Observing and Preventing Traffic Analysis Leaks

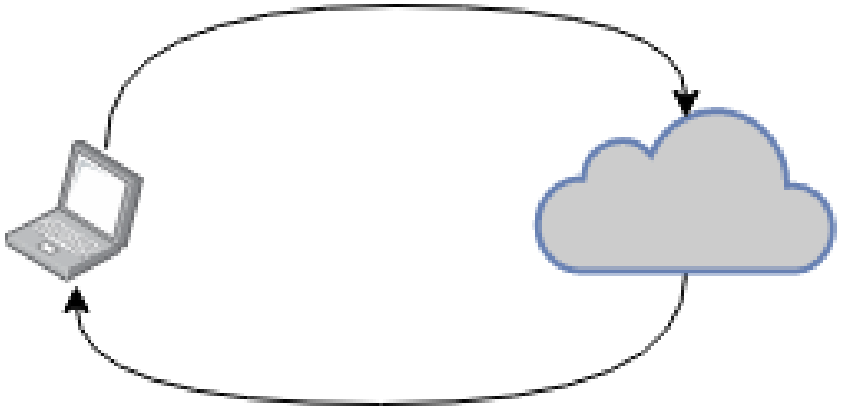
Ed Younis

CS 261

Oct 1st, 2018

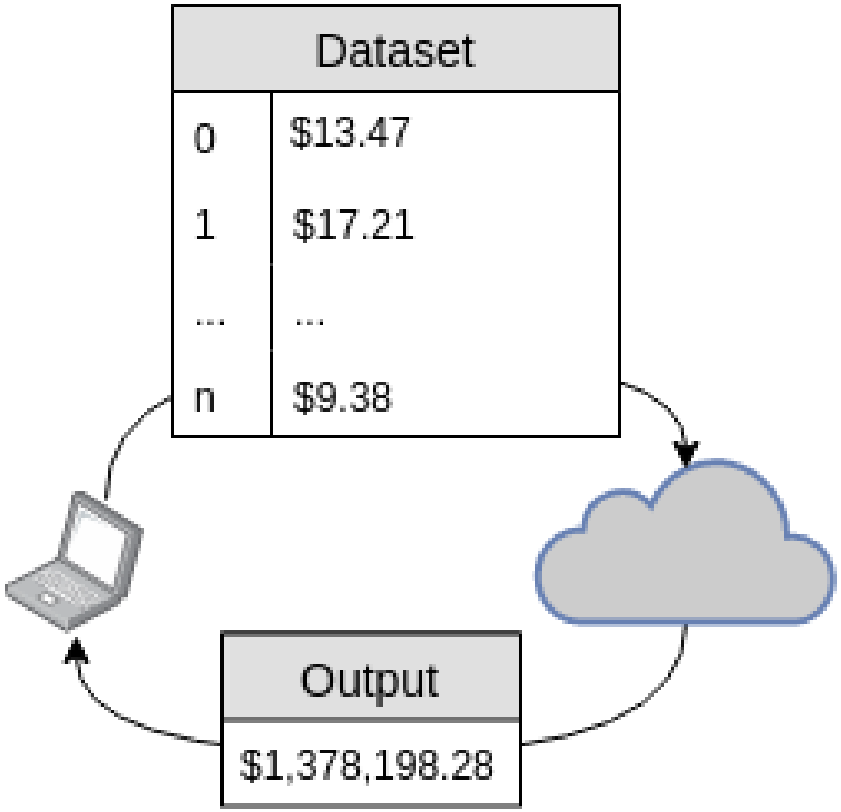
System

- Public Cloud Infrastructure
- User uploads data and code
- User receives results



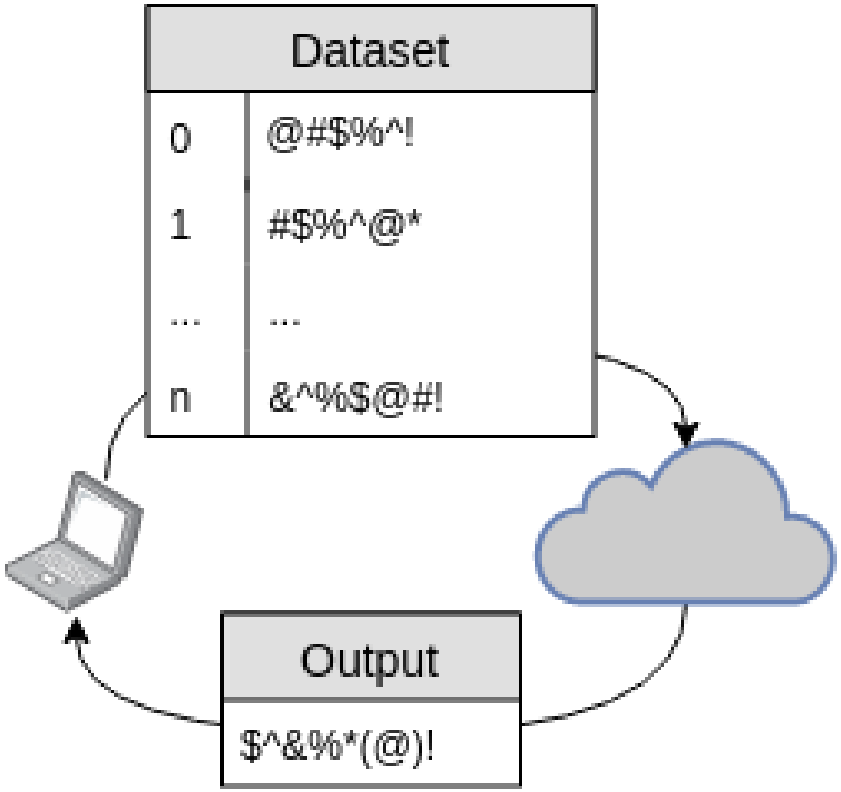
System

- Public Cloud Infrastructure
- User uploads data and code
- User receives results



