

Materialized Views

March 28, 2018

```
CREATE VIEW salesSinceLastMonth AS
  SELECT l.*
  FROM lineitem l, orders o
  WHERE l.orderkey = o.orderkey
        AND o.orderdate > DATE('2015-03-31')
```

```
CREATE VIEW salesSinceLastMonth AS
  SELECT l.*
  FROM lineitem l, orders o
  WHERE l.orderkey = o.orderkey
        AND o.orderdate > DATE('2015-03-31')

  SELECT partkey FROM salesSinceLastMonth
  ORDER BY shipdate DESC LIMIT 10;
```

```
CREATE VIEW salesSinceLastMonth AS
SELECT l.*
FROM lineitem l, orders o
WHERE l.orderkey = o.orderkey
      AND o.orderdate > DATE('2015-03-31')
```

```
SELECT partkey FROM salesSinceLastMonth
ORDER BY shipdate DESC LIMIT 10;
```

```
SELECT suppkey, COUNT(*)
FROM salesSinceLastMonth
GROUP BY suppkey;
```

```
CREATE VIEW salesSinceLastMonth AS
SELECT l.*
FROM lineitem l, orders o
WHERE l.orderkey = o.orderkey
      AND o.orderdate > DATE('2015-03-31')
```

```
SELECT partkey FROM salesSinceLastMonth
ORDER BY shipdate DESC LIMIT 10;
```

```
SELECT suppkey, COUNT(*)
FROM salesSinceLastMonth
GROUP BY suppkey;
```

```
SELECT partkey, COUNT(*)
FROM salesSinceLastMonth
GROUP BY partkey;
```

Materialization

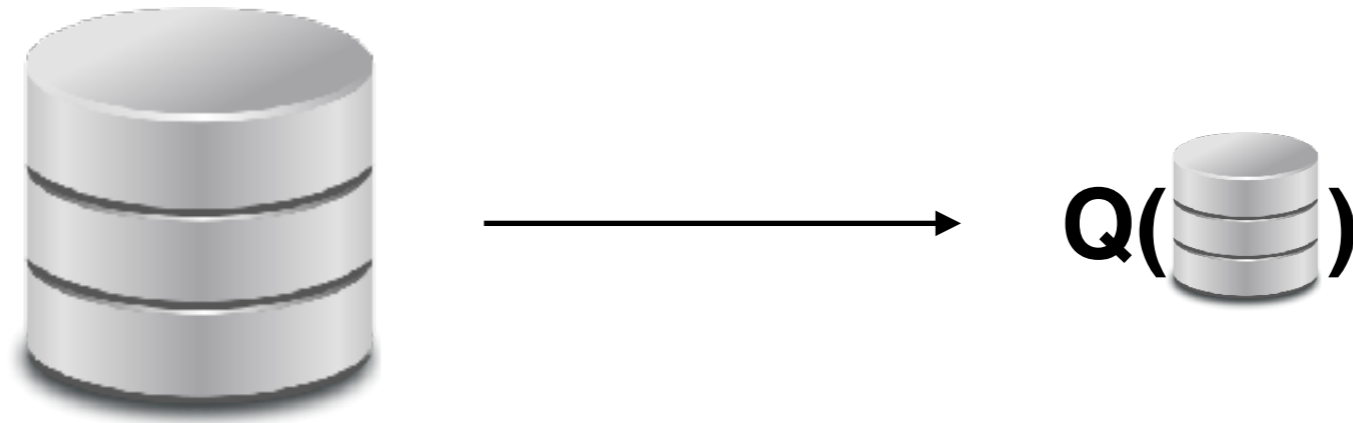
Views exist to be queried frequently

Pre-compute and save the view's contents!
(like an index)

