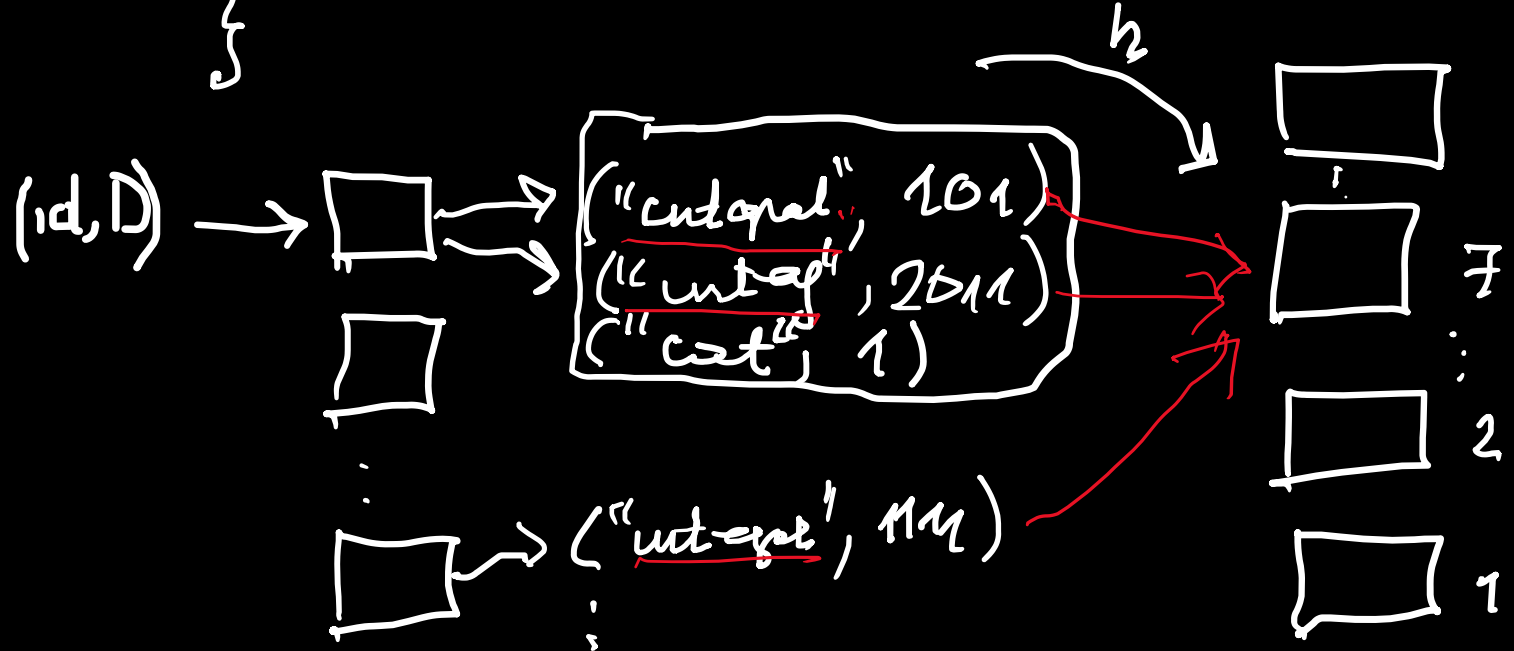
Example. Inverted index

- $d_1, \ldots, d_n$: documents e.g. $n = 10^9$ all web pages

```
map (idD, D) {
    for all w in D do
        emit (w, idD);
}
```

```
reducer (k, L) {
    emit (k, L);
}
```

← identity mapp.