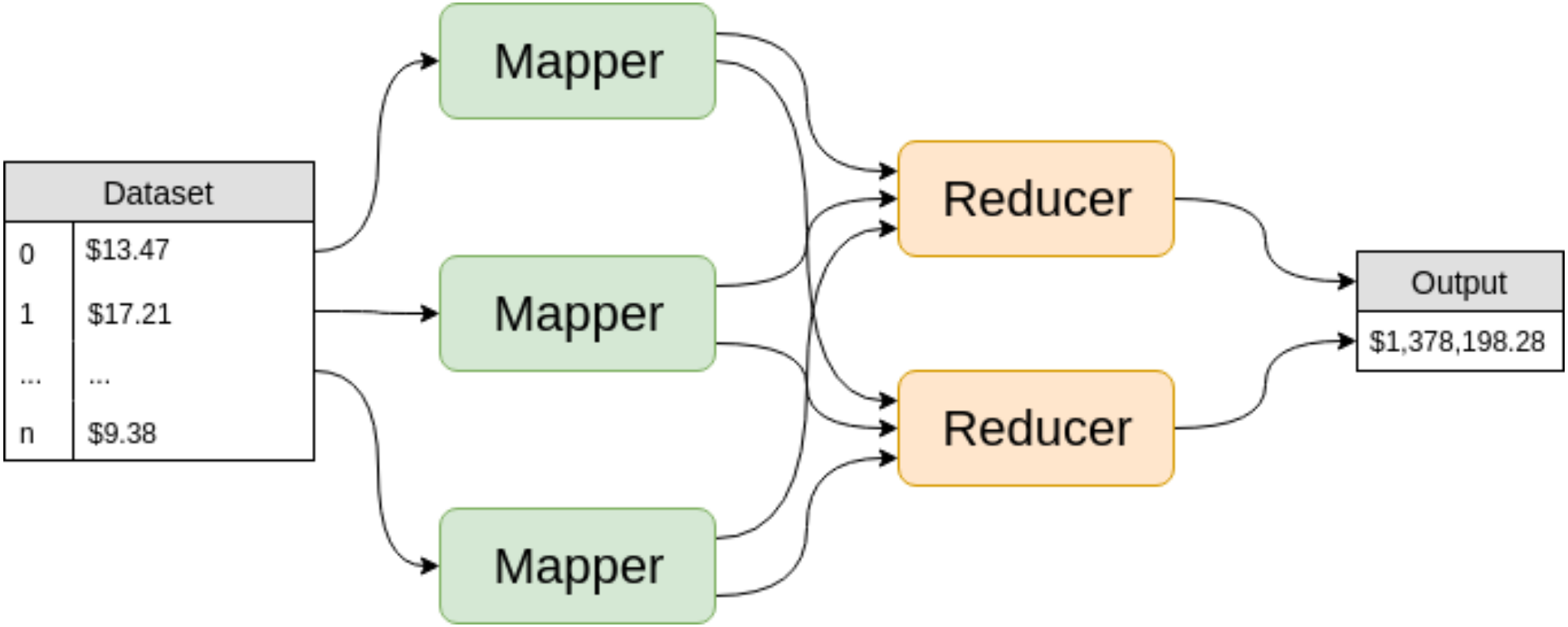
System

- Public Cloud Infrastructure
- User uploads data and code
- User receives results



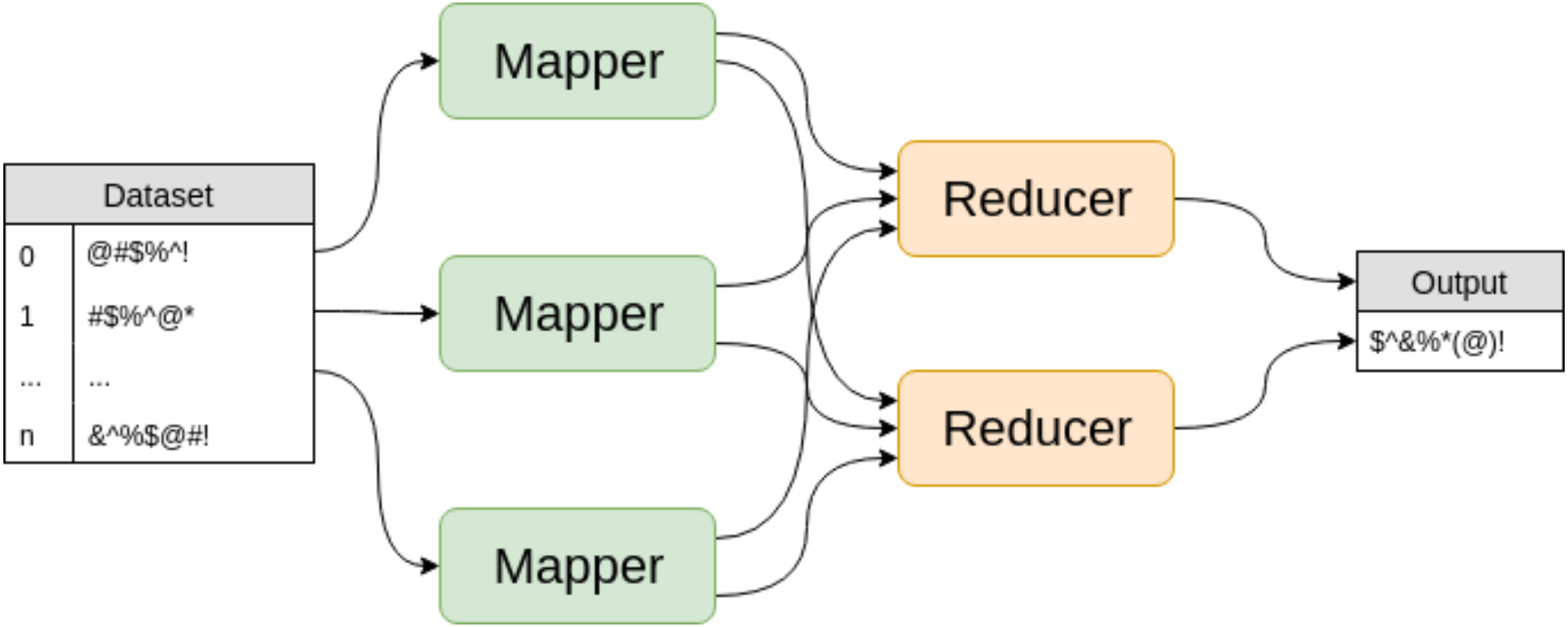
System

- Map Reduce Framework
- Data is distributed to the mappers
- Selected data is then directed to the reducers
- Reduces finally perform computation



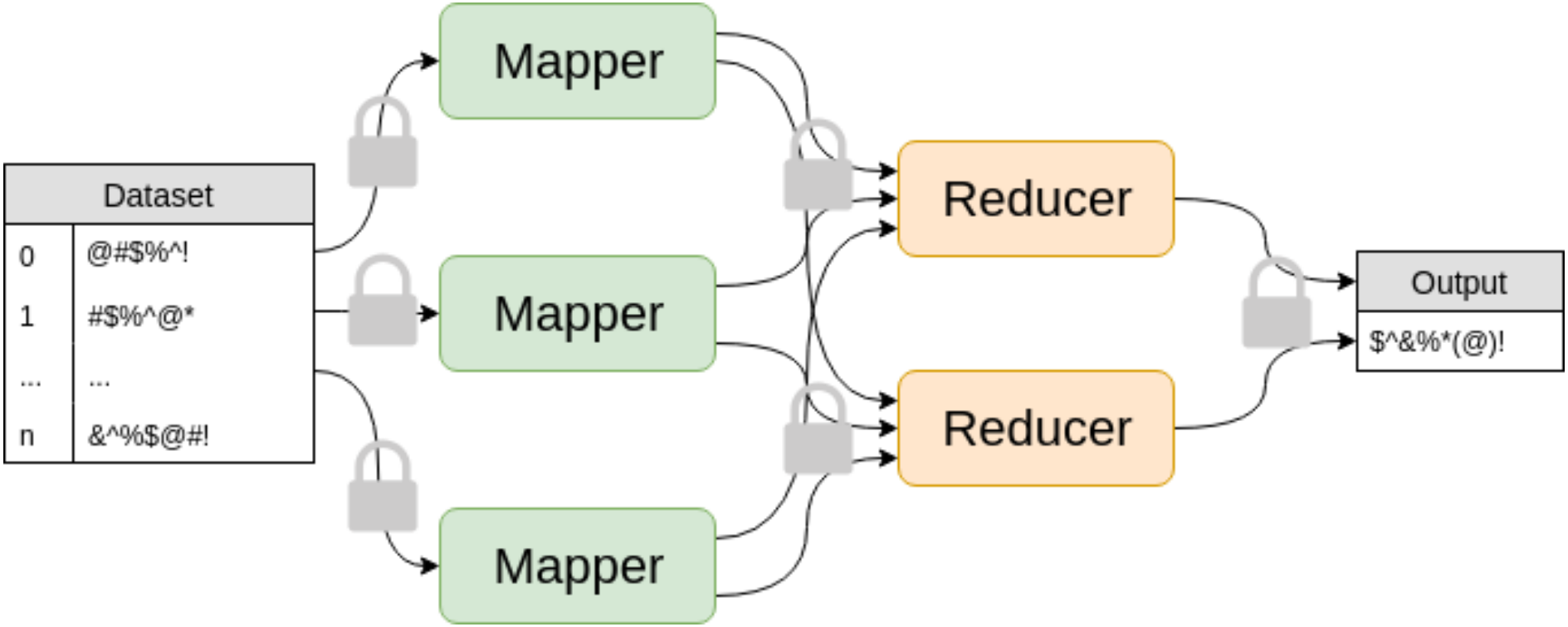
System

- Map Reduce Framework
- Data is distributed to the mappers
- Selected data is then directed to the reducers
- Reduces finally perform computation