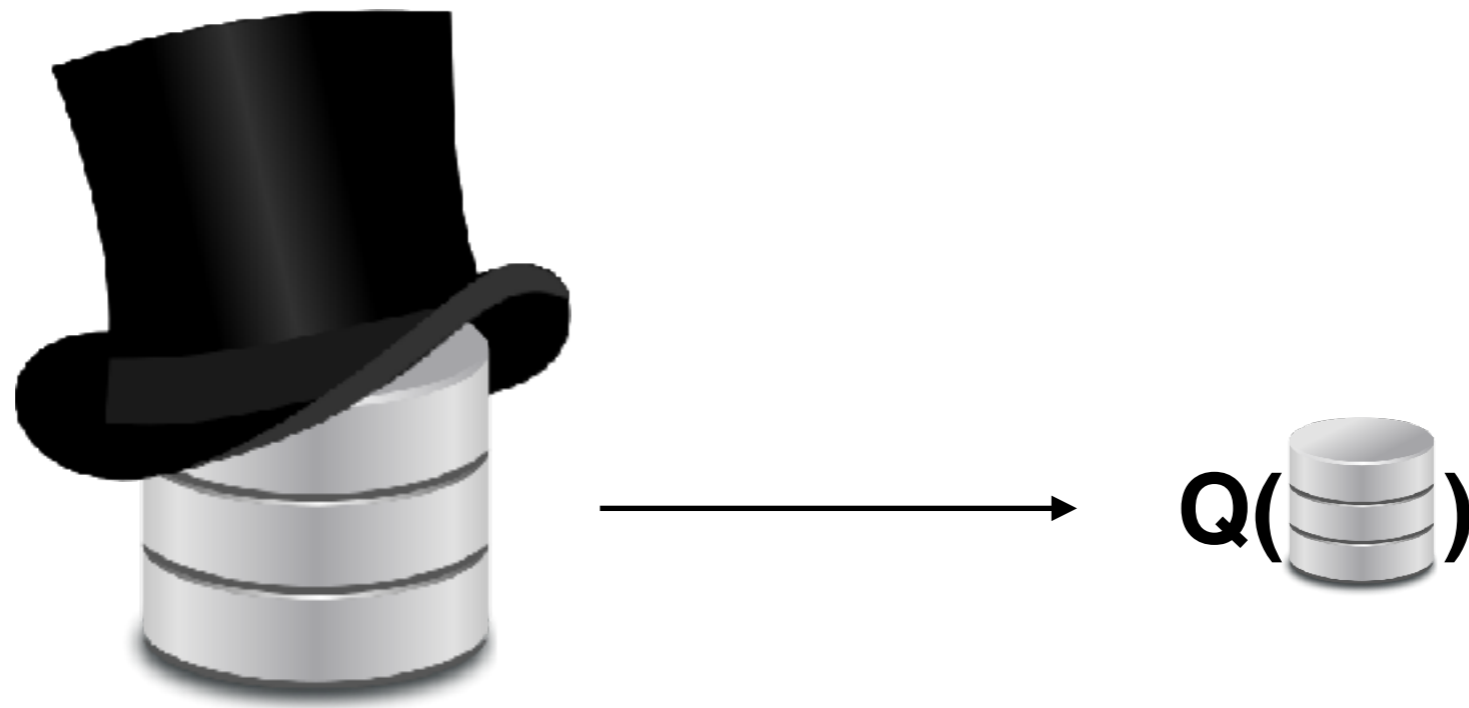
Materialized Views



Materialized Views



Materialized Views



Materialized Views



When the base data changes, the view needs to be updated

View Maintenance

VIEW ← **Q(D)**

View Maintenance

```
WHEN  $D \leftarrow D + \Delta D$  DO:  
    VIEW  $\leftarrow Q(D + \Delta D)$ 
```

View Maintenance

WHEN $D \leftarrow D + \Delta D$ DO:
VIEW $\leftarrow Q(D + \Delta D)$

Re-evaluating the query from scratch is expensive!

View Maintenance

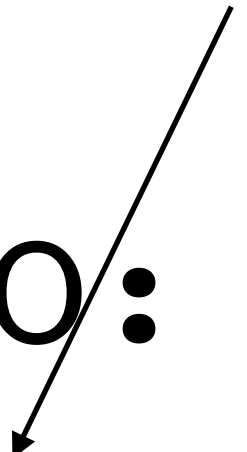
WHEN $D \leftarrow D + \Delta D$ DO:

$VIEW \leftarrow VIEW + \Delta Q(D, \Delta D)$

View Maintenance

(ideally) Smaller & Faster Query

WHEN $D \leftarrow D + \Delta D$ DO:
 VIEW $\leftarrow \text{VIEW} + \Delta Q(D, \Delta D)$



View Maintenance

(ideally) Smaller & Faster Query

WHEN $D \leftarrow D + \Delta D$ DO:

VIEW $\leftarrow \text{VIEW} + \Delta Q(D, \Delta D)$

(ideally) Fast “merge” operation.

Intuition

$$D = \{1, 2, 3, 4\} \quad \Delta D = \{5\}$$

$$Q(D) = \text{SUM}(D)$$

Intuition

$$D = \{1, 2, 3, 4\} \quad \Delta D = \{5\}$$

$$\underline{Q}(D) = \text{SUM}(D)$$

$$\underline{Q}(D + \Delta D) \sim O(|D| + |\Delta D|)$$

Intuition

$$D = \{1, 2, 3, 4\} \quad \Delta D = \{5\}$$

$$\underline{Q}(D) = \text{SUM}(D)$$

$$\underline{Q}(D + \Delta D) \sim O(|D| + |\Delta D|)$$

$$\text{VIEW} + \text{SUM}(\Delta D) \sim O(|\Delta D|)$$

Intuition

$$R = \{1, 2, 3\}, S = \{5, 6\} \quad \Delta R = \{4\}$$

$$Q(R, S) = \text{COUNT}(R \times S)$$

Intuition

$$R = \{1, 2, 3\}, S = \{5, 6\} \quad \Delta R = \{4\}$$

$$Q(R, S) = \text{COUNT}(R \times S)$$

$$Q(R + \Delta R, S) \sim O((|R| + |\Delta R|) * |S|)$$

Intuition

$$R = \{1, 2, 3\}, S = \{5, 6\} \quad \Delta R = \{4\}$$

$$Q(R, S) = \text{COUNT}(R \times S)$$

$$Q(R + \Delta R, S) \sim O((|R| + |\Delta R|) * |S|)$$

$$\text{VIEW} + \text{COUNT}(|\Delta R| * |S|) \sim O(|\Delta R| * |S|)$$

Incremental View Maintenance

WHEN $D \leftarrow D + \Delta D$ DO:

$VIEW \leftarrow VIEW + \Delta Q(D, \Delta D)$

Incremental View Maintenance

WHEN $D \leftarrow D + \Delta D$ DO:

$VIEW \leftarrow VIEW + \Delta Q(D, \Delta D)$

Basic Challenges of IVM

What does ΔR represent?

Incremental View Maintenance

WHEN $D \leftarrow D + \Delta D$ **DO:**

VIEW $\leftarrow \text{VIEW} + \Delta Q(D, \Delta D)$

Basic Challenges of IVM

What does ΔR represent?

How to interpret $R \pm \Delta R$?

Incremental View Maintenance

WHEN $D \leftarrow D + \Delta D$ DO:

$VIEW \leftarrow VIEW + \Delta Q(D, \Delta D)$

Basic Challenges of IVM

What does ΔR represent?

How to interpret $R \pm \Delta R$?

How to compute ΔQ ?

What is ΔR ?

What is ΔR ?

What does it need to represent?

What is ΔR ?

What does it need to represent?

Insertions

Deletions

Updates

What is ΔR ?

What does it need to represent?



Insertions

Deletions

Updates

(Delete Old Record & Insert Updated Record)

What is ΔR ?

What is ΔR ?

A Set/Bag of Insertions

A Set/Bag of Deletions

What is $+$?

R

A Set/Bag

ΔR

A Set/Bag of Insertions

A Set/Bag of Deletions

What is $+$?

$$\mathbf{R} \quad + \quad \mathbf{\Delta R}$$

A Set/Bag

$+$

A Set/Bag of Insertions

A Set/Bag of Deletions

What is $+$?

R **+** **ΔR**

A Set/Bag

A Set/Bag of Insertions

+

A Set/Bag of Deletions

R

U

$\Delta R_{\text{inserted}}$

-

$\Delta R_{\text{deleted}}$

What is +?

R **+** **ΔR**

A Set/Bag

+

A Set/Bag of Insertions

A Set/Bag of Deletions

R

U

$\Delta R_{\text{inserted}}$

-

$\Delta R_{\text{deleted}}$

But this breaks closure of '+'!