$h(\text{"int-equal"}) = 7$

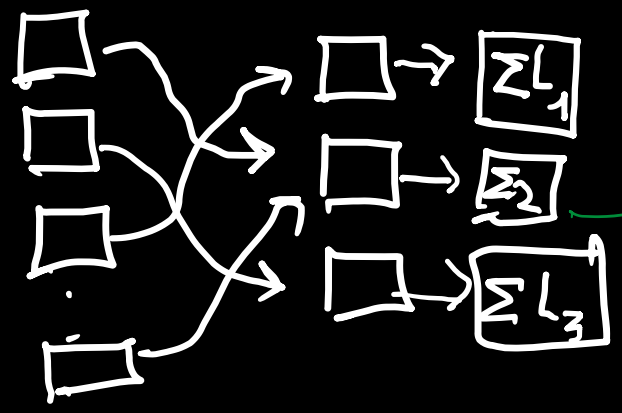
what reducer do:

$L = [(\text{"int-equal"}, 101), (\text{"table"}, 111), \ldots]$

Aggregates of data:

$\vec{x} = (x_1, \dots, x_n)$ ← doubles

- sum: $\sum x_i$; • $\min(\vec{x})$; • $\max(\vec{x})$



$\varphi(x) =$ "last 10 digits in x" $\in \{0, \dots, 2^{10}-1\}$

```
map(x) {
  .emit(φ(x), x);
}
```

$$\sum_{L=1}^{r} \left(\sum_{i} L_i \right)$$

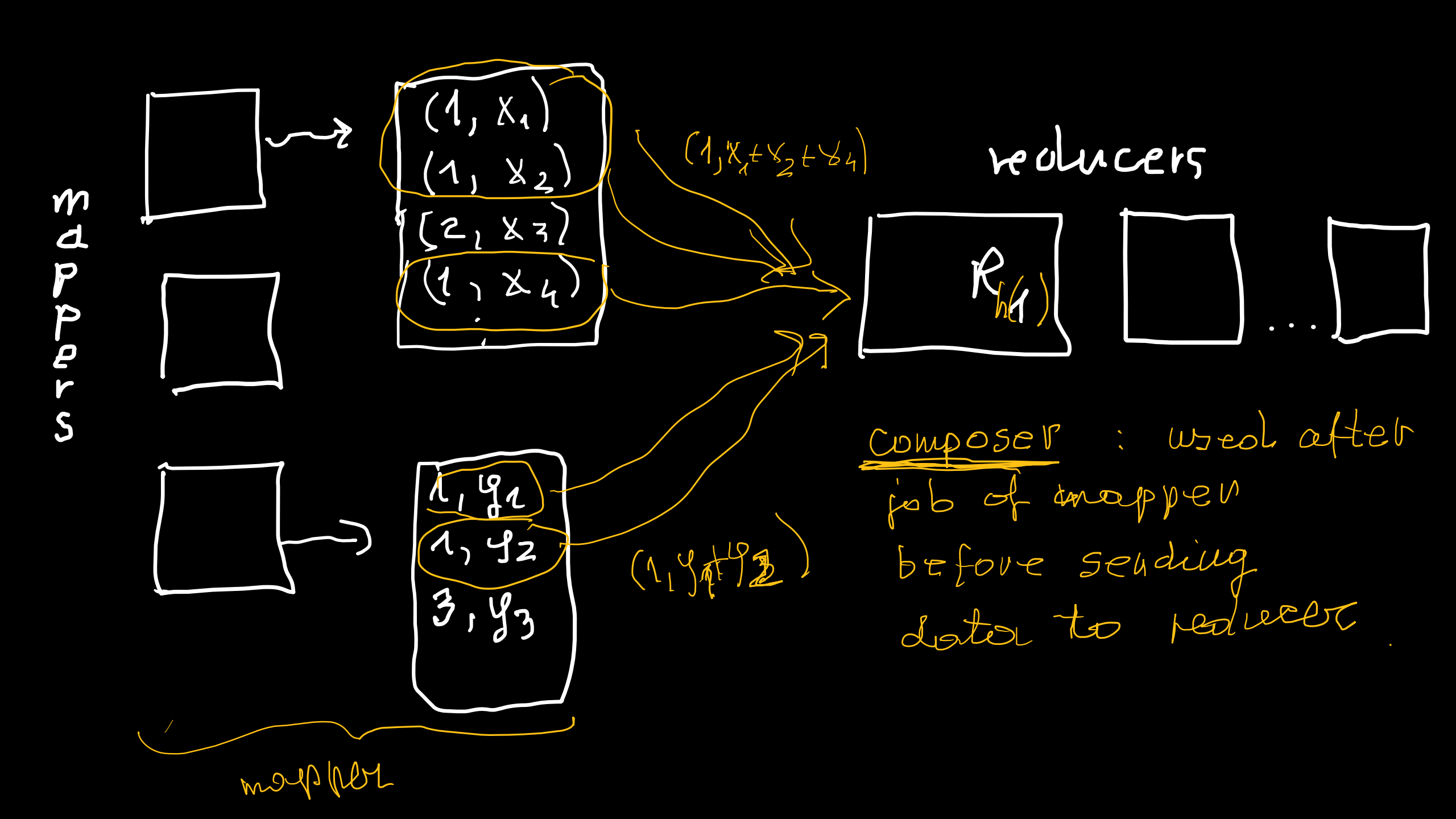
=

```
reduce(k, L) { emit(1, ΣL); }
```