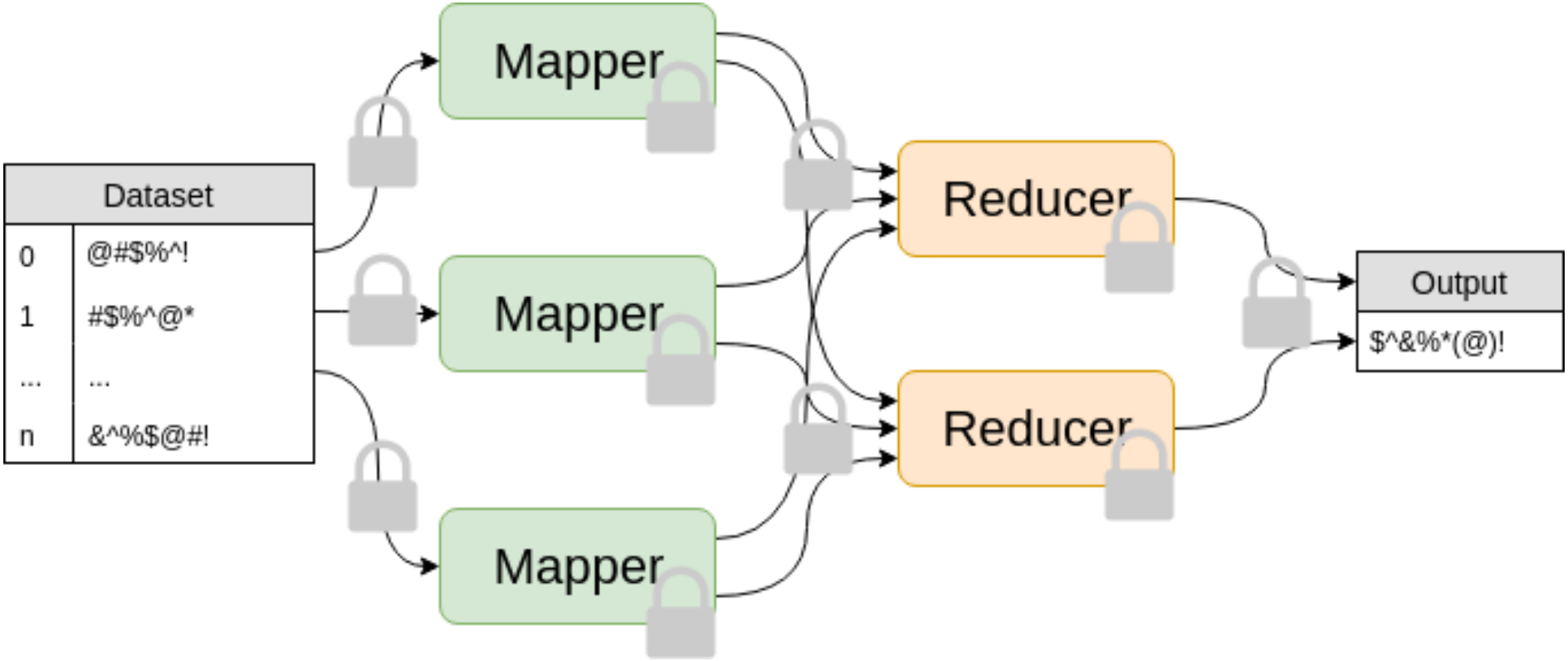
System

- Map Reduce Framework
- Data is distributed to the mappers
- Selected data is then directed to the reducers
- Reduces finally perform computation



System

- Map Reduce Framework
- Data is distributed to the mappers
- Selected data is then directed to the reducers
- Reduces finally perform computation



Assumptions

- Encrypted data has confidentiality and integrity.
- Hardware is secure.

Assumptions

- Encrypted data has confidentiality and integrity.
- Hardware is secure.
- A **passive attacker** can observe communication between every node.
- A **passive attacker** will have background knowledge about a job, its input, and its output. We will assume that the attacker can infer the specific job and some statistics about the data.

Assumptions

- Encrypted data has confidentiality and integrity.
- Hardware is secure.
- A **passive attacker** can observe communication between every node.
- A **passive attacker** will have background knowledge about a job, its input, and its output. We will assume that the attacker can infer the specific job and some statistics about the data.
- An **active attacker** additionally can control resources and scheduling.

Assumptions

Realistic?

We will assume that the attacker can infer the specific job and some statistics about the data.

What do you think?

Assumptions

Realistic?

We will assume that the attacker can infer the specific job and some statistics about the data.

What do you think?

Lets play a game!