Incremental View Maintenance

$$\text{VIEW} \leftarrow \text{VIEW} + \Delta Q (D, \Delta D)$$

Incremental View Maintenance

$VIEW \leftarrow VIEW - \Delta Q(D, \Delta D)$

Incremental View Maintenance

$VIEW \leftarrow VIEW - \Delta Q(D, \Delta D)$

Given $Q(R, S, \dots)$

Construct $\Delta Q(R, \Delta R, S, \Delta S, \dots)$

Delta Queries

$$\Delta(\sigma(R))$$

σ

|

R

Delta Queries

$$\Delta(\sigma(R))$$

σ

|

R

R

ΔR

Original R

Inserted
Tuples of R

Delta Queries

$$\Delta(\sigma(R))$$

σ

|

R

Original R

σ

|

ΔR

Inserted
Tuples of R

Delta Queries

$$\Delta(\sigma(R)) = \sigma(\Delta R)$$

σ
|
R

R

Original R

σ
|
 ΔR

Inserted
Tuples of R

Delta Queries

$$\Delta(\sigma(R)) = \sigma(\Delta R)$$

σ
|
R

R

σ
|
 ΔR

Original R

Inserted
Tuples of R

Does this work for deleted tuples?

Delta Queries

$$\Delta(\pi(R)) = \pi(\Delta R)$$

π
|
R

R

π
|
 ΔR

Delta Queries

$$\Delta(\pi(R)) = \pi(\Delta R)$$

π
|
R

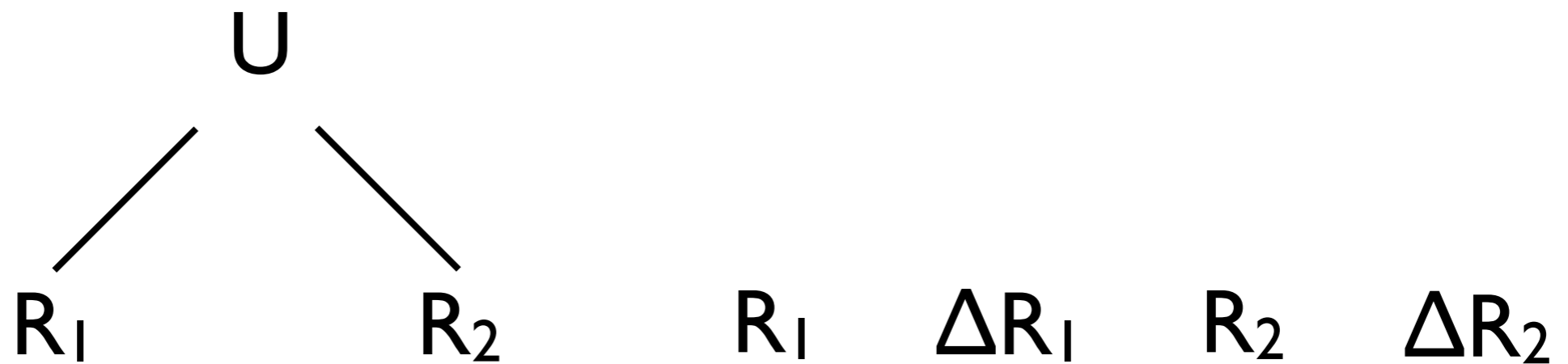
R

π
|
 ΔR

Does this work (completely) under set semantics?

