Lower Bounds for (Batch) PIR with Private Preprocessing

Kevin Yeo

Google | Columbia University

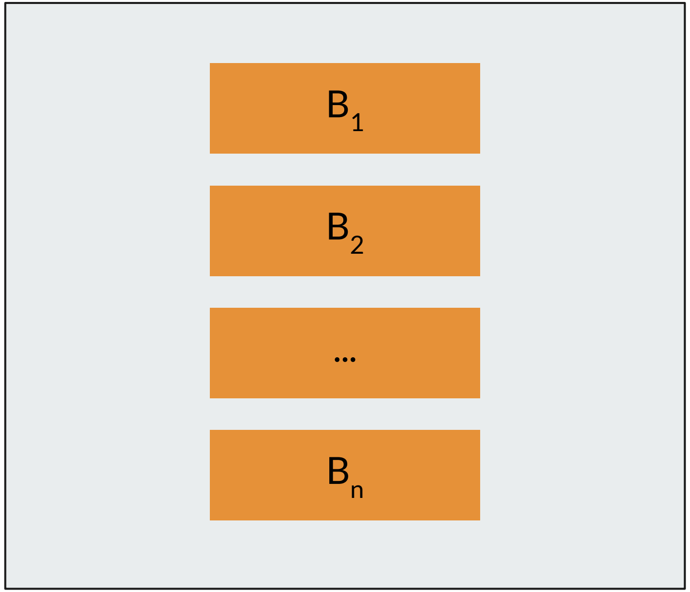
Outline

(Batch) PIR with Private Preprocessing

Our Contributions

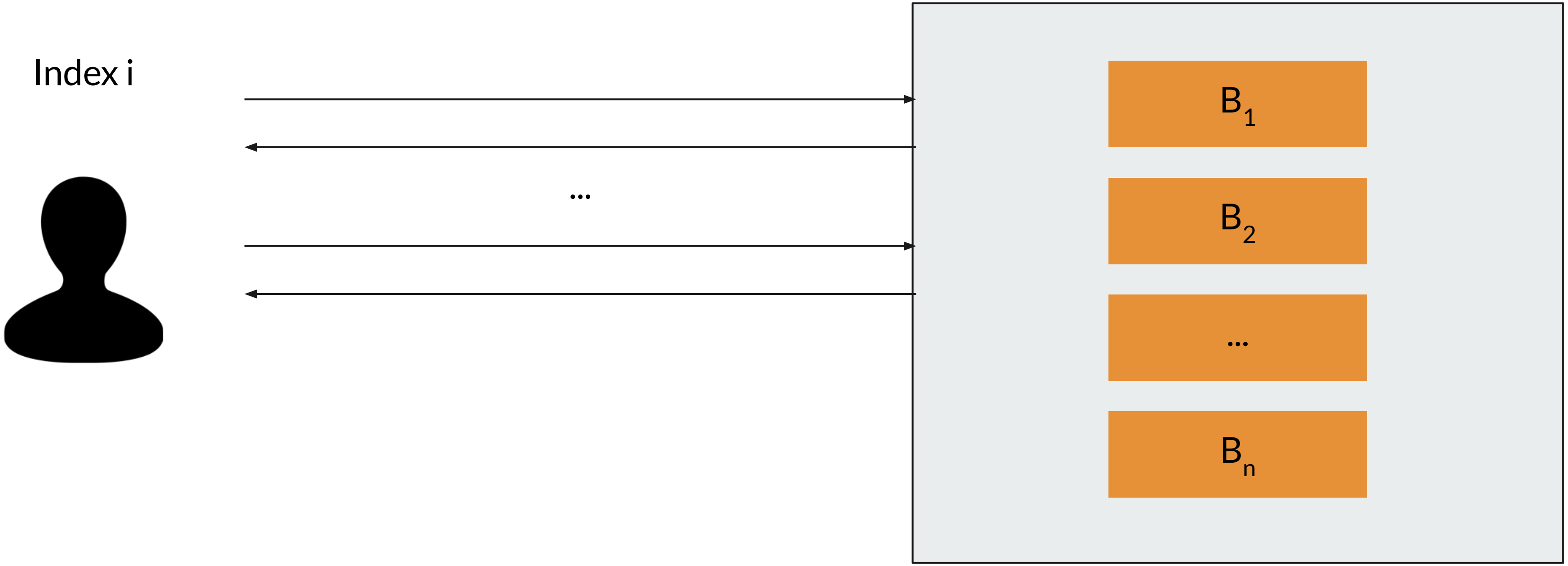
Lower Bound Proof

Private Information Retrieval

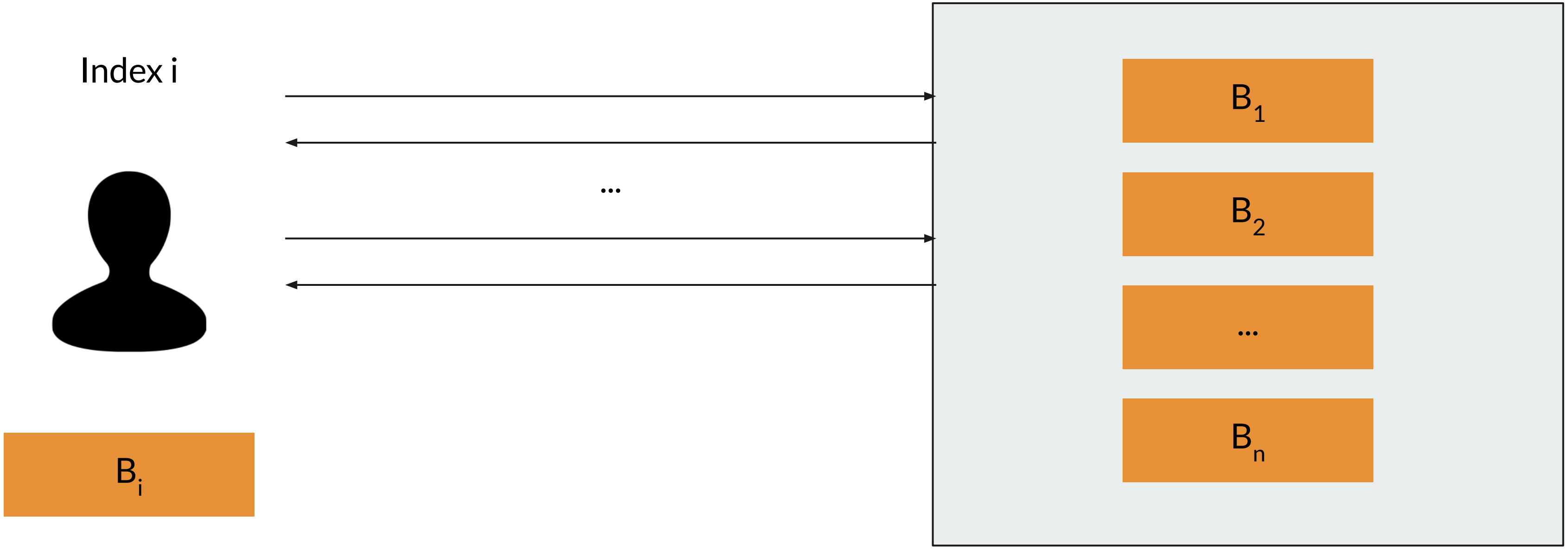


Private Information Retrieval

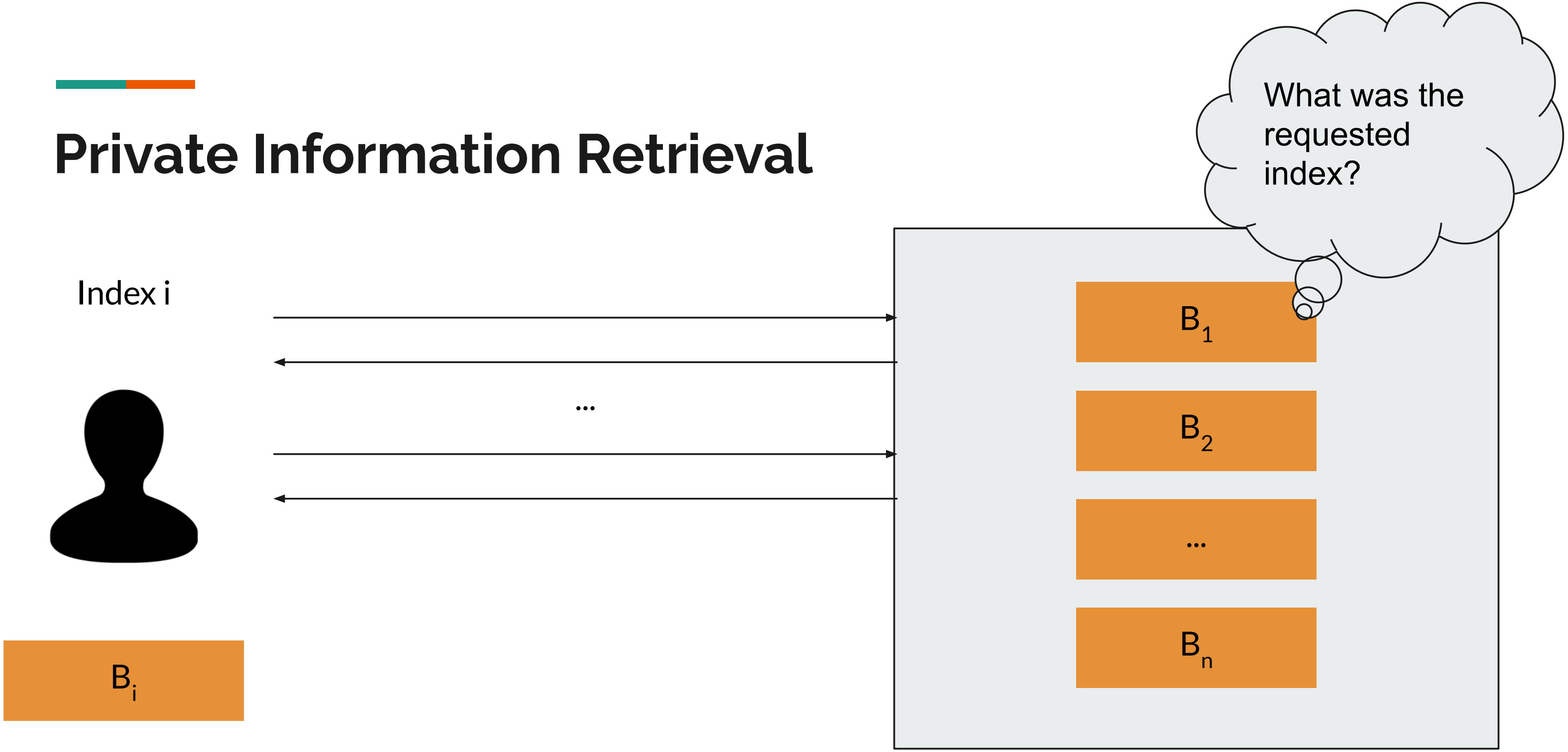
Private Information Retrieval



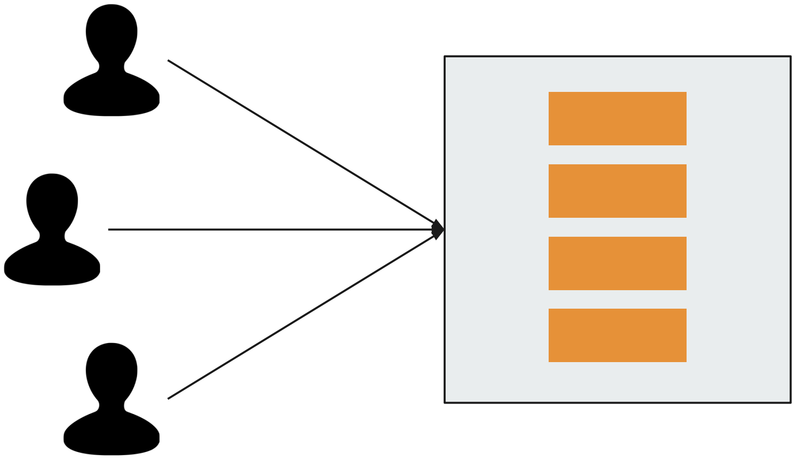
Private Information Retrieval

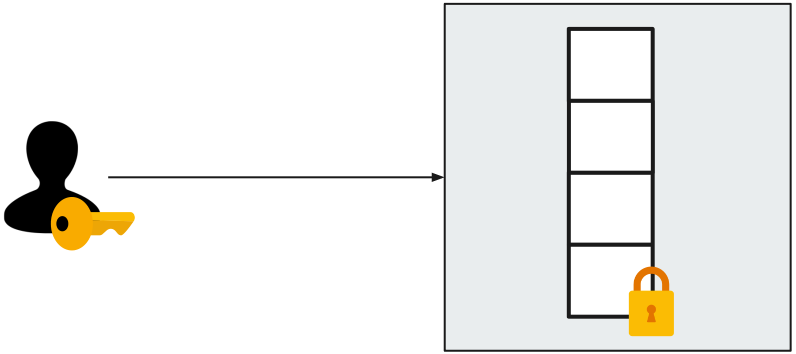