WHAT'S THAT
?
Dataset
imgflip.com

U.S. 1990 Census Sample

- 900 MB
- 68 Attributes
- 2.5 million records

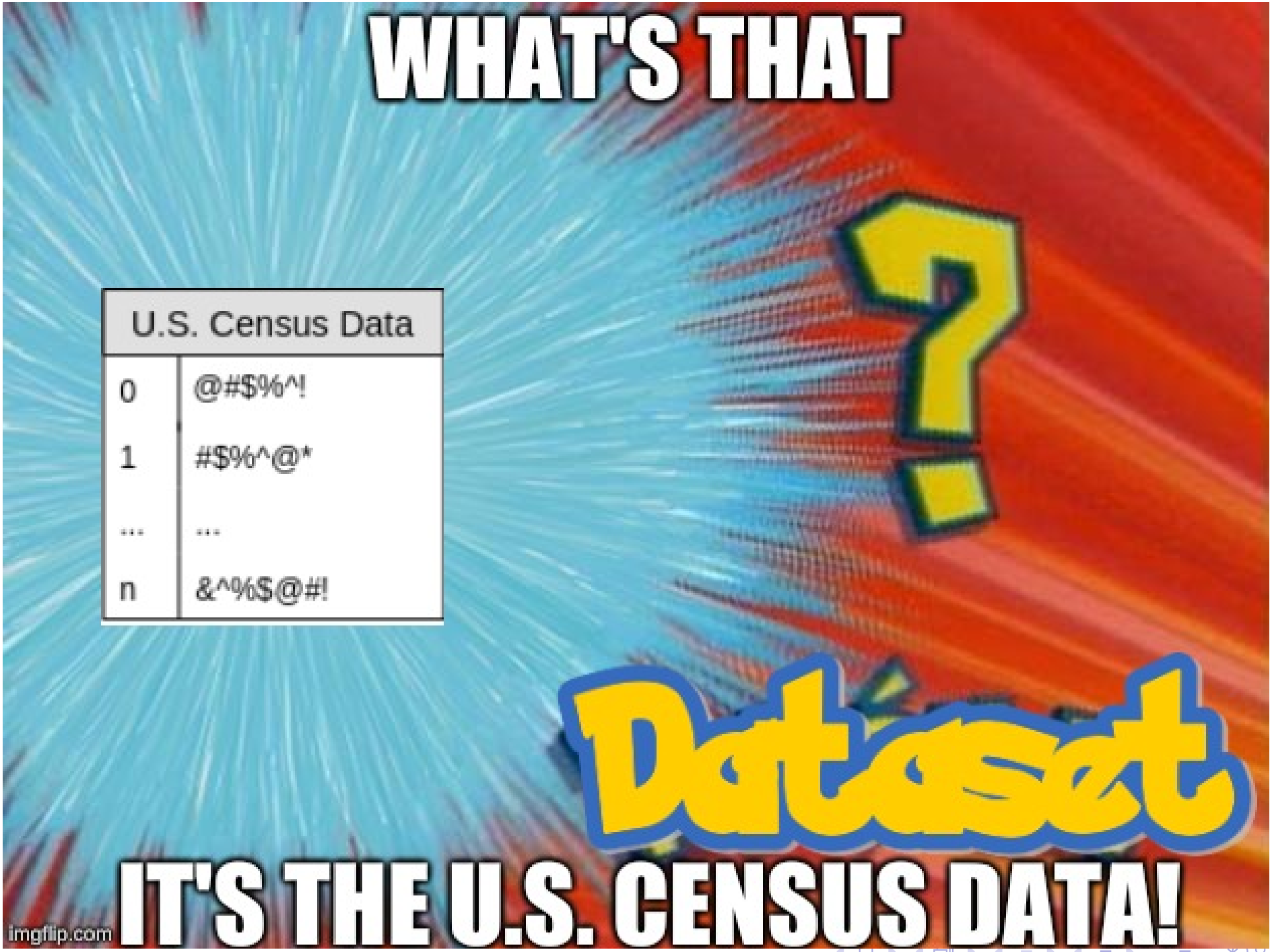
New York 2013 Taxi Rides

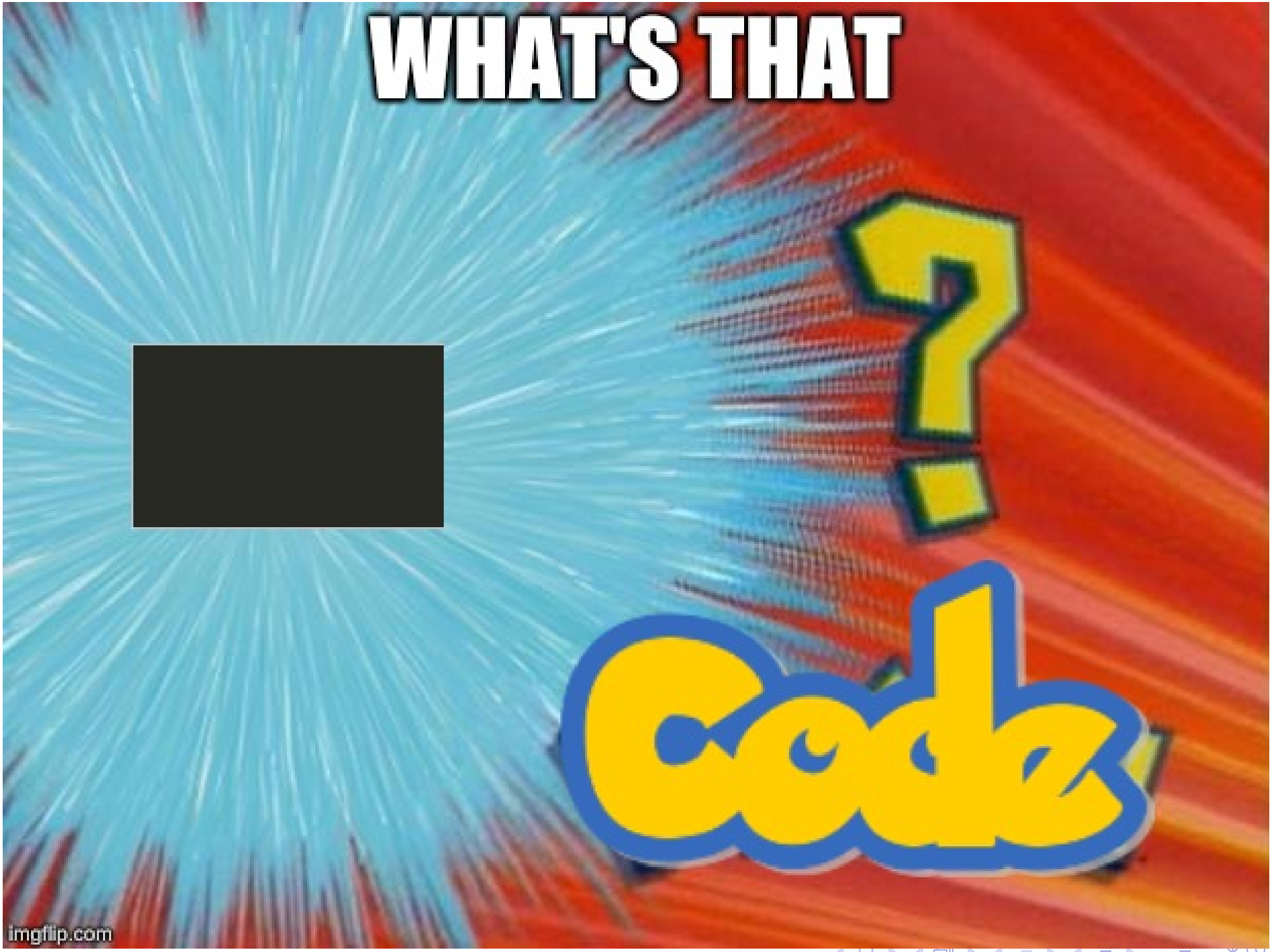
- 24 GB
- 14 Attributes
- 14 million records