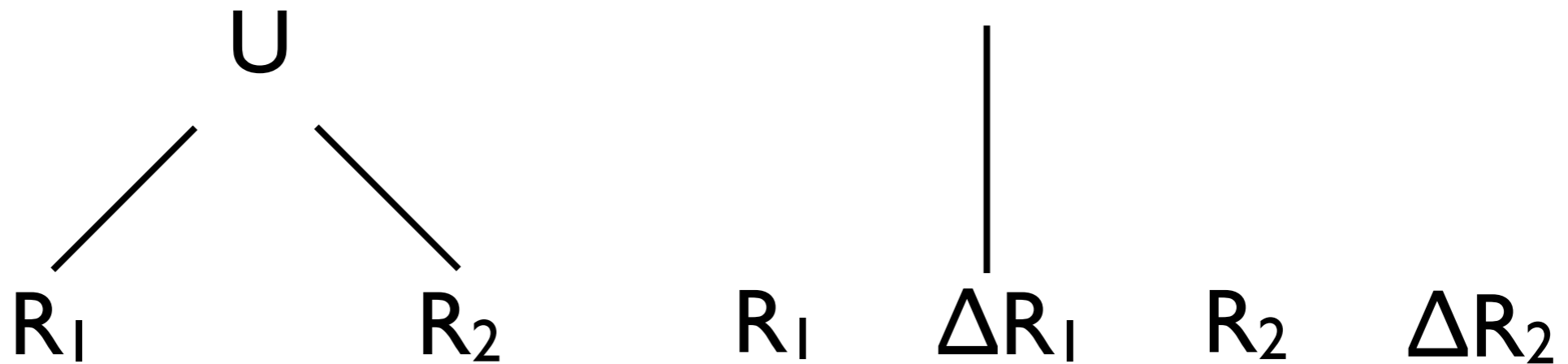
Delta Queries

$$\Delta(R_1 \cup R_2)$$



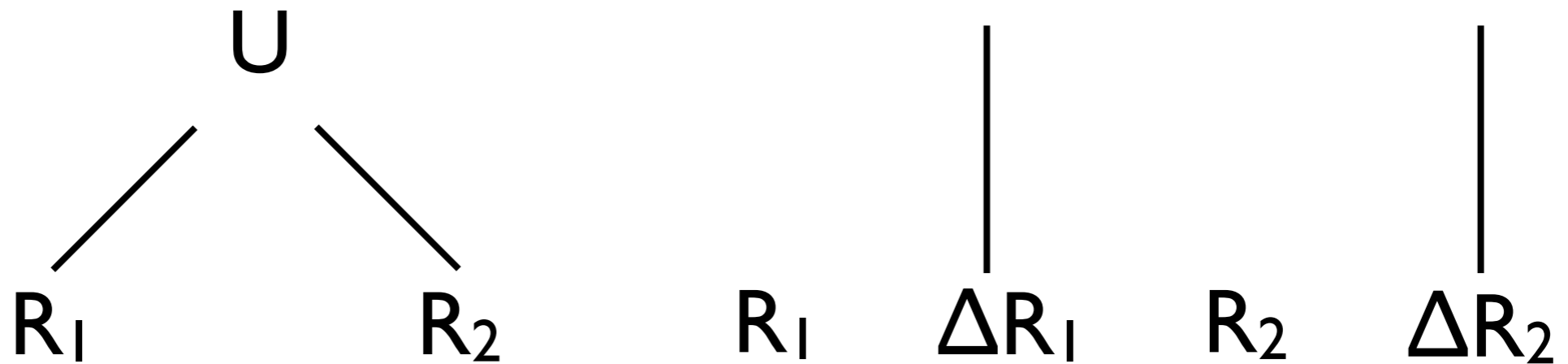
Delta Queries

$$\Delta(R_1 \cup R_2) = \Delta R_1 \cup \Delta R_2$$

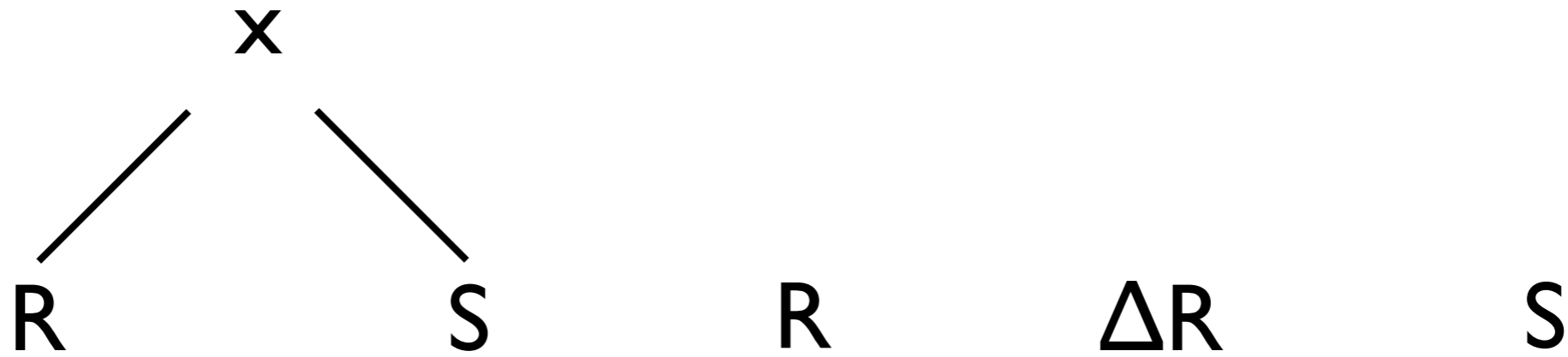


Delta Queries

$$\Delta(R_1 \cup R_2) = \Delta R_1 \cup \Delta R_2$$



Delta Queries



Delta Queries

$R : \{ 1, 2, 3 \}$ $S : \{ 5, 6 \}$

Delta Queries

$$R : \{ 1, 2, 3 \} \quad S : \{ 5, 6 \}$$

$$R \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle 2, 5 \rangle, \langle 2, 6 \rangle, \langle 3, 5 \rangle, \langle 3, 6 \rangle \}$$

Delta Queries

$$R : \{ 1, 2, 3 \} \quad S : \{ 5, 6 \}$$

$$R \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle 2, 5 \rangle, \langle 2, 6 \rangle, \langle 3, 5 \rangle, \langle 3, 6 \rangle \}$$

$$\Delta R_{\text{inserted}} = \{ 4 \}$$

$$\Delta R_{\text{deleted}} = \{ 3, 2 \}$$

Delta Queries

$$R : \{ 1, 2, 3 \} \quad S : \{ 5, 6 \}$$

$$R \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle 2, 5 \rangle, \langle 2, 6 \rangle, \langle 3, 5 \rangle, \langle 3, 6 \rangle \}$$

$$\Delta R_{\text{inserted}} = \{ 4 \}$$

$$\Delta R_{\text{deleted}} = \{ 3, 2 \}$$

$$(R + \Delta R) \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle \mathbf{4}, 5 \rangle, \langle \mathbf{4}, 6 \rangle \}$$

Delta Queries

$$R : \{ 1, 2, 3 \} \quad S : \{ 5, 6 \}$$

$$R \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle 2, 5 \rangle, \langle 2, 6 \rangle, \langle 3, 5 \rangle, \langle 3, 6 \rangle \}$$

$$\Delta R_{\text{inserted}} = \{ 4 \}$$

$$\Delta R_{\text{deleted}} = \{ 3, 2 \}$$

$$(R + \Delta R) \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle \mathbf{4}, 5 \rangle, \langle \mathbf{4}, 6 \rangle \}$$

$$\Delta_{\text{inserted}}(R \times S) = \Delta R_{\text{inserted}} \times S$$

$$\Delta_{\text{deleted}}(R \times S) = \Delta R_{\text{deleted}} \times S$$

Delta Queries

$$R : \{ 1, 2, 3 \} \quad S : \{ 5, 6 \}$$

$$R \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle 2, 5 \rangle, \langle 2, 6 \rangle, \langle 3, 5 \rangle, \langle 3, 6 \rangle \}$$

$$\Delta R_{\text{inserted}} = \{ 4 \}$$

$$\Delta R_{\text{deleted}} = \{ 3, 2 \}$$

$$(R + \Delta R) \times S = \{ \langle 1, 5 \rangle, \langle 1, 6 \rangle, \langle \mathbf{4}, 5 \rangle, \langle \mathbf{4}, 6 \rangle \}$$

$$\Delta_{\text{inserted}}(R \times S) = \Delta R_{\text{inserted}} \times S$$

$$\Delta_{\text{deleted}}(R \times S) = \Delta R_{\text{deleted}} \times S$$

What if R and S both change?