Private Information Retrieval

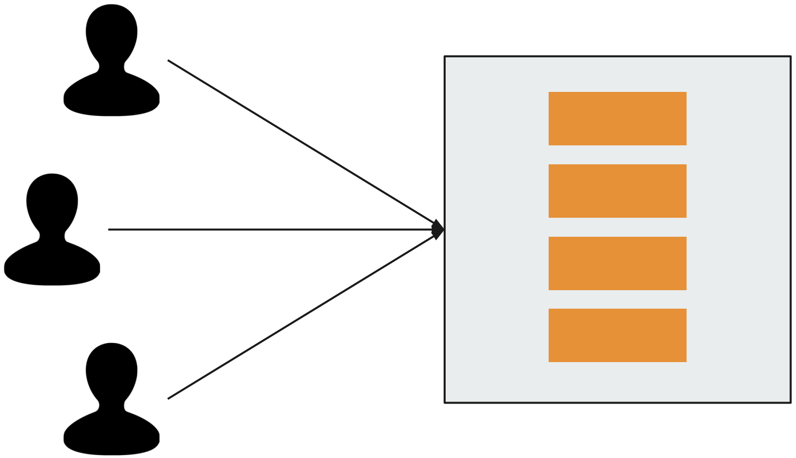


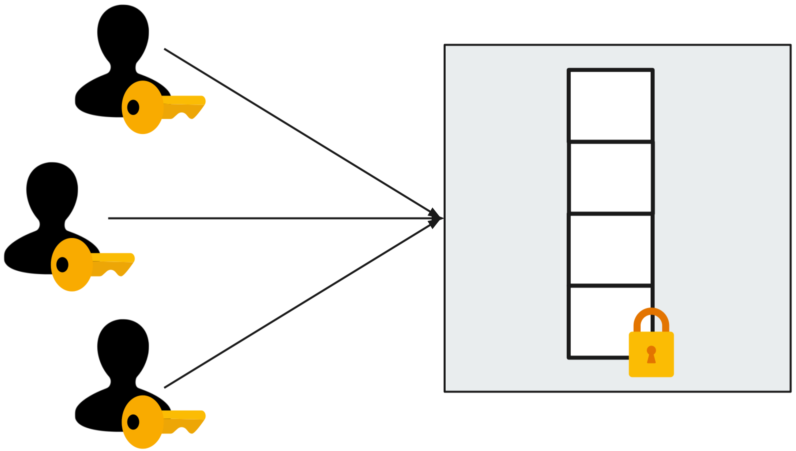
Comparison between PIR and ORAM





Comparison between PIR and ORAM





Linear Server Time Lower Bound

Thm. [Beimel, Ishai and Malkin '00]. In the standard PIR model, the server computation must be linear.

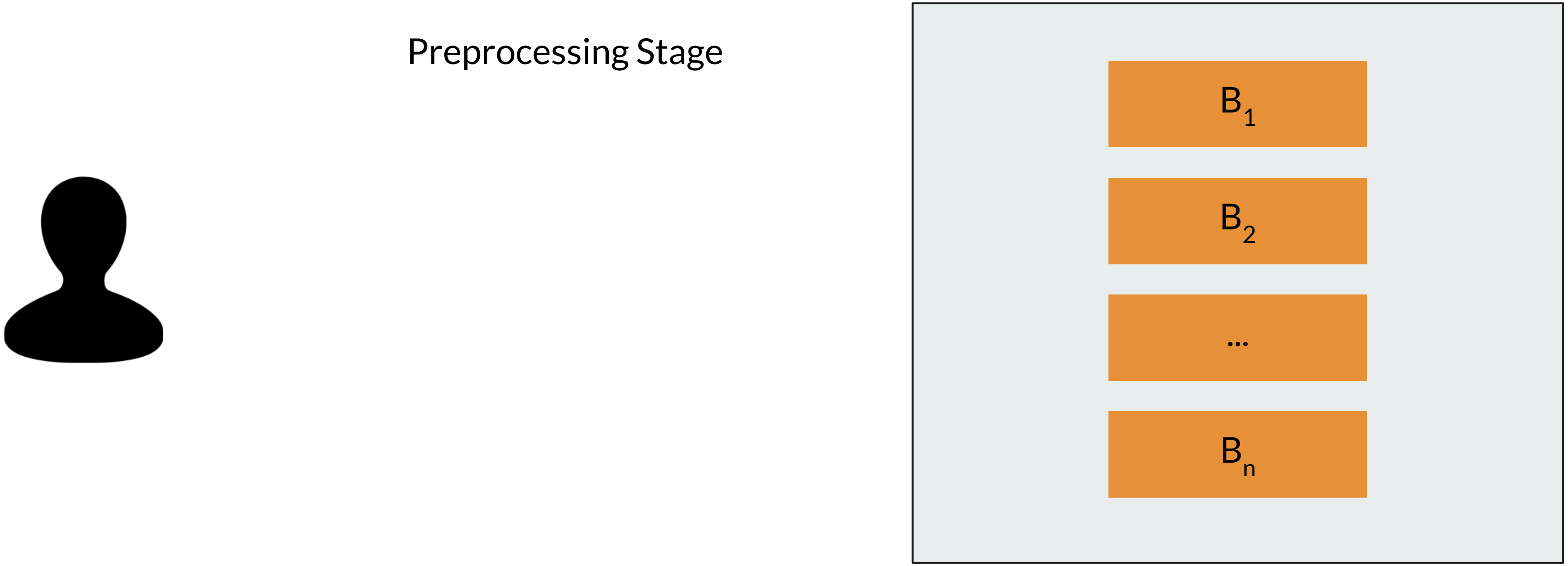
Linear Server Time Lower Bound

Thm. [Beimel, Ishai and Malkin '00]. In the standard PIR model, the server computation must be linear.

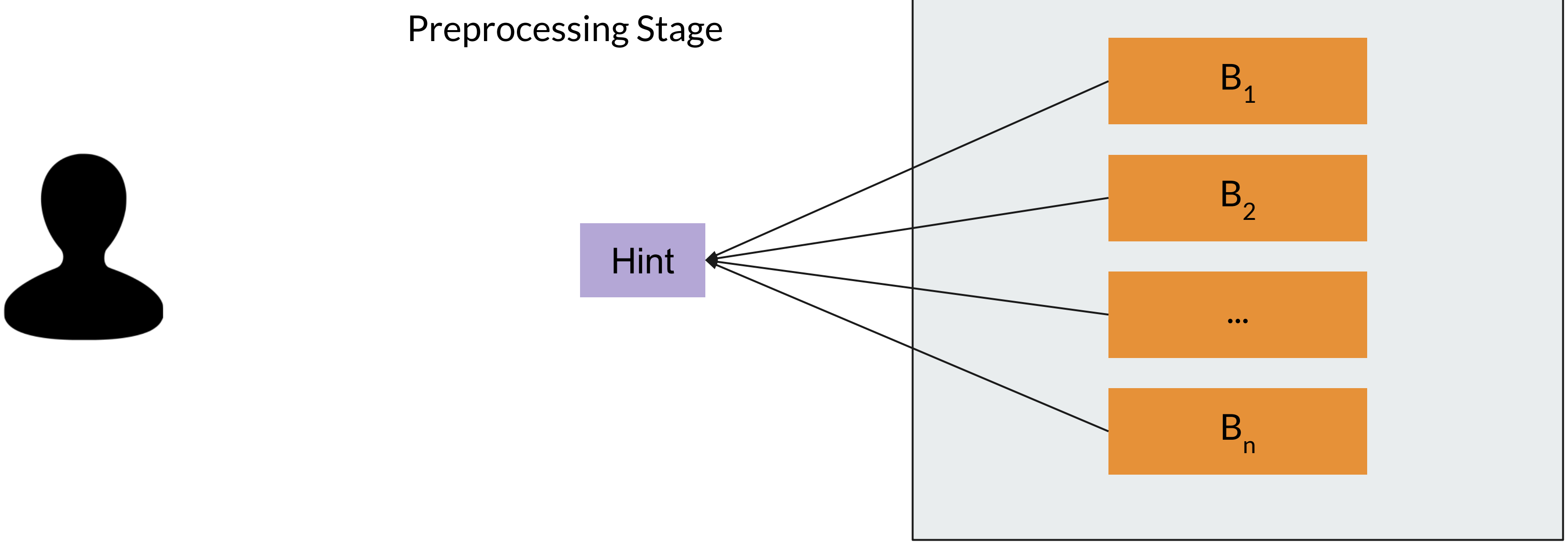
Two Ways to Circumvent Linear Lower Bound:

1. PIR with Preprocessing
2. Batch PIR

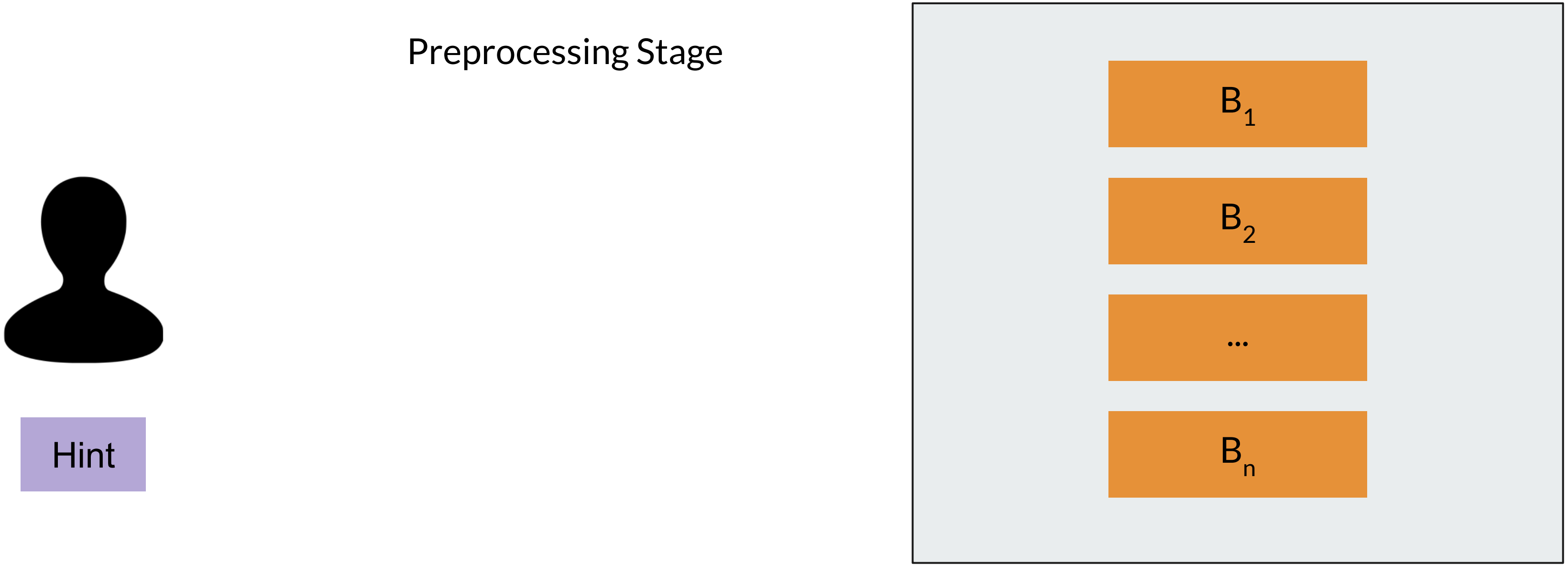
PIR with Private Preprocessing



PIR with Private Preprocessing



PIR with Private Preprocessing



PIR with Private Preprocessing

Query Stage

Index i

Hint

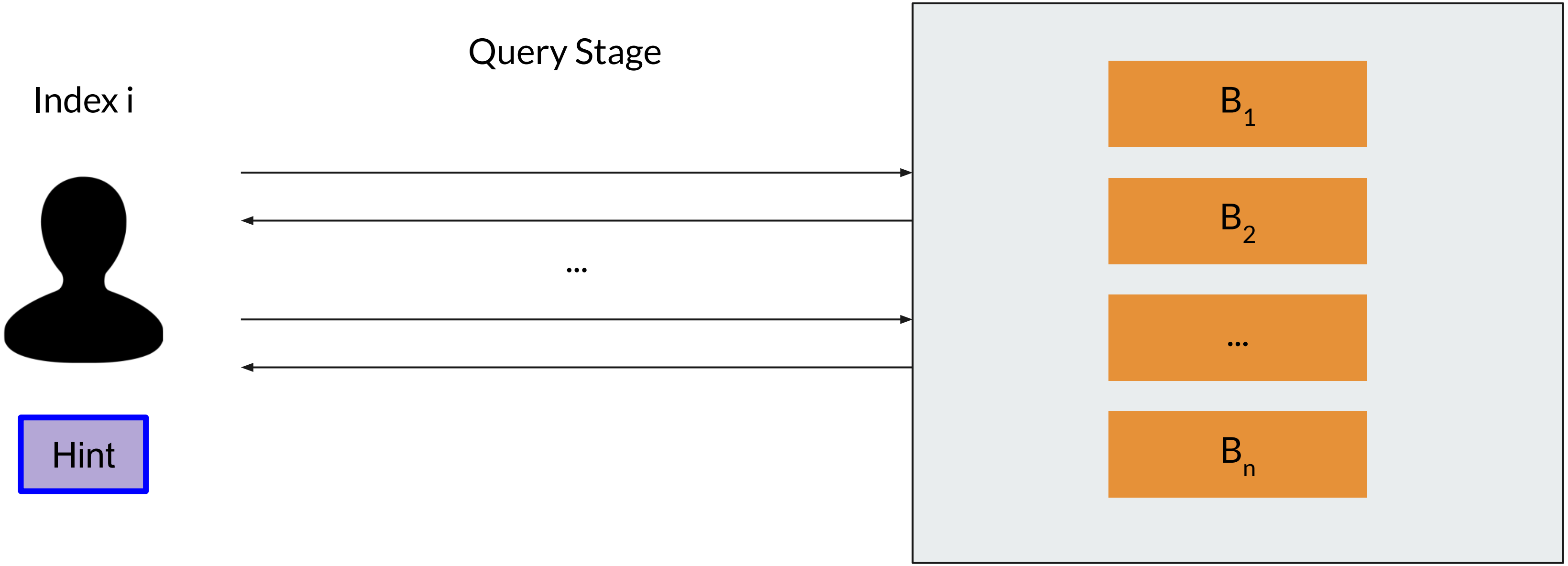
B_1

B_2

...

B_n

PIR with Private Preprocessing



Complexity Measures

Hint Size: The size of the **r**-bit hint stored by client.

Computational Time: The number of entries, **t**, probed during queries.

PIR with Private Preprocessing

- Doubly-Efficient PIR
  - [Boyle, Ishai, Pass, Wootters ‘17], [Canetti, Holmgren, Richelson ‘17]
- Private Stateful Information Retrieval
  - [Patel, Persiano, Y ‘18]
- Offline-Online PIR
  - [Corrigan-Gibbs, Kogan ‘20], [Shi, Aqeel, Chandrasekaran, Maggs ‘21], [Corrigan-Gibbs, Henzinger, Kogan ‘22] and many more…

PIR with Private Preprocessing

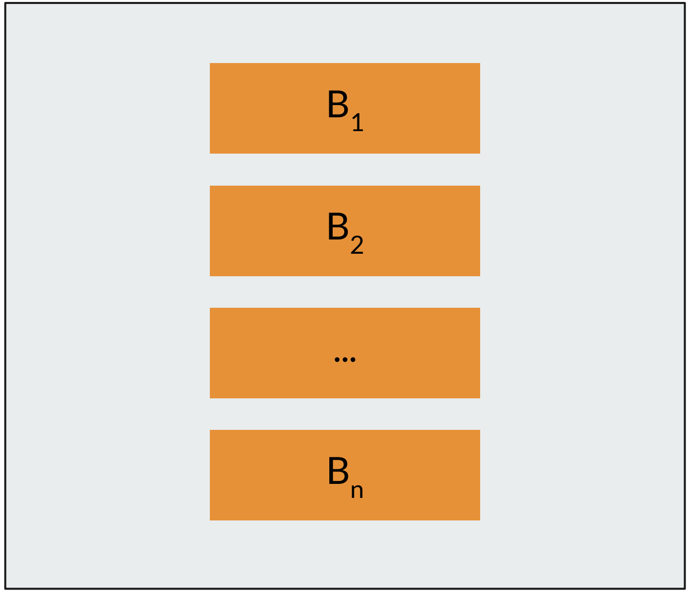
Thm. [Corrigan-Gibbs and Kogan '20]. There exists a construction with $t * r = O(n)$.

Sublinear Server Time: $t + r = O(n^{1/2})$

Thm. [Corrigan-Gibbs and Kogan '20]. There exists a construction with $t * r = \boldsymbol{\Omega}(n/\text{polylog}(n))$.

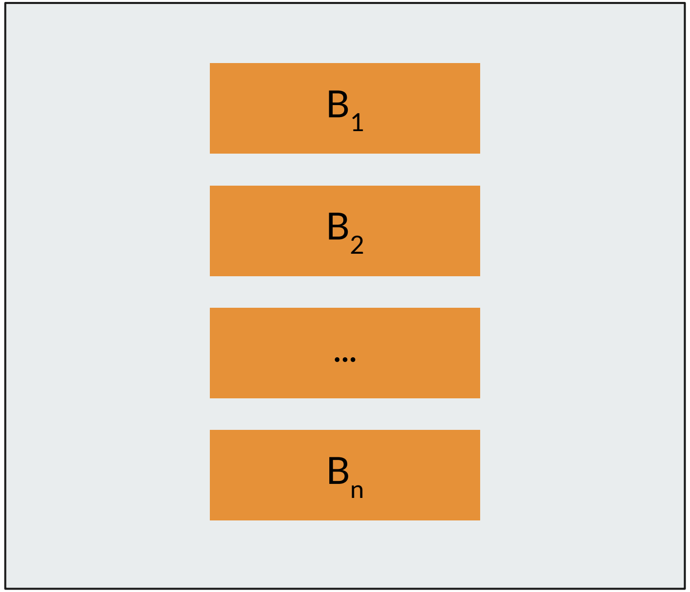
Batch PIR

Index i



Batch PIR

Set of Indices
$i_1, i_2, \ldots, i_k$



Batch PIR

Batch PIR

Thm. [Ishai, Kushilevitz, Ostrovsky and Sahai '04]. There exists a construction for k-query batch PIR requiring t = O(n polylog(n)).

Amortized Server Time per Query: ~O(n/k)

Batch PIR with Private Preprocessing

PIR with Private Preprocessing: $t * r = O(n)$

Batch PIR: $t = \tilde{O}(n/k)$

Can we combine private preprocessing and batch queries to obtain even faster server times?

Dream Batch PIR with Private Preprocessing

	Single-Query	k-Query
No Preprocessing	t = O(n)	
Private Preprocessing	t * r = O(n)	

Dream Batch PIR with Private Preprocessing

	Single-Query	k-Query
No Preprocessing	t = O(n)	t = ~O(n)
Private Preprocessing	t * r = O(n)	

Dream Batch PIR with Private Preprocessing

	Single-Query	k-Query
No Preprocessing	t = O(n)	t = ~O(n)
Private Preprocessing	t * r = O(n)	**t * r = ~O(n)???**

Our Contributions: Lower Bound

Thm. For any computationally-secure, **L**-server, **k**-query PIR with private preprocessing where $L = O(1)$, it must be

- If $r \geq k$, then $t * r = \Omega(n * k)$.
- If $r < k$, then $t = \Omega(n)$.

This holds when the PIR scheme errs with probability at most **1/15**.