$\sum (L_1 \| L_2 \| \dots \| L_r)$

{ emit (1, min(L)); } . . .

+ is commutative and associative


mappers
(1, x1)
(1, x2)
(2, x3)
(1, x4)
(1, y1)
(1, y2)
3, y3
(1, x1+x2+x4)
(1, y1+y2)
reducers
R h(1)
...
mapper
Composer : used after job of mapper before sending data to reducers.

```
compose (k, L) {
    emit (k, Σ L);
}
```

---

when we can use it (composer)?

```
reduce (k, L) {
    emit (k, θ(L));
}
```

$$\theta(x_1, \ldots, x_n) = \begin{cases} \Sigma x_i, \\ \min x_i \\ \max x_i \end{cases}$$

$$\theta(\theta(l_1), \theta(l_2), \ldots, \theta(l_m)) = \theta(l_{\pi(1)} \| \ldots \| l_{\pi(m)})$$

$\pi \leftarrow$ perm. of $1, \ldots, m$.

$$\theta(x_1, \ldots, x_5) = x_1 + \ldots + x_5$$

$$\theta\left(\theta(x_5, x_6), \theta(x_1, x_2, x_3, x_4)\right) =$$

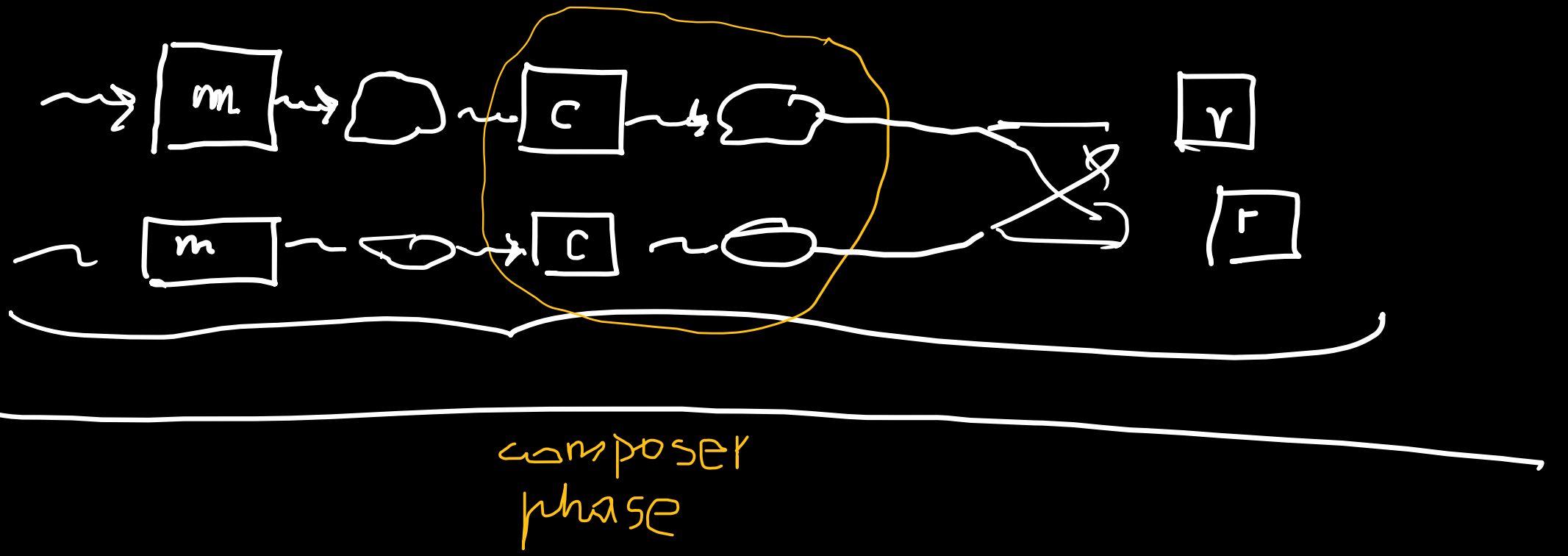
$$= \theta(x_5 + x_6,\ x_1 + x_2 + \ldots + x_4) =$$