Delta Queries

Computing a Delta Query

$$\Delta(\sigma(R)) = \sigma(\Delta R)$$

$$\Delta(\pi(R)) = \pi(\Delta R)$$

$$\Delta(R_1 \cup R_2) = \Delta R_1 \cup \Delta R_2$$

$$\Delta(R_1 \times R_2) = ??$$

Delta Queries

$$(R_1 \cup \Delta R_1) \times (R_2 \cup \Delta R_2)$$

Delta Queries

$$(R_1 \cup \Delta R_1) \times (R_2 \cup \Delta R_2)$$

$$(R_1 \times R_2) \cup (R_1 \times \Delta R_2) \cup (\Delta R_1 \times R_2) \cup (\Delta R_1 \times \Delta R_2)$$

Delta Queries

$$(R_1 \cup \Delta R_1) \times (R_2 \cup \Delta R_2)$$

$$(R_1 \times R_2) \cup (R_1 \times \Delta R_2) \cup (\Delta R_1 \times R_2) \cup (\Delta R_1 \times \Delta R_2)$$

**The original
query**

Delta Queries

$$(R_1 \cup \Delta R_1) \times (R_2 \cup \Delta R_2)$$

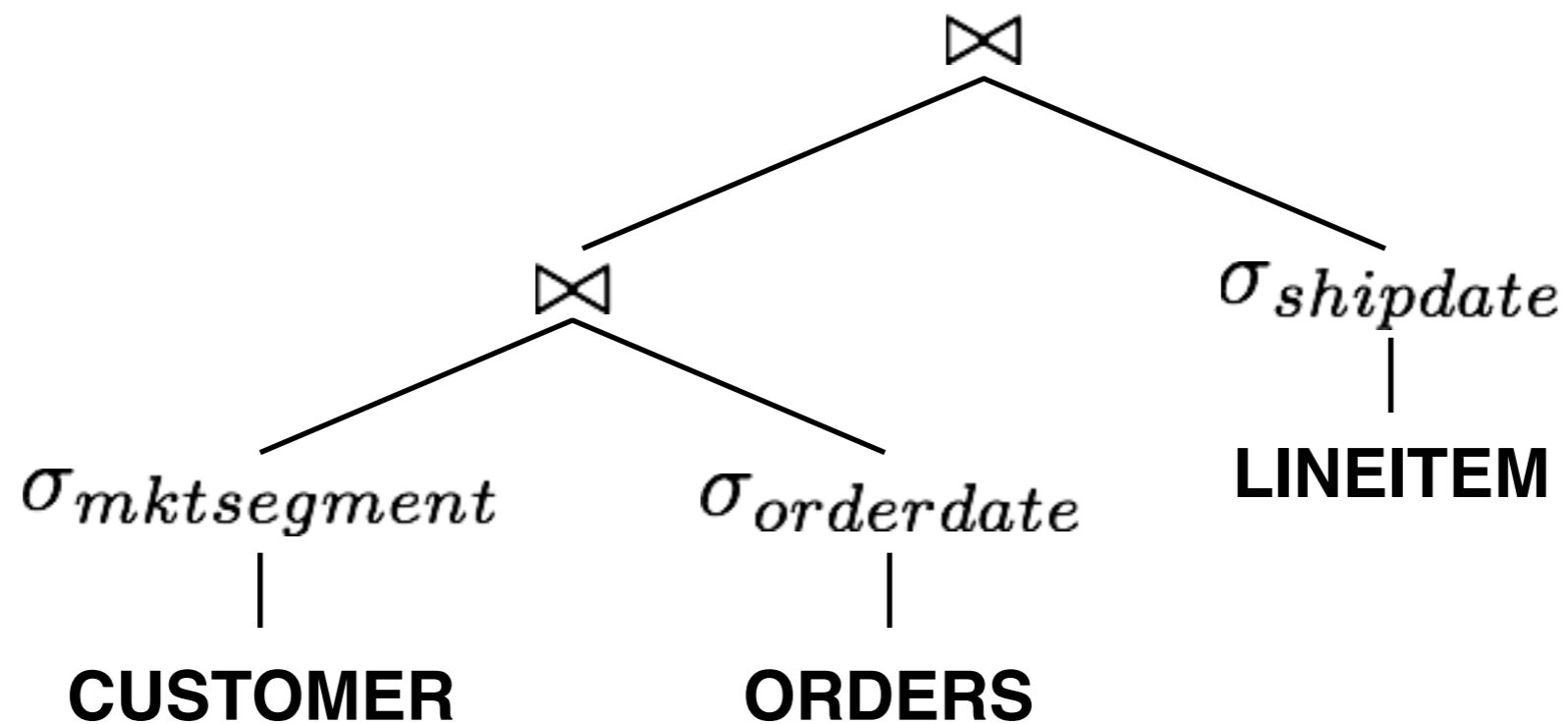
$$(R_1 \times R_2) \cup (R_1 \times \Delta R_2) \cup (\Delta R_1 \times R_2) \cup (\Delta R_1 \times \Delta R_2)$$