Our Contributions: Lower Bound

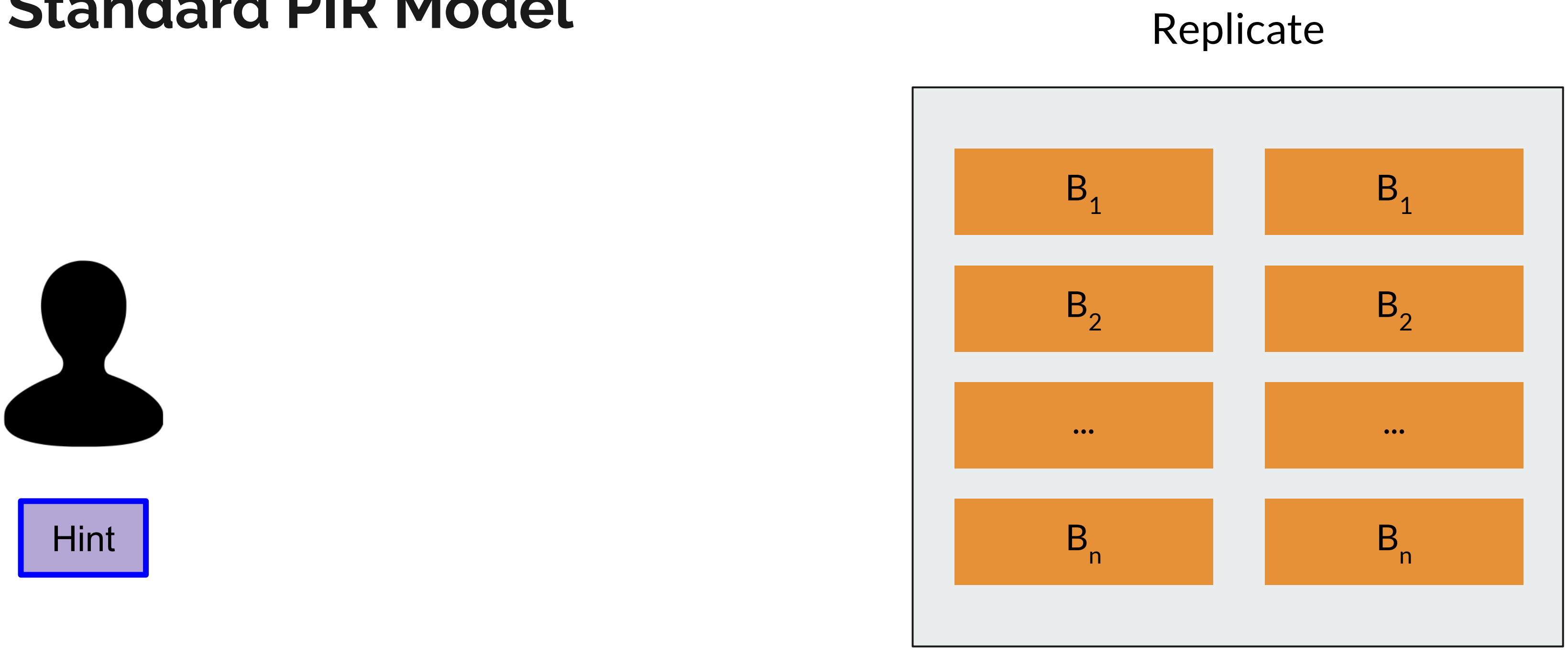
Thm. For any computationally-secure, L-server, **single-query** PIR with private preprocessing where $L = O(1)$, it must be

$$t * r = \Omega(n).$$

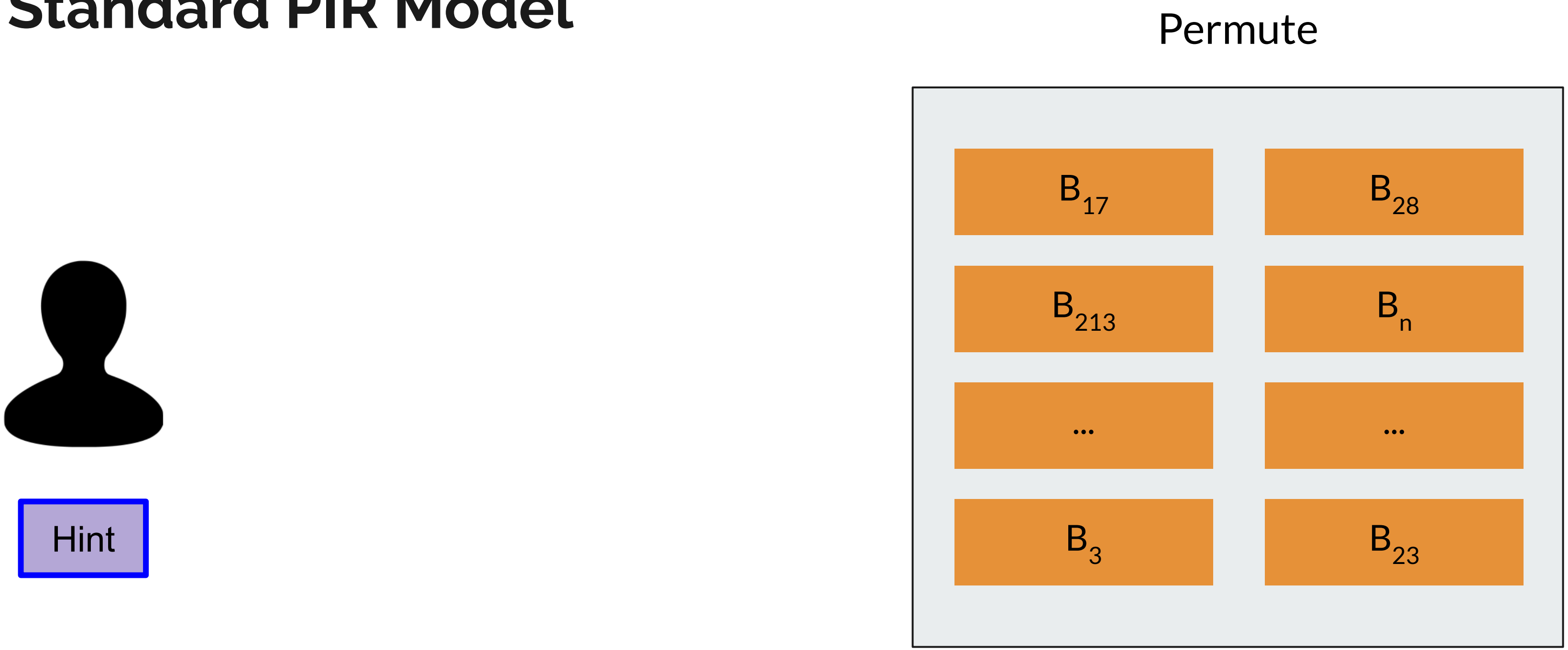
Improves upon prior single-query lower bound of $t * r = \Omega(n / \mathrm{polylog}(n))$ [Corrigan-Gibbs, Kogan ‘20]

Standard PIR Model

Standard PIR Model



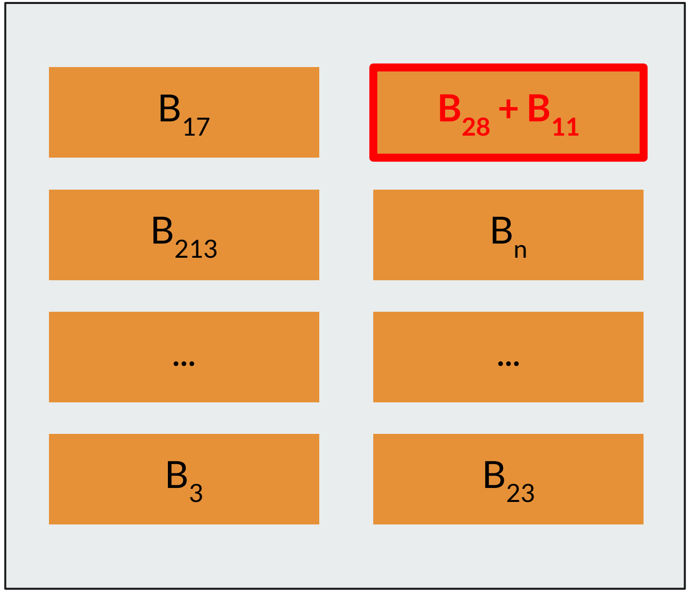
Standard PIR Model



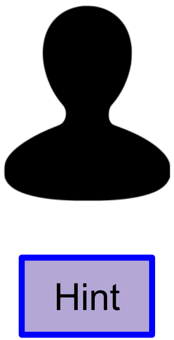
Standard PIR Model

Hint

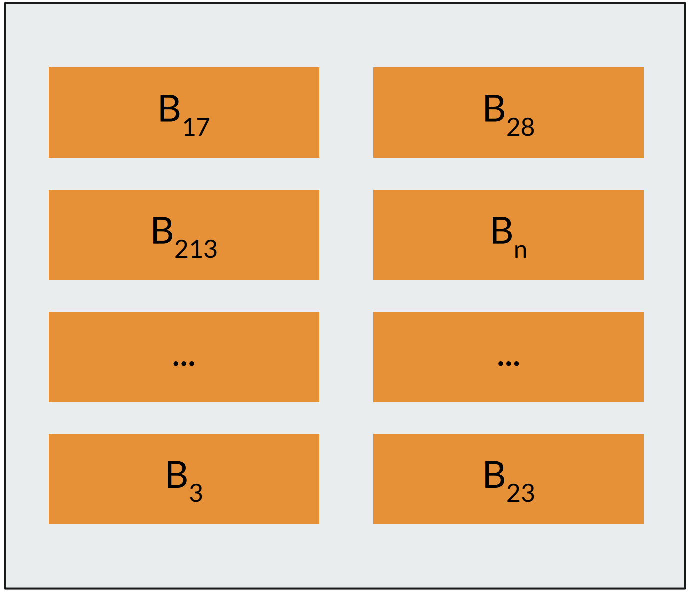
No Encoding



Standard PIR Model



Arbitrary Encoding



Standard PIR Model vs. Cell Probe Model

B_{17}	B_{28}
B_{213}	B_n
…	…
B_3	B_{23}

$f(B_{17}, B_1)$	$B_{28} + B_{11}$
B_{213} / B_1	$B_n * 13$
…	…
$B_3 - B_{19}$	$B_{23} / 100$

Our Contributions: Lower Bound Barrier

Thm. If for every single-query, 2-server PIR with private preprocessing, it is holds that $t * r = \mathbf{\Omega(n)}$ in the **cell probe model**, then the online matrix-vector (OMV) conjecture is true.

Barrier: OMV is a core conjecture in fine-grained complexity.

Our Contributions: Upper Bound

Thm. Given any single-query PIR with private preprocessing with **$t * r = f(n)$**, there exists a k-query PIR with private preprocessing satisfying

$$t * r = \tilde{O}(k * f(n)).$$

Prior reductions required either multiple rounds or certain assumptions on single-query scheme.