$$= (x_5 + x_6) + (x_1 + x_2 + x_3 + x_4) =$$

$$= x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

(+) is commutative, associative

use composer if operation
is commutative and associative



$x_1 \ldots x_n$; Σ, min, max

```
map (x) {
  emit (φ(x), (x, x, x));
}
```

$L = [(x_1, x_{\varphi(1)}, x_1), (x_2, x_2, x_2), \ldots (x_K, x_K, x_K)]$

```
compose (K, L) {
  emit (K, Σ π1(L), min (π2(L)), max (π3(L)));
}
```

```
reducer (K, L) {
  — if —
}
```

$L = [(s_1, m_1, M_1), (s_2, m_2, M_2), \ldots]$

Mean value of $x_1 \dots x_n$.

$$\frac{x_1 + \dots + x_n}{n}$$

```
map(x) { emit(φ(x), (1, x)) }
compose (k, L) {
    emit (k, (Σ π_1(L), Σ π_2(L)))
}
reduce (k, L) {
}
```

$L = [(1, x_1), (1, x_2), \dots (1, x_4)]$

$(3, x_1 + x_2 + x_3)$

$R_1 : [(n_1, s_1), (n_2, s_2), \vdots$

$$\frac{s_1 + \dots + s_n}{n_1 + \dots + n_r}$$

WARIANCE

$$\mathrm{Var}(\vec{x}) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \underbrace{\frac{x_1 + \dots + x_n}{n}}_{\mu}\right)^2 =$$

$n = |\vec{x}|$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(x_i^2 - 2x_i\mu + \mu^2\right) = \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2 - 2\mu\sum x_i + \mu^2\sum_{i=1}^{n} 1\right)$$

$$= \frac{1}{n}\left(\sum x_i^2 - 2\mu\cdot(\mu\cdot n) + \mu^2\cdot n\right) =$$

$$= \frac{1}{n}\sum x_i^2 - \mu^2, = \frac{1}{n}\sum x_i^2 - \left(\frac{\sum x_i}{n}\right)^2$$

$$\mathrm{Var}(X) = E(X^2) - [E(X)]^2$$

map(x) { emit(φ(x), (1, x, x²)); }

reduce(k, L) {

  emit(1, Σπ₁(L), Σπ₂(L), Σπ₃(L))

}

k :

