**Cats Precomputation**

1. Precomputation Table:

CREATE TABLE cats.pre\_WUsers AS

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

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 or replace FUNCTION sales.Pre\_cat\_sale\_func() RETURNS trigger AS $Pre\_cat\_sale\_func$

BEGIN

UPDATE sales.Pre\_cat\_sale pre

SET dollar\_value = pre.dollar\_value + NEW.price

FROM sales.product p

WHERE pre.category\_id = p.category\_Id

AND p.product\_id = NEW.product\_id;

IF NOT FOUND THEN

INSERT INTO sales.Pre\_cat\_sale(category\_id, dollar\_value)

SELECT p.category\_id, NEW.price

FROM sales.product p

WHERE p.product\_id = NEW.product\_id;

END IF;

RETURN NEW;

END;

$Pre\_cat\_sale\_func$ LANGUAGE plpgsql;

CREATE TRIGGER trig\_Sale\_Cat

BEFORE INSERT ON sales.sale

FOR EACH ROW EXECUTE PROCEDURE sales.Pre\_cat\_sale\_func();

CREATE or replace FUNCTION sales.Pre\_cus\_sale\_func() RETURNS trigger AS $Pre\_cus\_sale\_func$

BEGIN

UPDATE sales.PRE\_Customer\_Sale

SET dollar\_value = dollar\_value + NEW.price

WHERE customer\_id = NEW.customer\_id;

IF NOT FOUND THEN

INSERT INTO sales.Pre\_cus\_sale(customer\_id, dollar\_value)

SELECT NEW.customer\_id, NEW.price;

END IF;

RETURN NEW;

END;

$Pre\_cus\_sale\_func$ LANGUAGE plpgsql;

CREATE TRIGGER trigger\_Sale\_Customer

BEFORE INSERT ON sales.sale

FOR EACH ROW EXECUTE PROCEDURE sales.Pre\_cus\_sale\_func();

**4. Original Query Cost:**

WITH WeightOfUsers

AS

(

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

**5. New Query Cost:**

WITH WeightOfUsers

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, WeightOfUsers 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**

### **6. 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);