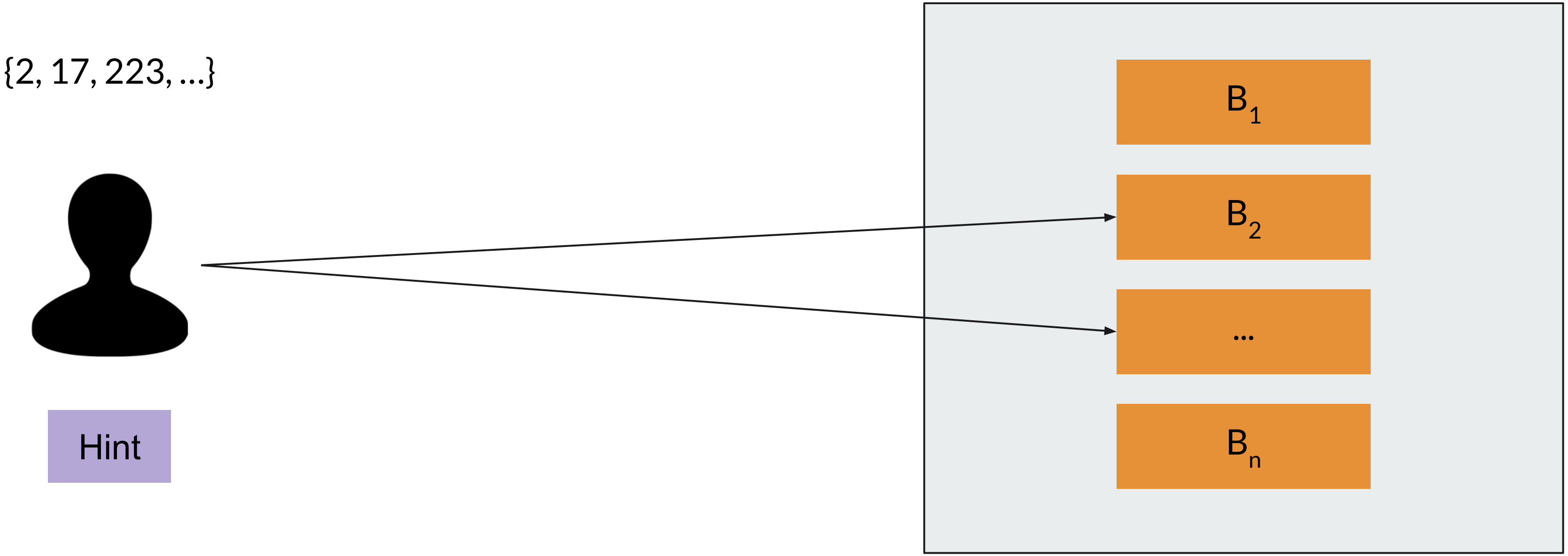
Lower Bound Proof Techniques

1. Relationship between Queried and Probed Entries
2. Discovering Good Batch Queries
3. Impossible Encoding of Database

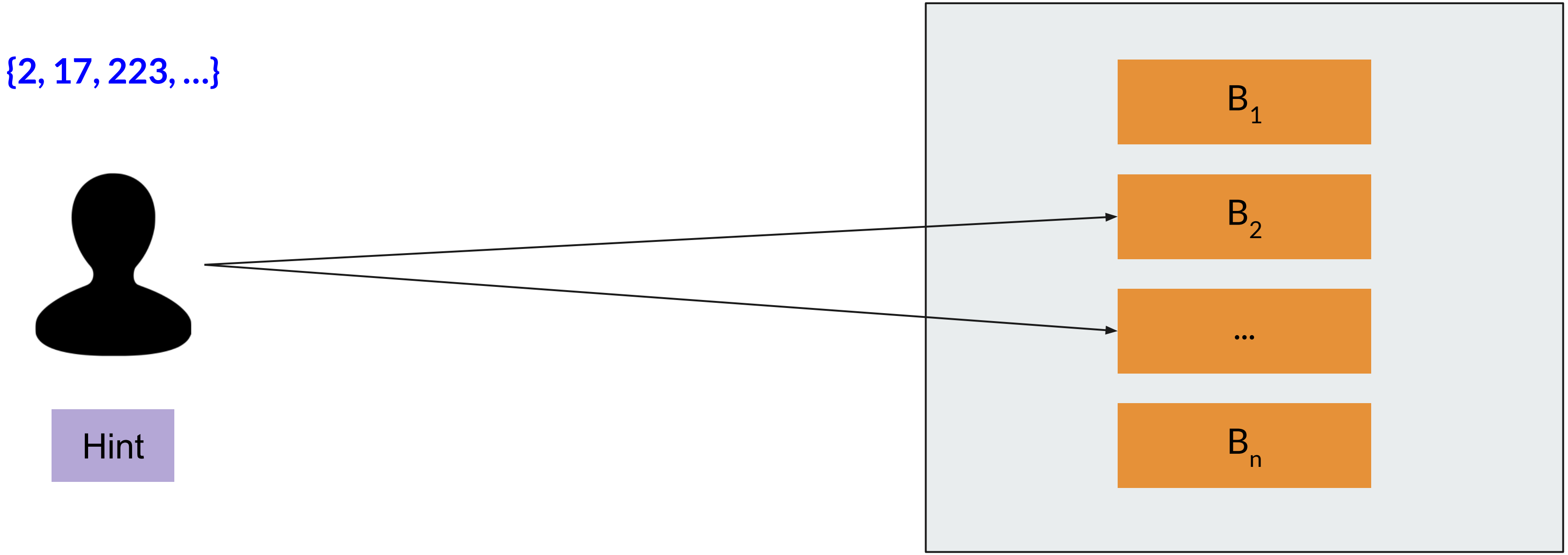
Lower Bound Proof Techniques

1. **Relationship between Queried and Probed Entries**
2. **Discovering Good Batch Queries**
3. Impossible Encoding of Database

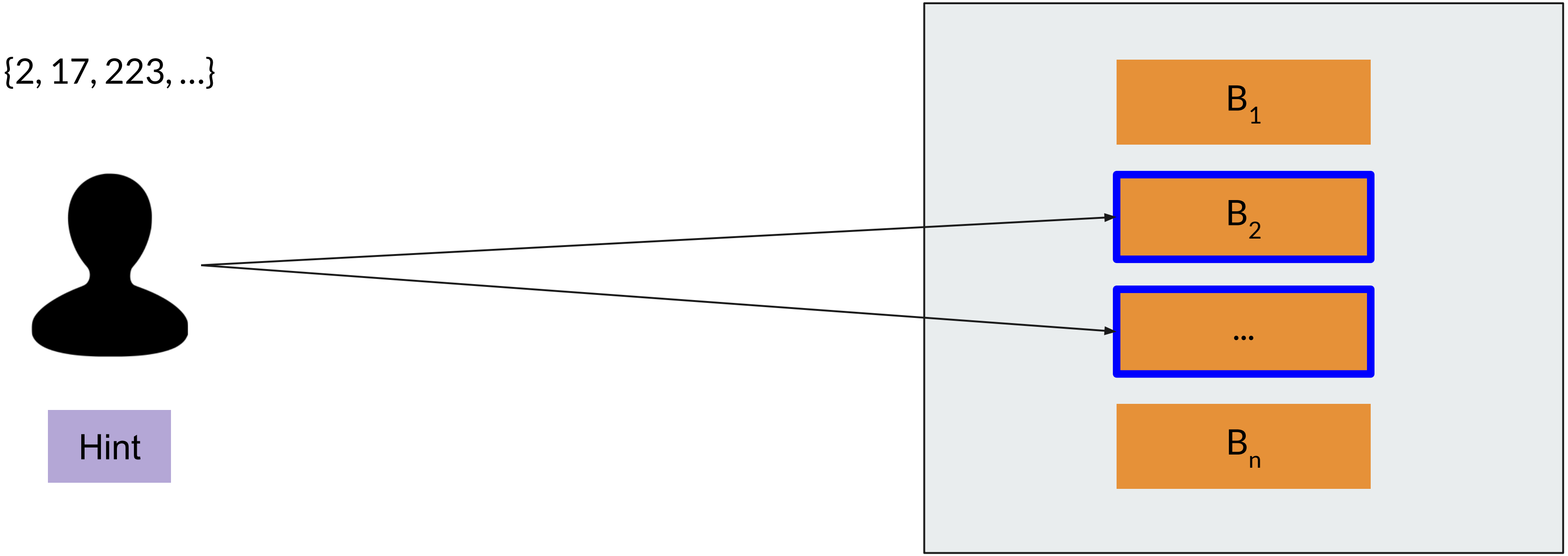
Relationship between Queried and Probed Entries



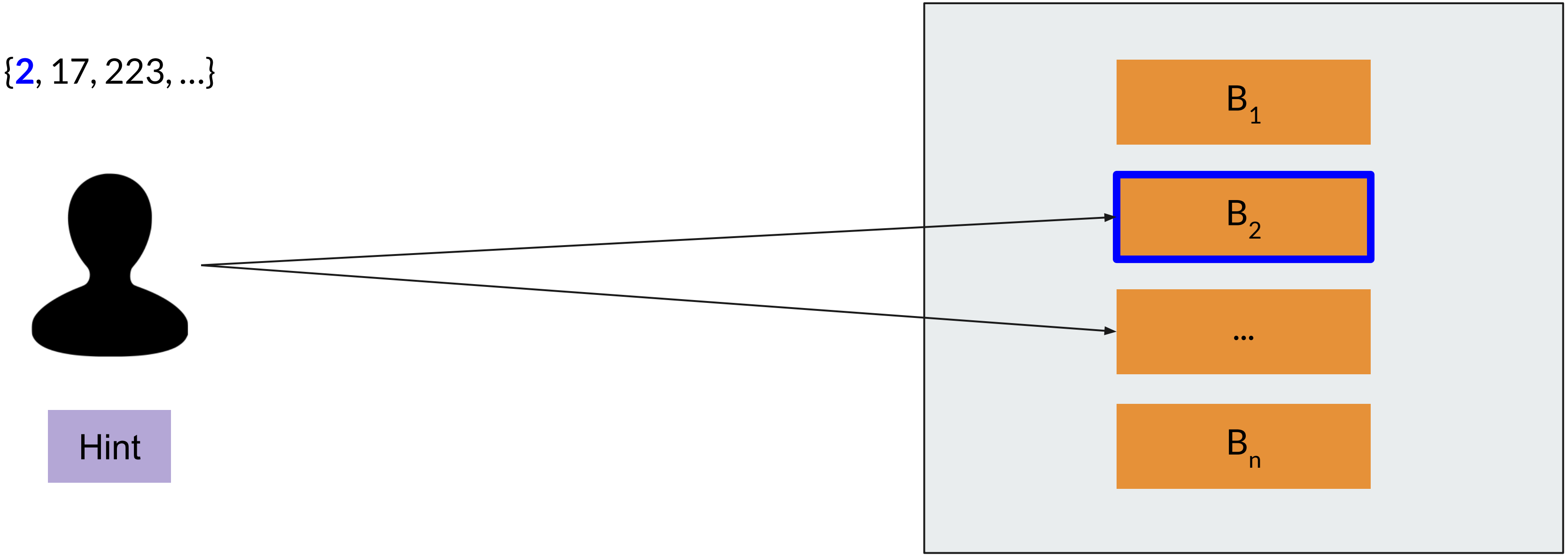
Relationship between Queried and Probed Entries