Bob's Deli Shop Data

- <1 MB
- 3 Attributes
- 3 records

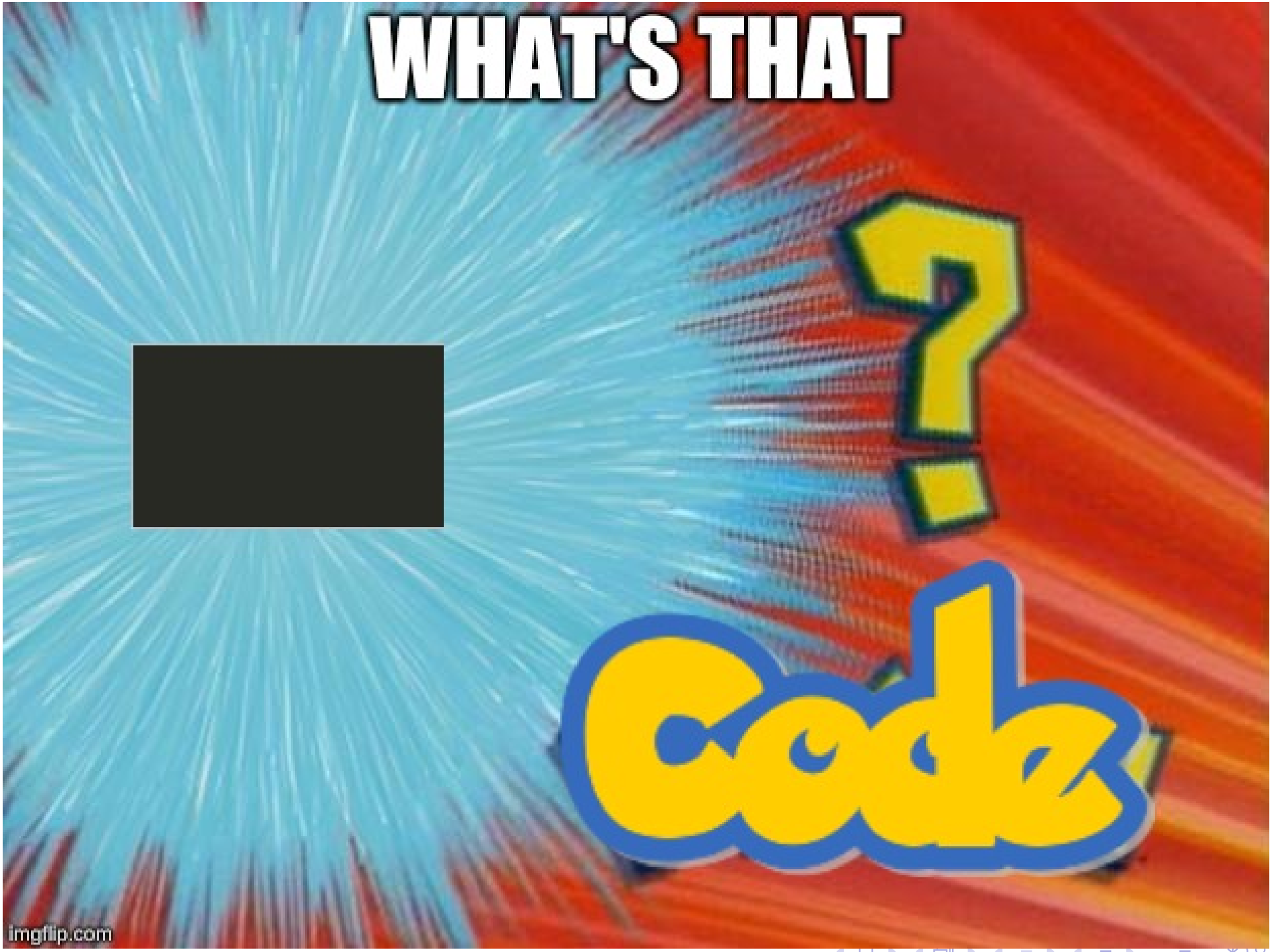

WHAT'S THAT
~900 MB
?
Dataset
imgflip.com

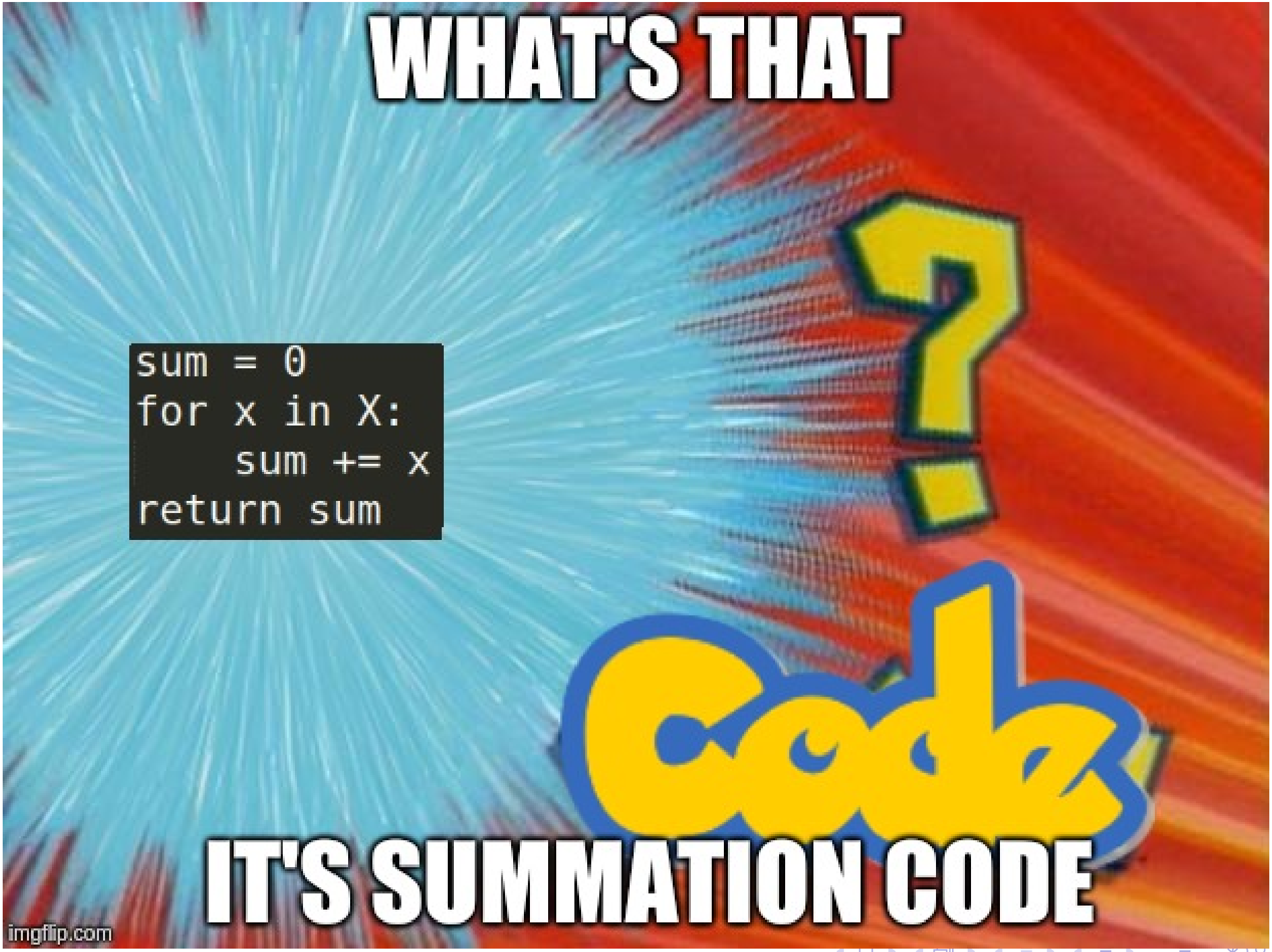

WHAT'S THAT
U.S. Census Data
0 @#$%^!
1 #$%^@*
... ...
n &^%$@#!
Dataset
IT'S THE U.S. CENSUS DATA!
imgflip.com


WHAT'S THAT
?
Code
imgflip.com

```
/*  The last step is then to "logical OR" all these images together. */
loop_ub = bw_a_filled->size[0] - 2;
b_loop_ub = bw_a_filled->size[1] - 2;
i0 = bw_filled->size[0] * bw_filled->size[1];
bw_filled->size[0] = loop_ub + 1;
bw_filled->size[1] = b_loop_ub + 1;
emxEnsureCapacity((emxArray__common *)bw_filled, i0, (int32_T)sizeof(boolean_T));
emxFree_boolean_T(&b_bw_b);
for (i0 = 0; i0 <= b_loop_ub; i0++) {
  for (i1 = 0; i1 <= loop_ub; i1++) {
    bw_filled->data[i1 + bw_filled->size[0] * i0] = (bw_a_filled->data[(i1 +
      bw_a_filled->size[0] * (1 + i0)) + 1] || bw_b_filled->data[(i1 +
      bw_b_filled->size[0] * i0) + 1] || bw_c_filled->data[i1 +
      bw_c_filled->size[0] * i0] || bw_b->data[i1 + bw_b->size[0] * (1 + i0)]);
  }
}
```

```
sum = 0
for x in X:
    sum += x
return sum
```