**The original
query**

The delta query

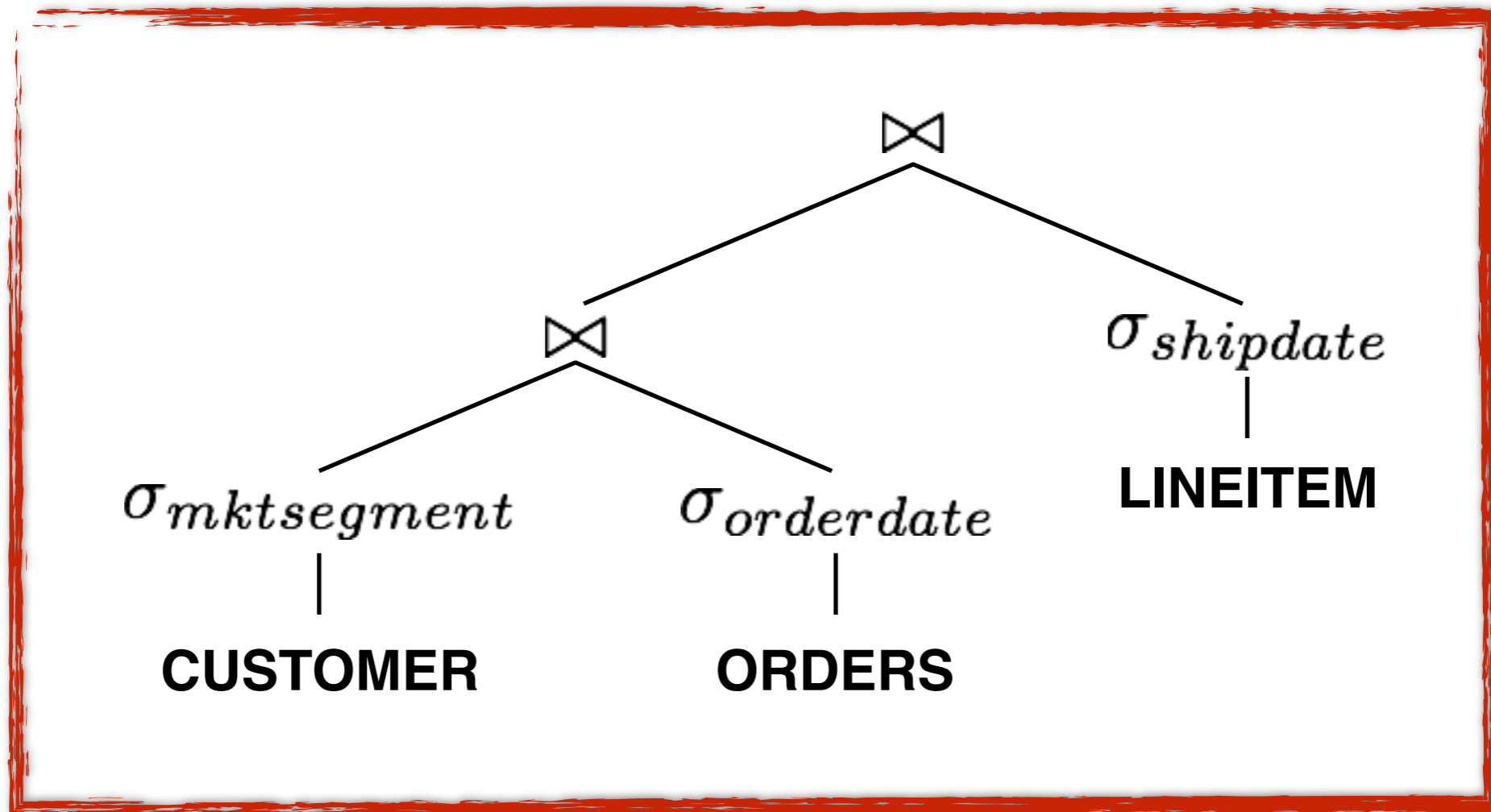
How about an example...

Delta Queries



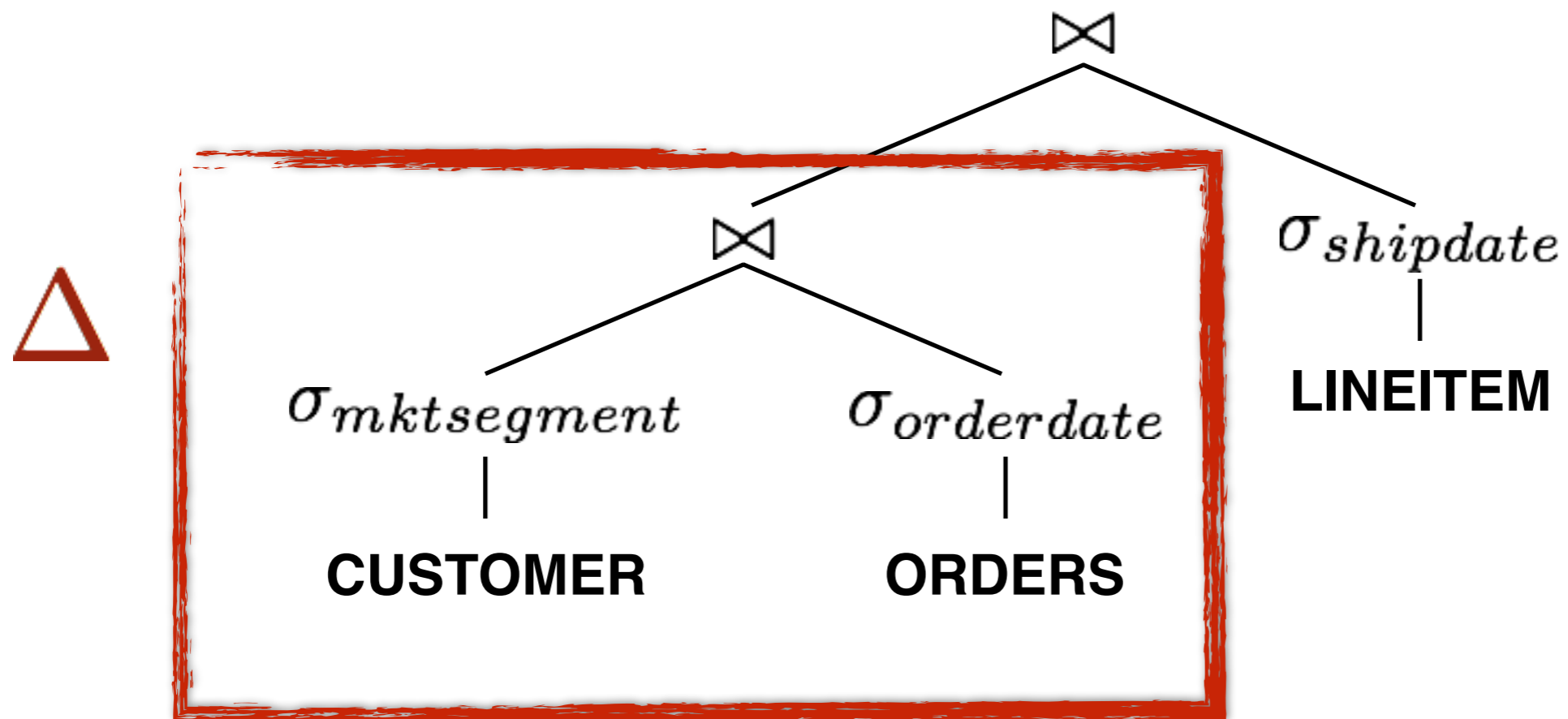
Let's say you have an insertion into LINEITEM