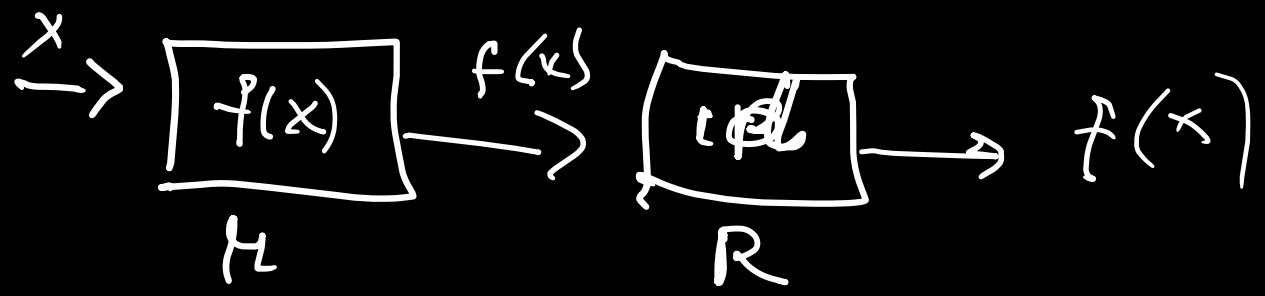
L = [(1, x₁, x₁²), (1, x₂, x₂²), ...]

mean(x), var(x)

EXERCISE: calc.: $\frac{1}{n}\sum(x_i - \mu)^3$ { central 3-th moment

x ⟶ (φ(x), (1, x, x², x³))

What can be calculated?



Set operations: $\cup, \cap, \setminus$.

A, B – too big sets

$x \in A \rightsquigarrow$ ("A", x)
$x \in B \rightsquigarrow$ ("B", x)

} input for mapper

```
map(X, x) {
  emit(x, X);
}
```

SUM

```
reduce(x, L){
  emit(1, x);
}
```

INTERS.

```
reduce(x, L){
  if(|L|=2) then emit(1, x);
}
```

DIFF:

```
reduce(x, L){if(L=["A"]) then emit(1, x)}
```

reducer

(x, L)

L = ["A"]

L = ["B"]

L = ["A", "B"]

L = ["B", "A"]

SELECT $a_1 .., a_4$ from R

WHERE $\theta(a_1 .. a_4)$

if $\theta(a_1 .. a_4) \neq \text{false}$

map $(a_1 .. a_4)$ {emit $(\varphi(a_1 .. a_4), (a_1 .. a_4))$}

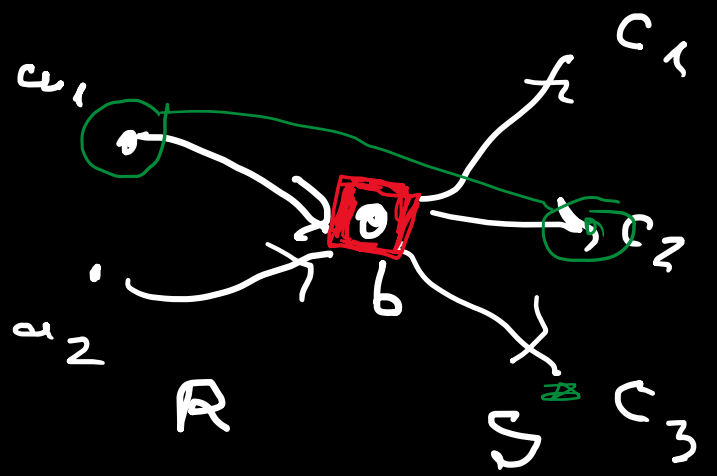
JOIN: $R(a,b)$, $S(b,c)$

SELECT R.a, S.c FROM R, S

WHERE R.b = S.b

INPUT: $R(a,b) \rightsquigarrow$ ("R", a, b)

$S(b,c) \rightsquigarrow$ ("S", b, c)



("R", a, b) ⟶ (b, (R, a))

("S", b, c) ⟶ (b, (S, c))

```
map (X, x, y) {
    if (X = "R") {emit (y, ("R", x));}
        else     {emit (x, ("S", y));}
}

reducer (k, L) {
    (L1, L2) ⇐ split (L)
    for all (R,x) in L1 for all (S,y) in L2 emit (1, [x, y]); }
```

starts from R

starts from S

L = [(R, x1), (S, y1), (R, x2), (R, x3), (S, y2) ...]

↓

[(R, x1), (R, x2) ... (R, xk)]

[(S, y1), ... (S, ye)]