\mathcal{D}_i, (\varepsilon_i, \delta_i)))$ for $i \in \{1, 2\}$
$\rho_{\text{rel}} = (\sqrt{\varepsilon_{\text{rel}} + \log \frac{1}{\delta_{\text{rel}}}} - \sqrt{\log \frac{1}{\delta_{\text{rel}}}})^2$, $\hat{\boldsymbol{Q}}_m = \emptyset$, and $\hat{\boldsymbol{B}}_0 \in [0,1]^{n'_1 \times n'_2}$ be an arbitrary initialization.
for $t = 1, \cdots, T$ do
    $q_1, \cdots, q_K = \text{RN}_{\text{K}}((\mathcal{D}_1, \mathcal{D}_2, \boldsymbol{B}), (\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2, \hat{\boldsymbol{B}}_{t-1}), q^{\mathcal{Q}_{\text{cross, k}}/\hat{\boldsymbol{Q}}_m}, \frac{\rho}{2 \cdot T \cdot d_{\max}})$
    for $i = 1, \cdots, K$ do
        $\hat{a}_i = \text{GM}(q_i(\mathcal{B}), \frac{\rho}{2 \cdot T \cdot K \cdot d_{\max}}))$
        $\hat{\boldsymbol{Q}}_m = \hat{\boldsymbol{Q}}_m \cup \{\hat{\boldsymbol{q}}_i\}$, Where $\hat{\boldsymbol{q}}_i$ is the vectorized of query $q_i$ w.r.t $(\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2)$
    end for
    $\hat{\boldsymbol{b}}_t^r = \arg\min_{\substack{\hat{\boldsymbol{b}} \in [0,1]^{n_1 \cdot n_2} \\ \boldsymbol{C}\hat{\boldsymbol{b}} = \boldsymbol{d}}} \|\hat{\boldsymbol{Q}}_m \hat{\boldsymbol{b}} - \boldsymbol{a}\|_2^2$
    $\hat{\boldsymbol{B}}_t = \text{CategoricalRounding}(\hat{\boldsymbol{b}}_t^r)$
end for
Output: $\mathcal{B} = (\hat{\mathcal{D}}_1, \hat{\mathcal{D}}_2, \hat{\boldsymbol{B}}_T)$
```

Algorithm 8 Categorical Rounding for One-to-Many databases.

```
Input: relaxed bi-adjacency $\boldsymbol{b}^r \in [0,1]^{n_1,n_2}\mathbf{1}^T, \boldsymbol{b}^r = m$
Let $\boldsymbol{B}^r$ = Reshape $\boldsymbol{b}^r$ to a $n_1 \times n_2$ matrix
$\boldsymbol{B} = [0]^{n_1 \times n_2}$
for $i = 1, \cdots, n_1$ do
    Let $\boldsymbol{x}_i$ be a random variable that takes value $j$ with probability $\boldsymbol{B}^r_{i,j}$
    Let $\boldsymbol{B}_{i,\boldsymbol{x}_i} = 1$
end for
Output: unweighted bi-adjacency matrix with one edge for each entry in the first table $\boldsymbol{B}$
```