WHAT'S THAT
?
Code
imgflip.com


WHAT'S THAT
sum = 0
for x in X:
sum += x
return sum
?
Code
IT'S SUMMATION CODE
imgflip.com

Mock Example

Bob's Deli Shop Data for One Day

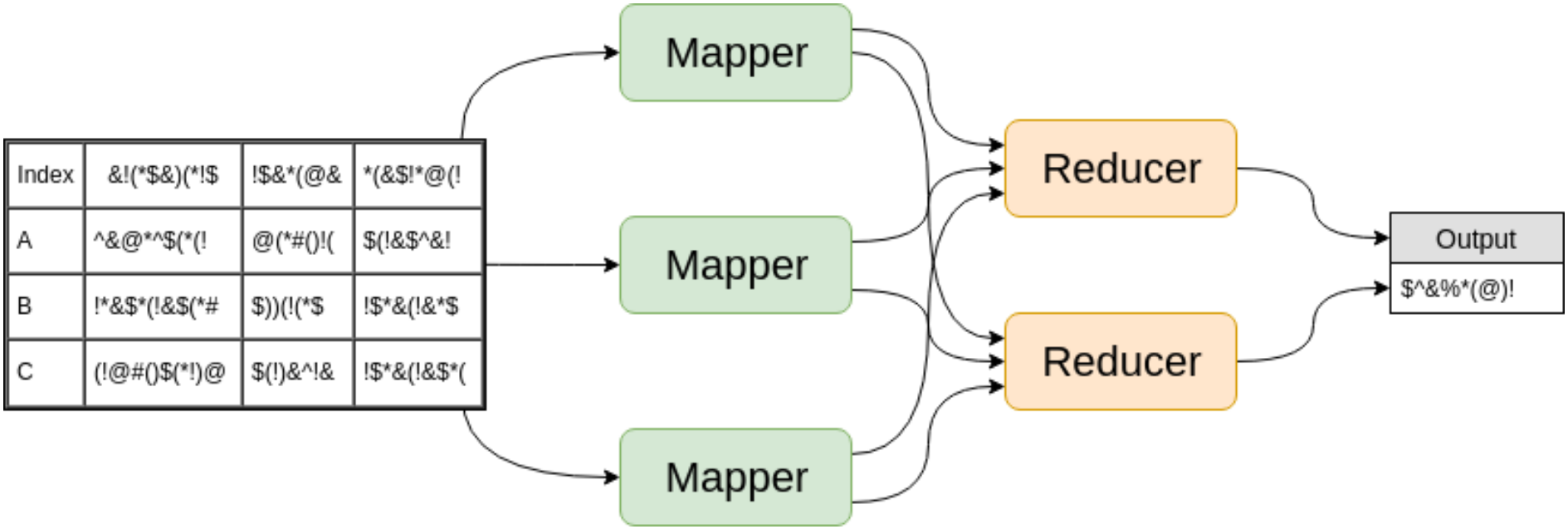
Index	Food Bought	Spent	Approx Age
1	Turkey Sandwich	$4.75	25-34
2	Chicken Sandwich	$4.75	25-34
3	Salad	$3.75	25-34

Mock Example

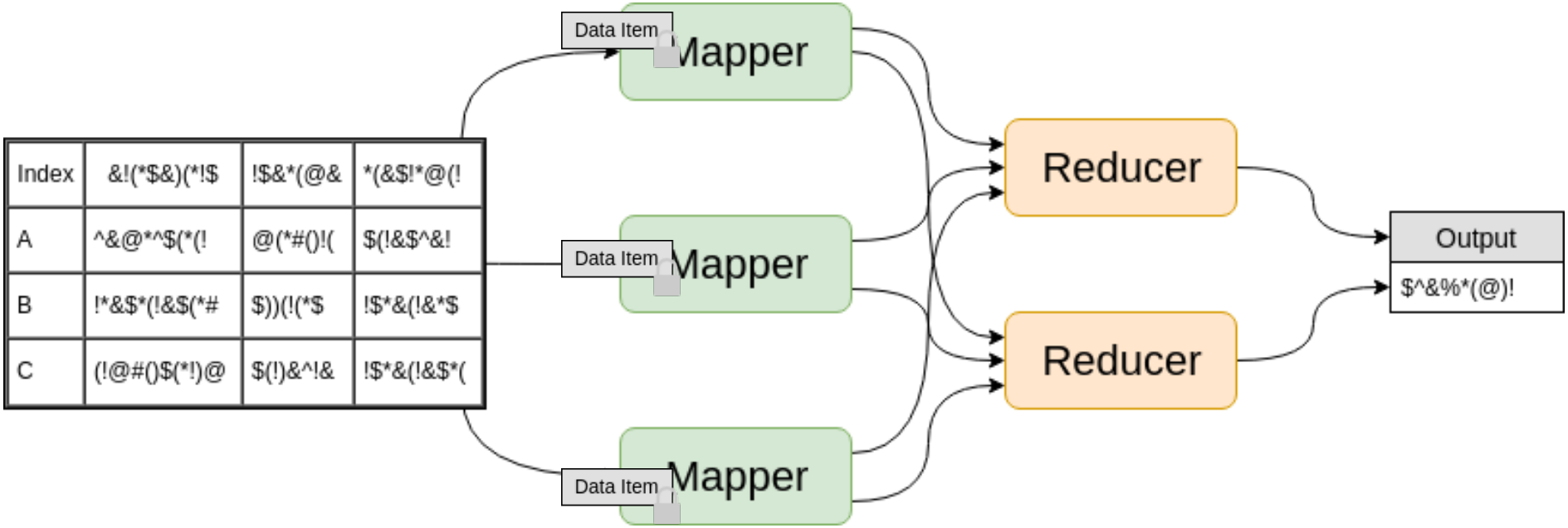
Bob's Deli Shop Data for One Day (Encrypted and Shuffled)

Index	&!(*$&)(*!$	!$&*(@&	*(&$!*@(!
A	^&@*^$(*(!	@(*#()!(	$(!&$^&!
B	!*&$*(!&$(*#	$))(!(*$	!$*&(!&*$
C	(!@#()$(*!)@	$(!)&^!&	!$*&(!&$*(