Delta Queries



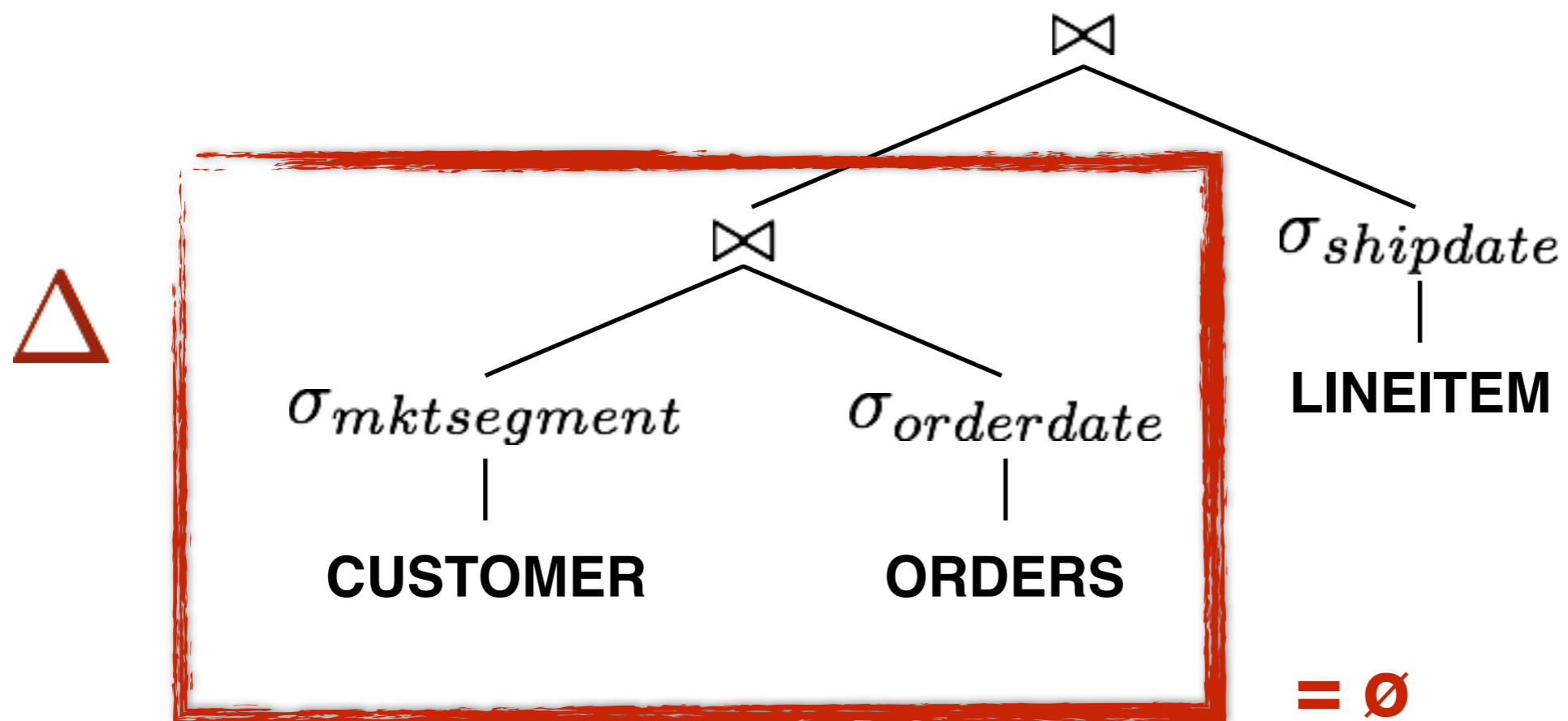
$$\Delta((\sigma(C) \bowtie \sigma(O)) \bowtie (\sigma(L)))$$

Delta Queries



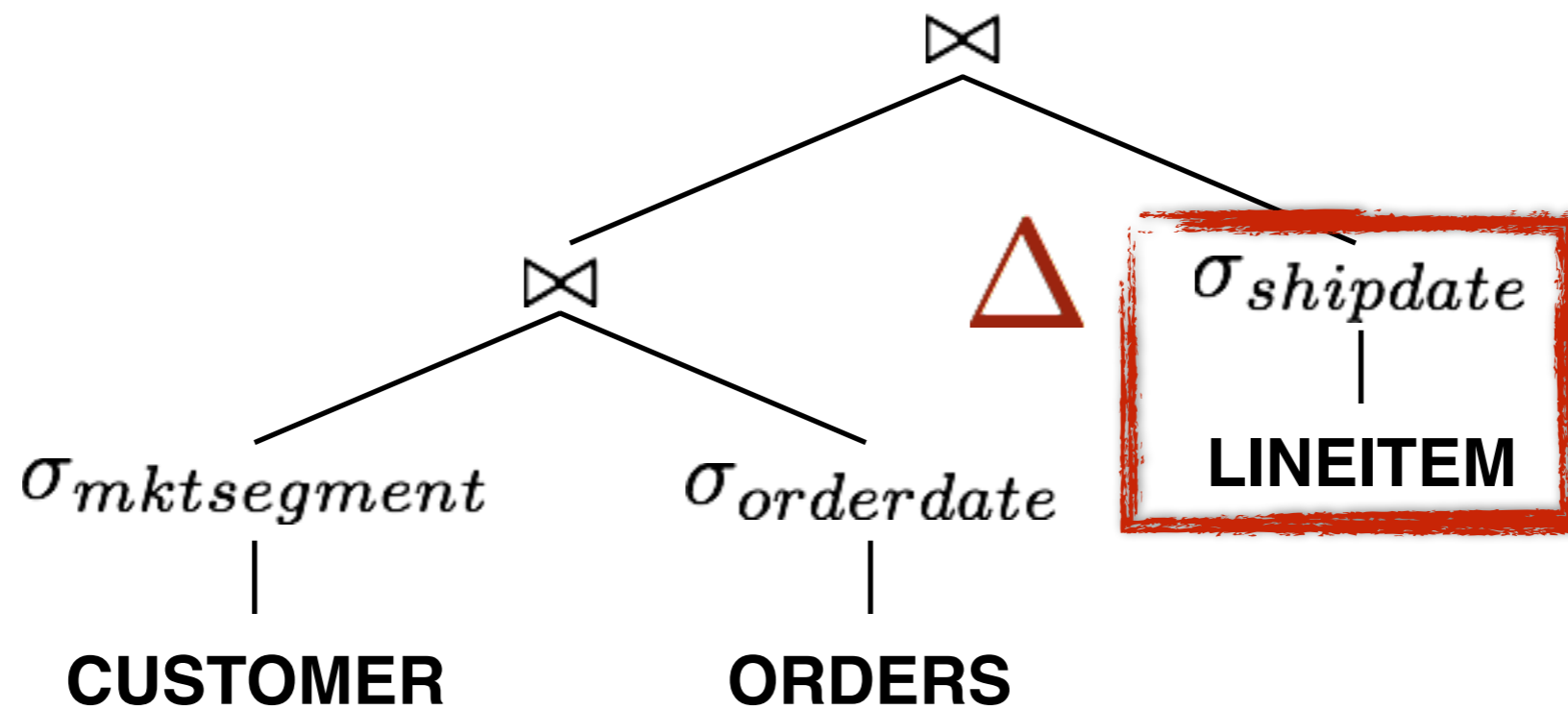
$$\Delta((\sigma(C) \bowtie \sigma(O)) \bowtie (\sigma(L)))$$