Relationship between Queried and Probed Entries



Relationship between Queried and Probed Entries

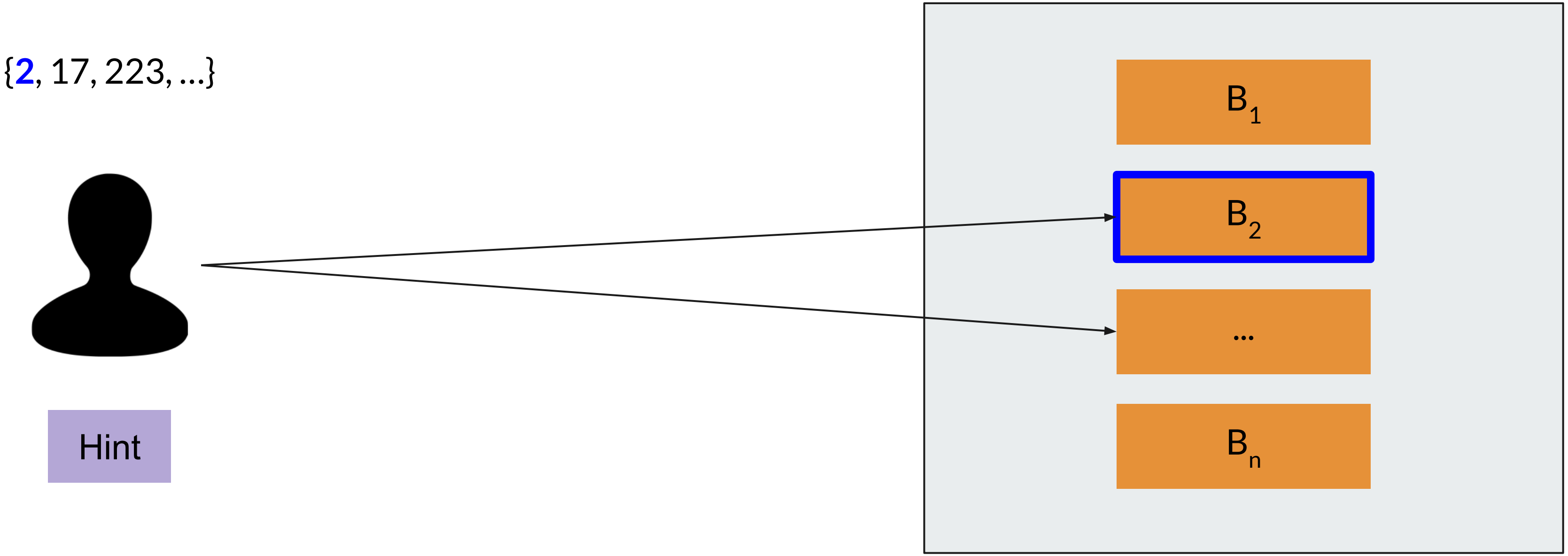


Relationship between Queried and Probed Entries

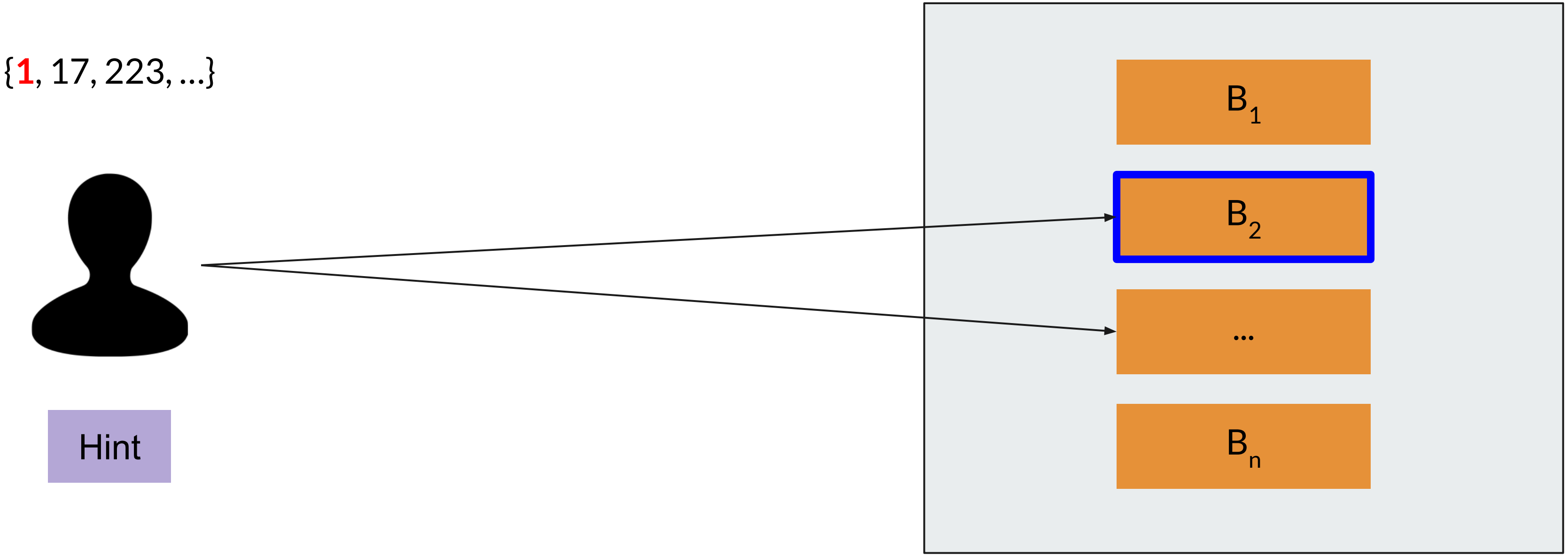
Assumption. Suppose at most half ($n/2$) entries are probed.

Question: If index i is queried, what is probability the i-th entry is probed?

Relationship between Queried and Probed Entries



Relationship between Queried and Probed Entries



Relationship between Queried and Probed Entries

Assumption. Suppose at most half (n/2) entries are probed.

Question: If index i is queried, what is probability the i-th entry is probed?

$$\Pr[\text{index } i \text{ is probed} \mid \text{index } i \text{ is queried}] \approx \Pr[\text{index } i \text{ is probed} \mid \text{index } i \text{ is not queried}]$$

Relationship between Queried and Probed Entries

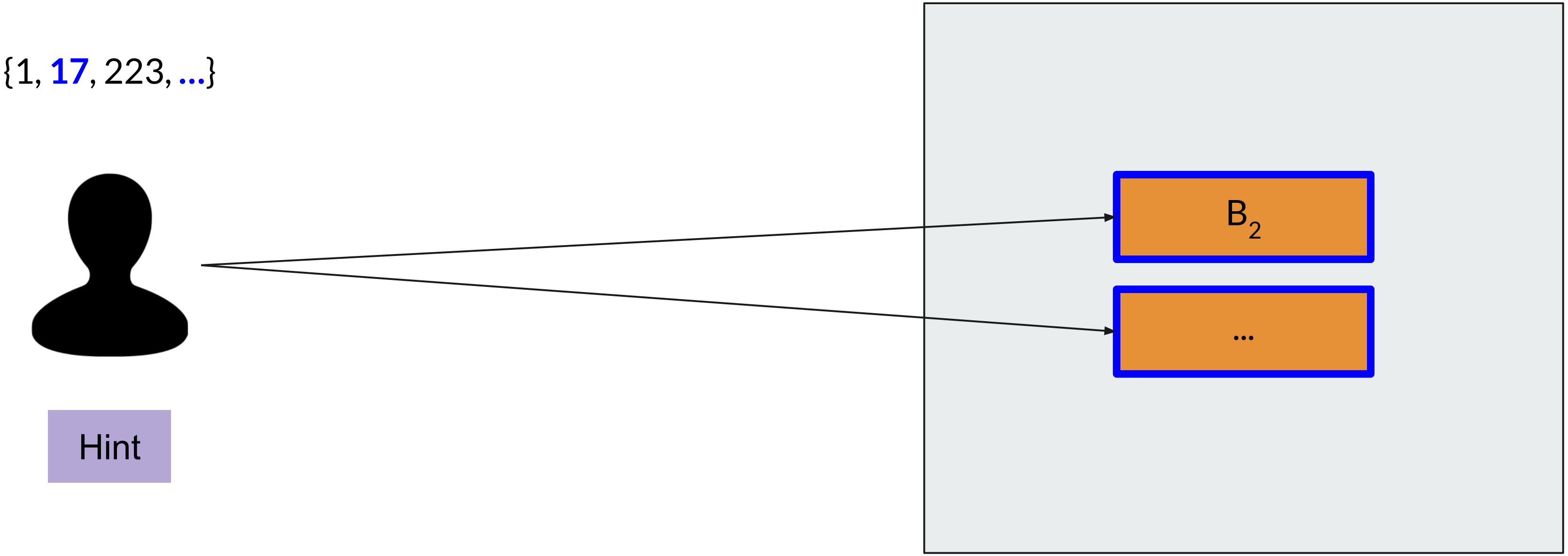
Assumption. Suppose at most **half (n/2) entries are probed**.

Question: If index i is queried, what is probability the i-th entry is probed?

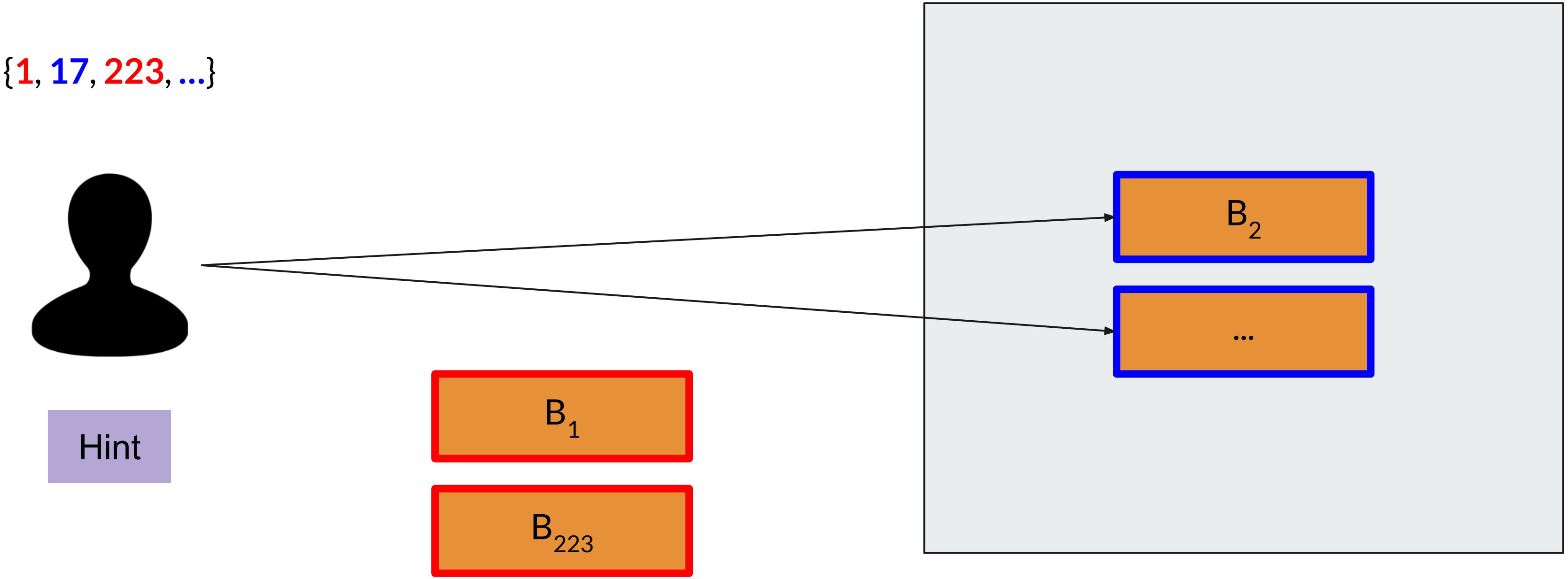
Pr[index i is probed | index i is queried] ≈ Pr[index i is probed | index i is **not** queried]