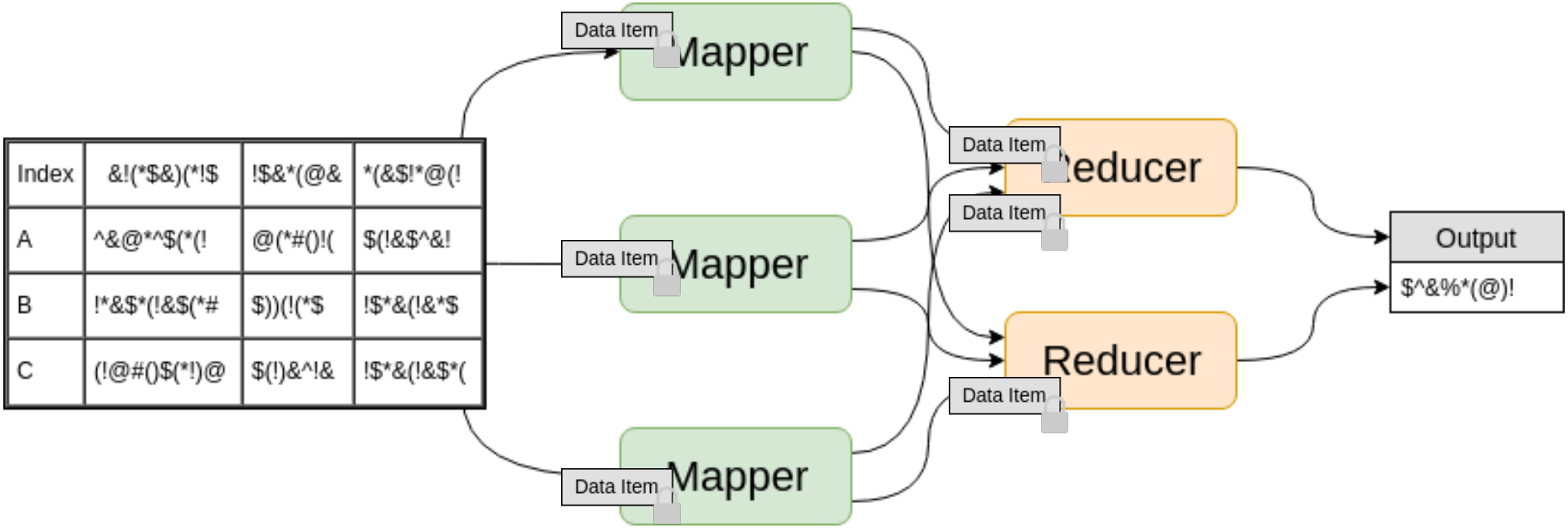
Mock Example



Mock Example



Mock Example



Formalizations

- The goal of this attack is not to uncover new facts about the dataset not readily available from the plaintext.
- Rather, it is to uncover dataset information from observing encrypted traffic.

Formalizations

- The goal of this attack is not to uncover new facts about the dataset not readily available from the plaintext.
- Rather, it is to uncover dataset information from observing encrypted traffic.
- We can track the number of records sent from mapper i to reducer j by maintaining a $M \times R$ matrix, where M and R are the number of mappers and reducers respectively.

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Formalizations

- The goal of this attack is not to uncover new facts about the dataset not readily available from the plaintext.
- Rather, it is to uncover dataset information from observing encrypted traffic.
- We can track the number of records sent from mapper i to reducer j by maintaining a $M \times R$ matrix, where M and R are the number of mappers and reducers respectively.
- An active attacker can send data one record at a time and track each one as a separate row.

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Formalizations

- The goal of this attack is not to uncover new facts about the dataset not readily available from the plaintext.
- Rather, it is to uncover dataset information from observing encrypted traffic.
- We can track the number of records sent from mapper i to reducer j by maintaining a $M \times R$ matrix, where M and R are the number of mappers and reducers respectively.
- An active attacker can send data one record at a time and track each one as a separate row.
- Gathering this information over a few jobs on the same dataset can be enough to recover the full data.

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

US Census Example

- Consider an attacker who has observed three aggregate jobs on the US Census Data. One for age groups, one for place of birth, and one for marital status.
- Consider these jobs were performed one record at a time. This attacker would have recorded 3 matrices, each with 2.5 million rows.

US Census Example

- These matrices combined with knowledge of the distributions (listed below) is enough to identify the keys that each reducer processed.

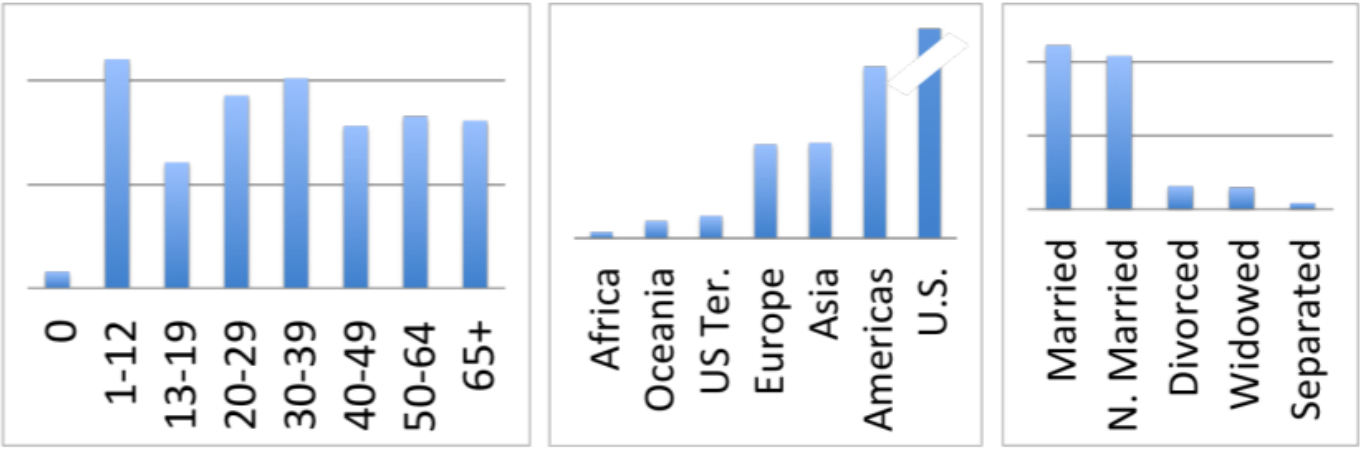


Figure 2: Distribution of Census records across age groups (left), place of birth (center) and marital status (right), where U.S. count is trimmed.

- Let A_{age} be the matrix for age groups. This can be queried like $A_{age}[i, \text{“}1-12\text{”}] == 1$ by overloading the reducers key with the label it processed.

US Census Example

At this point an attacker can answer the follow specific queries:

- Given the index of a record in a dataset, return the values of some attributes
- Test if the dataset contains a record that matches particular values for some of these attributes
- Given the values of some attributes, infer the possible values of the others in the dataset

US Census Example

At this point an attacker can answer the follow specific queries:

- Given the index of a record in a dataset, return the values of some attributes
- Test if the dataset contains a record that matches particular values for some of these attributes
- Given the values of some attributes, infer the possible values of the others in the dataset

For Example

There is one person with Age: 13-19; Birth Place: Oceania; Martial Status: Divorced. This information can be retrieved from the matrices and then the other attribute values can now be inferred.

Security Definitions

- Security for MapReduce is redefined in terms of a game and an adversary's ability to winning the game.
- The game is similar to our "What's that dataset?" game.

Security Definitions

- Security for MapReduce is redefined in terms of a game and an adversary's ability to winning the game.
- The game is similar to our "What's that dataset?" game.
- An adversary picks two datasets, the number of mappers and reducers, and the jobs each will perform.
- A challenger selects one of the datasets, which is then run through the system.
- The adversary can observe network traffic and finally must guess which dataset was selected.

