**Sales Precomputation**

**Precomputation Table:**

CREAT TABLE sales.pre\_cat\_sale as

SELECT category\_id, sum(price) dollar\_value

FROM sales.category

NATURAL INNER JOIN sales.product

NATURAL INNER JOIN sales.sale

GROUP BY category\_id;

CREATE TABLE sales.pre\_cus\_sale as

SELECT customer\_id, sum(price) dollar\_value

FROM sales.sale

GROUP BY customer\_id;

**Origin query cost:**

CREATE VIEW top\_customer (customer\_id) AS

--reusing Q1:

SELECT customer\_id

FROM q1

ORDER BY dollar\_value DESC

LIMIT 20;

CREATE VIEW top\_category (category\_id) AS

--reusing Q5:

SELECT category\_id, SUM (dollar\_value) AS dollar\_value

FROM q5

GROUP BY category\_id

ORDER BY dollar\_value DESC

LIMIT 20;

CREATE VIEW q6 AS

--reusing Q4:

SELECT ca.category\_id, cu.customer\_id,

coalesce (SUM (q.quantity\_sold), 0) AS quantity\_sold,

coalesce (SUM (q.dollar\_value), 0.0) AS dollar\_value

FROM (top\_customer cu CROSS JOIN top\_category ca) LEFT JOIN q4 q

ON q.customer\_id = cu.customer\_id AND q.category\_id = ca.category\_id

GROUP BY ca.category\_id, cu.customer\_id;

Total query runtime: 7 min. 400 rows retrieved.

**New query Cost:**

CREATE VIEW top\_customer (customer\_id) AS

--reusing Q1:

SELECT customer\_id

FROM q1

ORDER BY dollar\_value DESC

LIMIT 20;

CREATE VIEW top\_category (category\_id) AS

--reusing Q5:

SELECT category\_id, SUM (dollar\_value) AS dollar\_value

FROM q5

GROUP BY category\_id

ORDER BY dollar\_value DESC

LIMIT 20;

CREATE VIEW q6 AS

--reusing Q4:

SELECT ca.category\_id, cu.customer\_id,

coalesce (SUM (q.quantity\_sold), 0) AS quantity\_sold,

coalesce (SUM (q.dollar\_value), 0.0) AS dollar\_value

FROM (top\_customer cu CROSS JOIN top\_category ca) LEFT JOIN q4 q

ON q.customer\_id = cu.customer\_id AND q.category\_id = ca.category\_id

GROUP BY ca.category\_id, cu.customer\_id;

Total query runtime: 7 min. 400 rows retrieved.

### **Indices**

CREATE INDEX pre\_cat\_sale\_dol\_cat

ON sales.pre\_cat\_sale

USING btree

(dollar\_value DESC, category\_id);

CREATE INDEX product\_cid\_pid

ON sales.product

USING btree

(category\_id, product\_id);

CREATE INDEX sale\_pid\_num\_price

ON sales.sale

USING btree

(product\_id, quantity, price);