E[# of queried indices that are probed] ≤ k/2

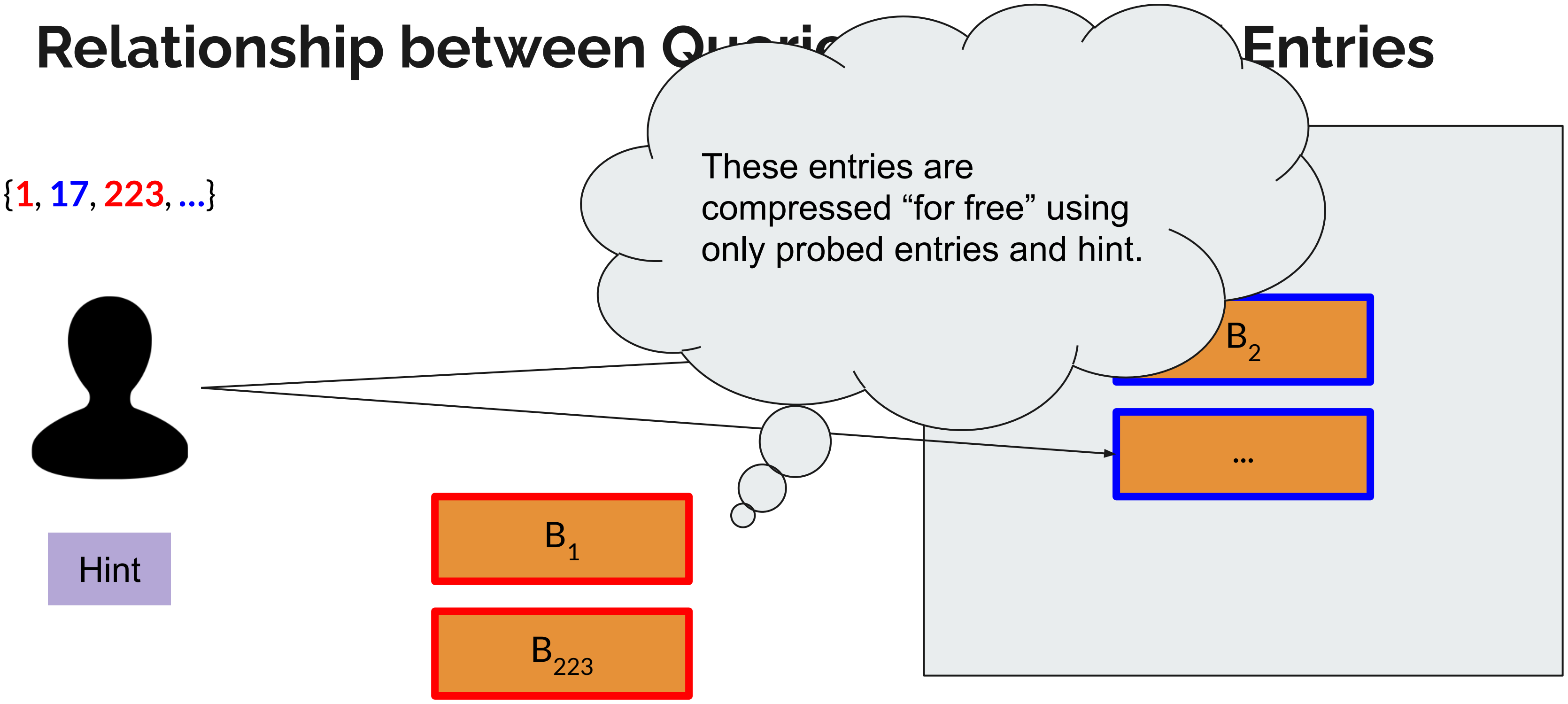
Relationship between Queried and Probed Entries



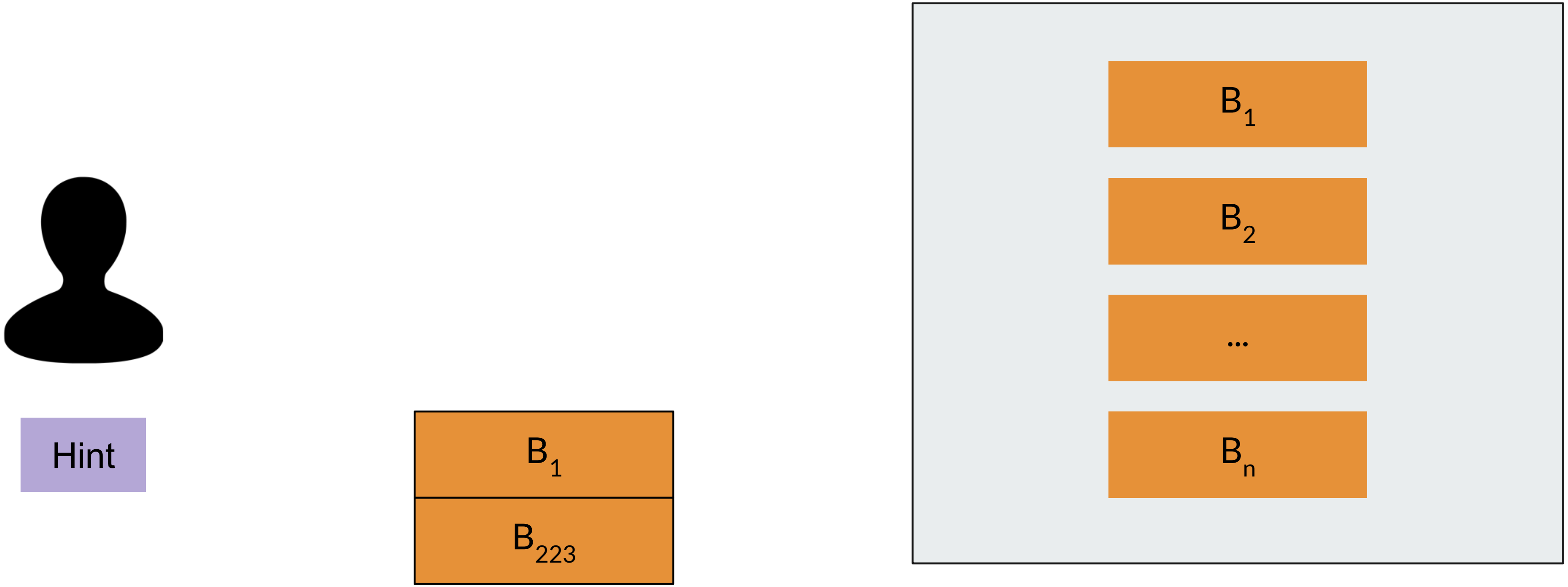
Relationship between Queried and Probed Entries