Security Definitions

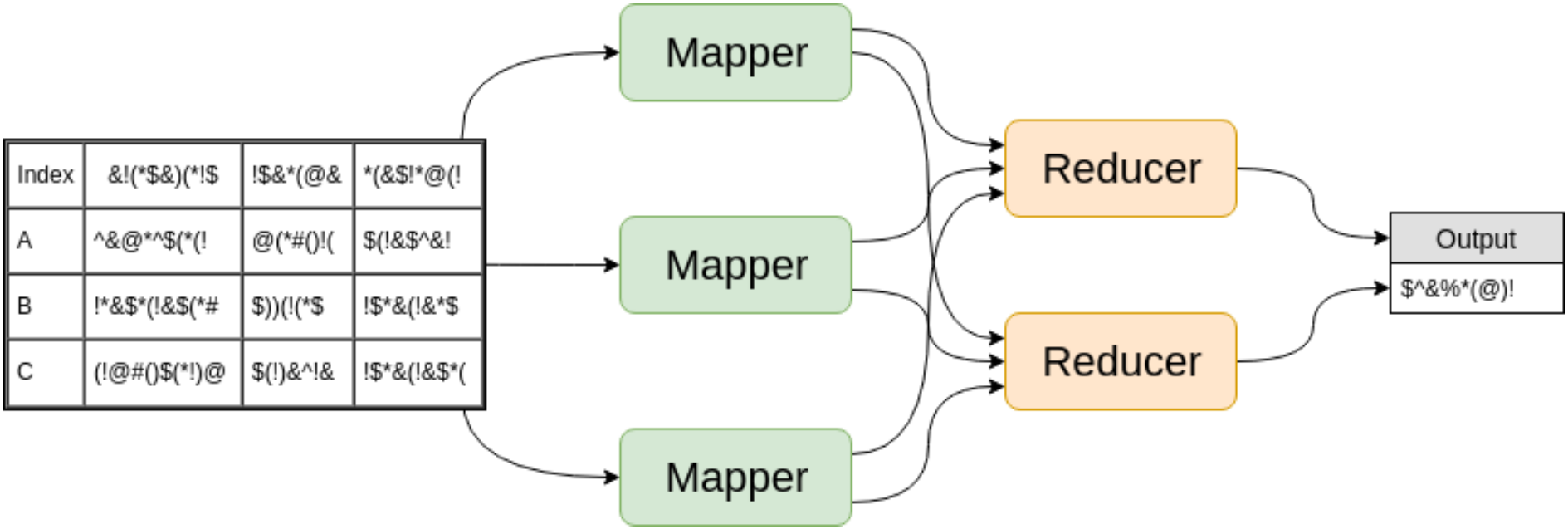
- Security for MapReduce is redefined in terms of a game and an adversary's ability to winning the game.
- The game is similar to our "What's that dataset?" game.
- An adversary picks two datasets, the number of mappers and reducers, and the jobs each will perform.
- A challenger selects one of the datasets, which is then run through the system.
- The adversary can observe network traffic and finally must guess which dataset was selected.

Ideally

The adversary should not have more than negligible advantage in winning game, i.e. should win very close to 50% of the time.

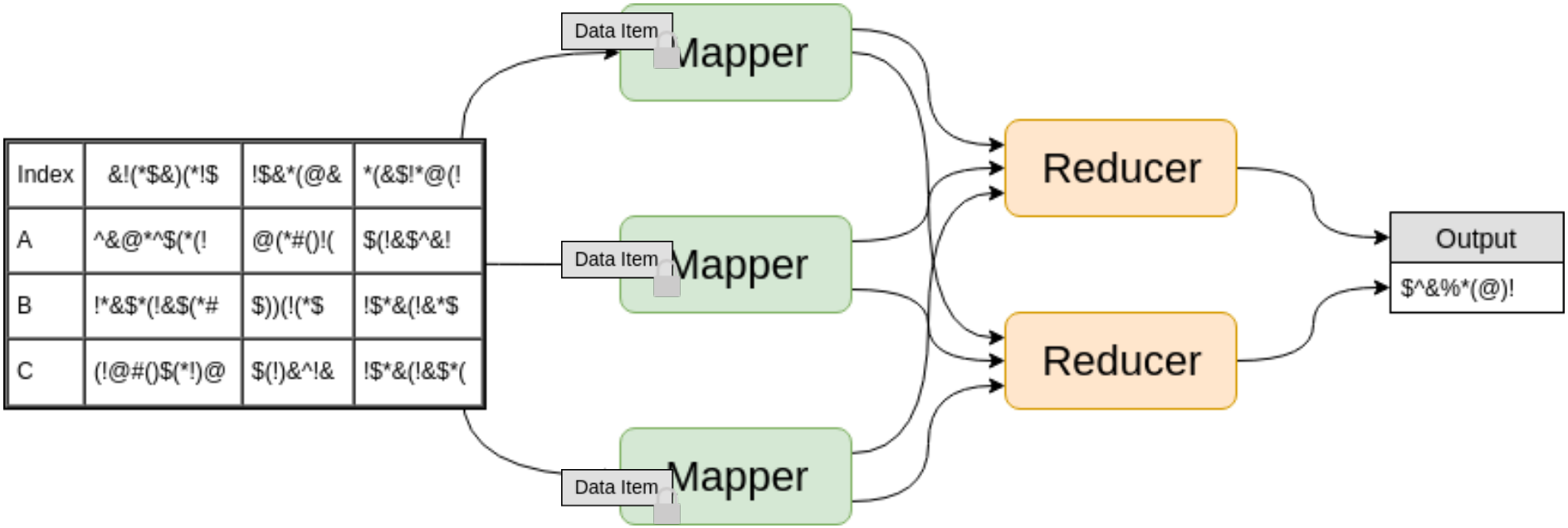
Counter Measures

To accomplish this, we can normalize the amount of intermediate data sent through the network.



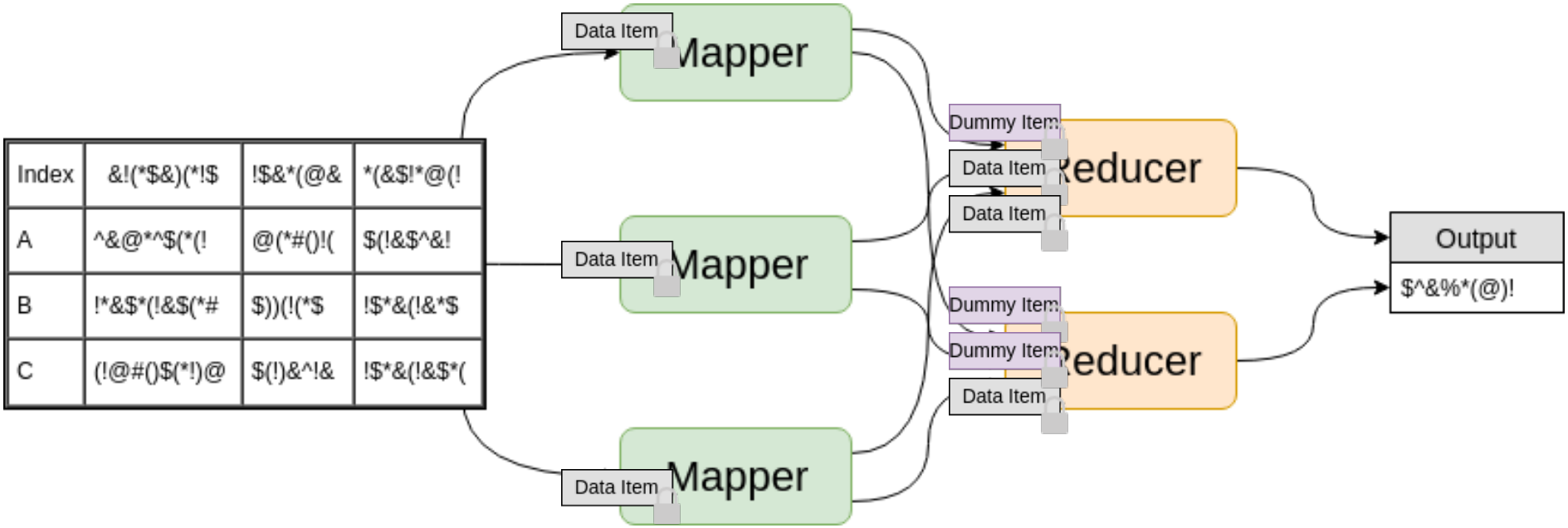
Counter Measures

To accomplish this, we can normalize the amount of intermediate data sent through the network.



Counter Measures

To accomplish this, we can normalize the amount of intermediate data sent through the network.



End of Presentation

Questions?

End of Presentation

Questions?

Feedback?