**Cats Precomputation**

1. Precomputation Table:

CREATE TABLE cats.pre\_WUsers AS

SELECT COUNT(\*) weight, lx.user\_id x\_user\_id, ly.user\_id y\_user\_id

FROM cats.like lx, cats.like ly

WHERE lx.video\_id = ly.video\_id

GROUP BY lx.user\_id, ly.user\_id;

**2. Trigger**

CREATE FUNCTION cats.Pre\_W\_Users\_Func() RETURNS trigger AS $Pre\_W\_Users\_Func$

BEGIN

CREATE TEMP TABLE users as

SELECT user\_id

FROM cats.like

WHERE video\_id = NEW.video\_id

UNION

SELECT NEW.user\_id;

WITH

U AS

(

SELECT user\_id

FROM users

)

UPDATE cats.Pre\_W\_Users

SET weight = weight + 1

WHERE

(

x\_user\_id IN

(

SELECT user\_id

FROM U

)

AND y\_user\_id = NEW.user\_id

)

OR

(

y\_user\_id IN

( SELECT user\_id FROM U) AND x\_user\_id = NEW.user\_id);

WITH U AS

(SELECT user\_id FROM users),

I as

(

SELECT u.user\_id x\_user\_id, NEW.user\_id y\_user\_id

FROM U

UNION

SELECT NEW.user\_id x\_user\_id , u.user\_id y\_user\_id

FROM U

UNION

SELECT NEW.user\_id x\_user\_id, NEW.user\_id y\_user\_id

),

X as

(

SELECT \*

FROM I

EXCEPT

SELECT x\_user\_id, y\_user\_id

FROM Cats.Pre\_W\_Users

)

INSERT INTO cats.Pre\_W\_Users

(x\_user\_id, y\_user\_id, weight)

SELECT \*, 1

FROM X;

RETURN NEW;

END;

$Pre\_W\_Users\_Func$ LANGUAGE plpgsql;

CREATE TRIGGER trigger\_like

BEFORE INSERT ON cats.like

FOR EACH ROW EXECUTE PROCEDURE cats.Pre\_W\_Users\_Func();

**3. Original Query Cost:**

WITH WeightOfUsers

AS

(

SELECT ly.user\_id, LOG(1 + COUNT(\*)) weight

FROM cats.like lx, cats.like ly

WHERE lx.user\_id = 1 AND lx.video\_id = ly.video\_id

GROUP BY ly.user\_id

)

SELECT l.video\_id, SUM(w.weight) sum\_weight

FROM cats.like l, WeightOfUsers w

WHERE l.user\_id = w.user\_id

GROUP BY l.video\_id

ORDER BY sum\_weight DESC LIMIT 10

**Limit (cost=228117.99..228118.02 rows=10)**

**4. New Query Cost:**

WITH W\_Users

AS

(

SELECT y\_user\_Id AS user\_id, log(sum(1 + weight)) weight

FROM cats.Pre\_WUsers

WHERE x\_user\_id = 49976

GROUP BY y\_user\_id

)

SELECT l.video\_id, SUM(w.weight) sum\_weight

FROM cats.LIKE l, W\_Users w

WHERE l.user\_id = w.user\_id

GROUP BY l.video\_id

ORDER BY sum\_weight desc limit 10

**cost=10661.91..10661.94**

### **5. Indices**

CREATE INDEX l\_vid\_uid

ON cats."like"

USING btree

(video\_id, user\_id);

CREATE INDEX pwu\_xy\_userid\_\_w

ON cats.pre\_WUsers

USING btree

(x\_user\_id, y\_user\_id, weight);

CREATE INDEX pwu\_userid

ON cats.pre\_WUsers

USING btree

(x\_user\_id);