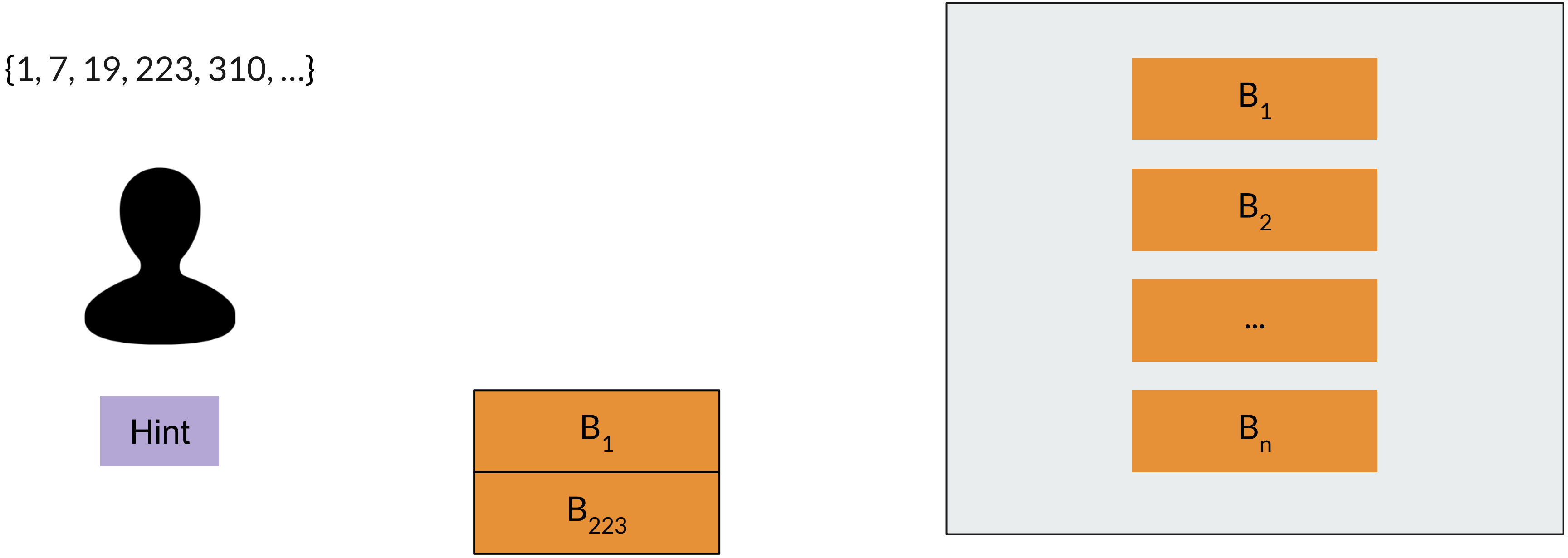
Relationship between Queri... Entries



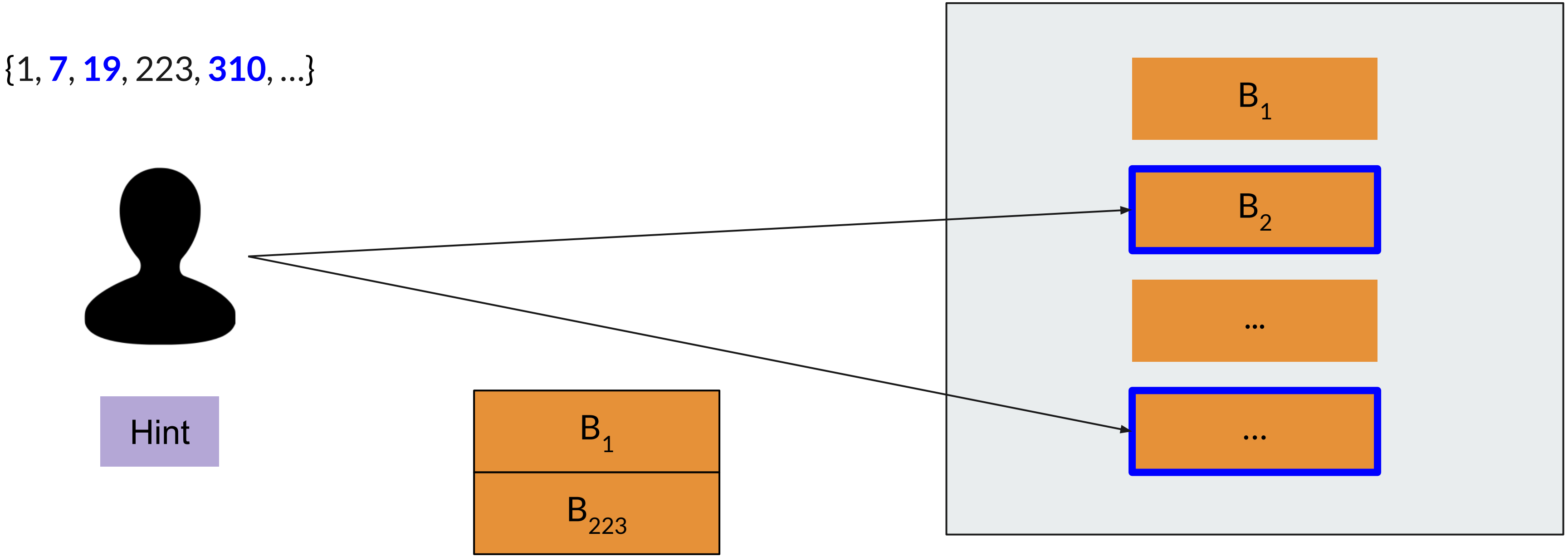
Discovering Good Batch Queries



Discovering Good Batch Queries

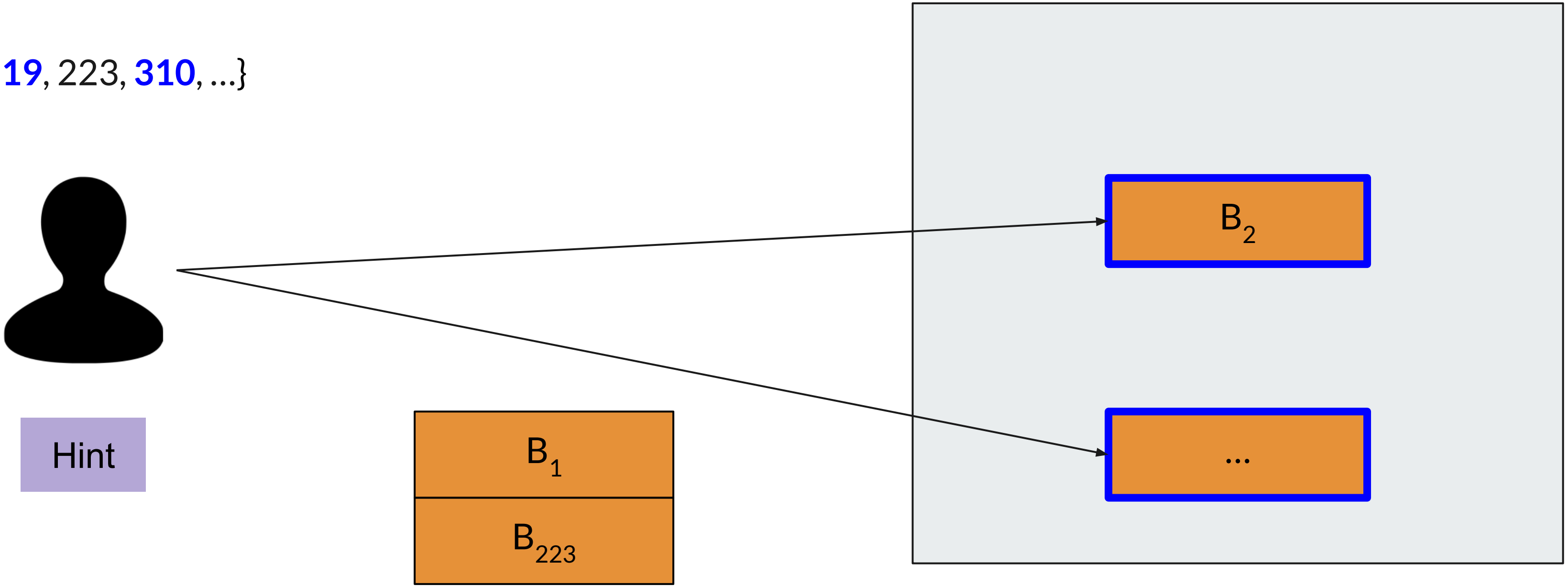


Discovering Good Batch Queries



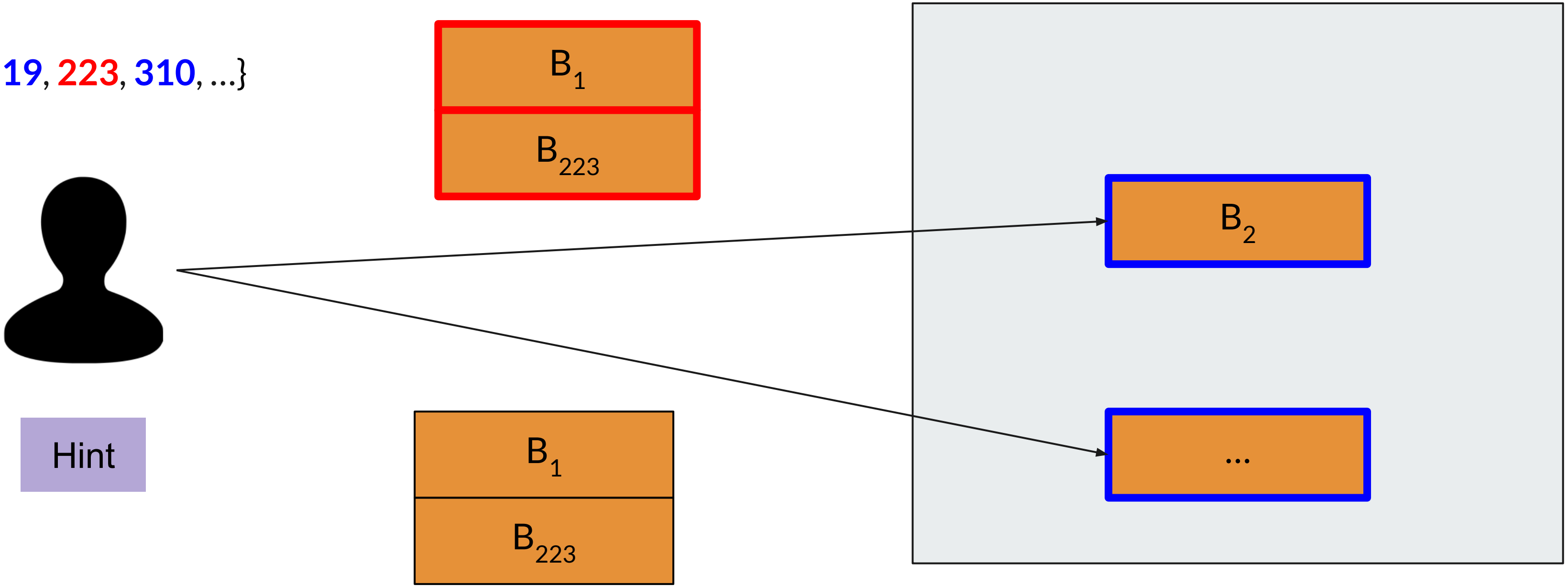
Discovering Good Batch Queries

{1, 7, 19, 223, 310, ...}



Discovering Good Batch Queries

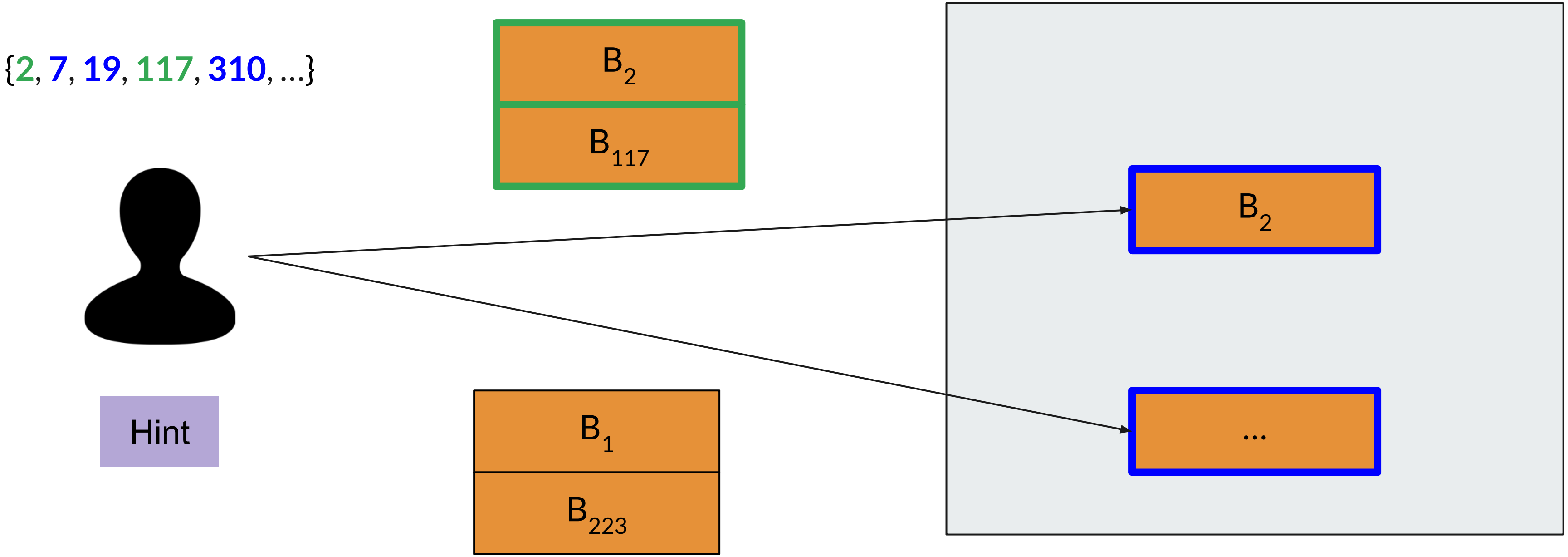
{1, 7, 19, 223, 310, ...}



Discovering Good Batch Queries

Goal: Find sequence of batch queries such that “free” entries are minimally overlapping.

Discovering Good Batch Queries



Discovering Good Batch Queries

Goal: Find sequence of batch queries such that “free” entries are minimally overlapping.

Lemma: Random batch queries satisfy this with high probability.

Lower Bound Proof Techniques

1. Relationship between Queried and Probed Entries
2. Discovering Good Batch Queries
3. **Impossible Encoding of Database**

Questions?

Email: kwlyeo@cs.columbia.edu