Delta Queries



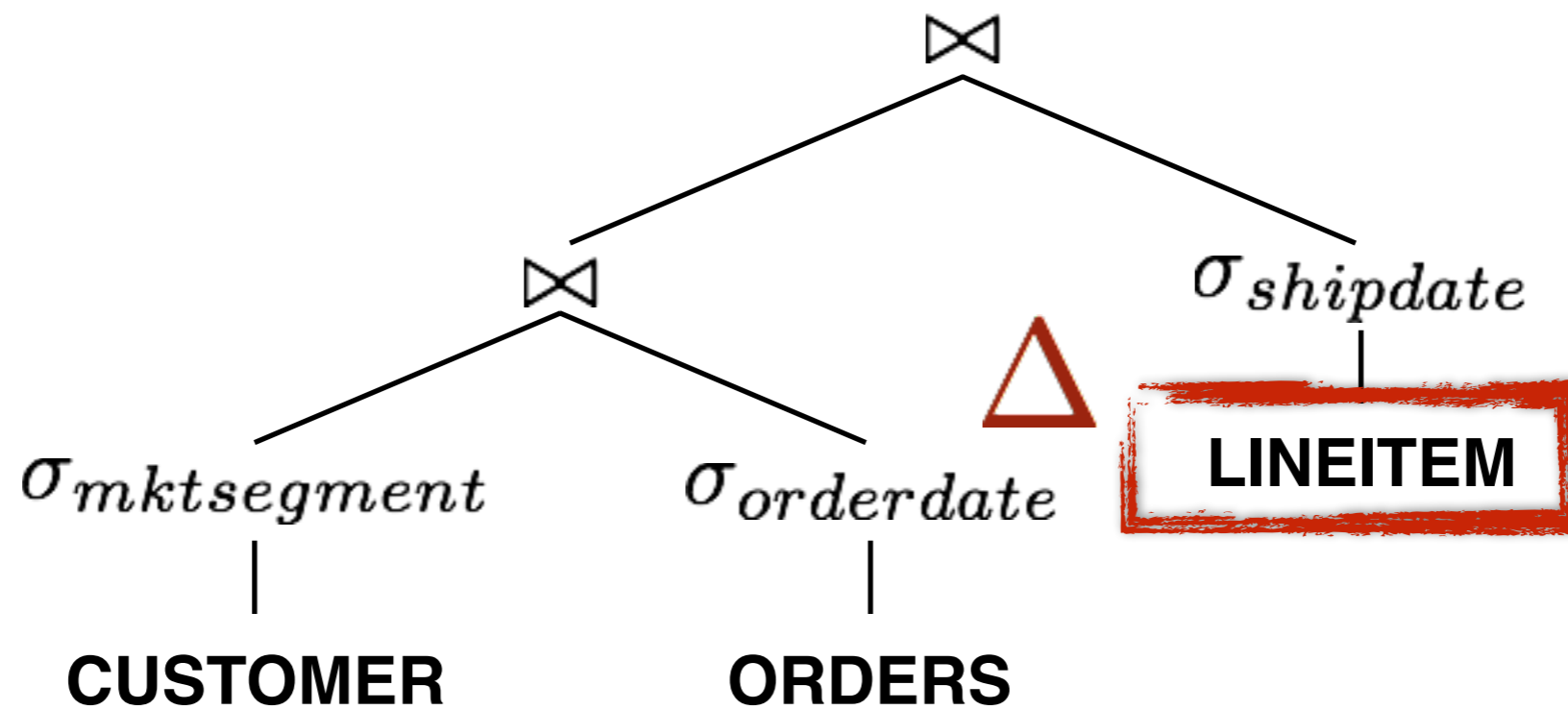
$$\Delta((\sigma(C) \bowtie \sigma(O)) \bowtie (\sigma(L)))$$

Delta Queries



$$((\sigma(C) \bowtie \sigma(O)) \bowtie \Delta(\sigma(L)))$$

Delta Queries



Delta Queries

```
SELECT *
FROM CUSTOMER C, ORDERS O, DELTA_LINEITEM DL
WHERE C.custkey = O.custkey
      AND DL.orderkey = O.orderkey
      AND C.mktsegment = ...
      AND O.orderdate = ...
      AND DL.shipdate = ...
```

Multisets

{ 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 5 }
(not compact)

{ 1 → x3, 2 → x5, 3 → x2, 4 → x6, 5 → x1 }

Multiset representation: Tuple → # of occurrences

Multisets

{ 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 4, 5 }
(not compact)

{ 1 → x3, 2 → x5, 3 → x2, 4 → x6, 5 → x1 }

Multiset representation: Tuple → ~~# of occurrences~~
multiplicity

Multiset Deltas

Insertions = Positive Multiplicity

Deletions = Negative Multiplicity

+ = Bag/Multiset Union