CREATE or replace function ft\_insert\_sales()

RETURNS trigger

AS

$TRIG\_FUNC$

DECLARE

g\_name varchar(63);

g\_price numeric(12,2);

BEGIN

SELECT G.good\_name, G.good\_price\*NEW.sales\_qty INTO g\_name, g\_price FROM goods G where G.goods\_id = NEW.good\_id;

IF EXISTS(select from good\_sum\_mart T where T.good\_name = g\_name)

THEN UPDATE good\_sum\_mart T SET sum\_sale = sum\_sale + g\_price where T.good\_name = g\_name;

ELSE INSERT INTO good\_sum\_mart (good\_name, sum\_sale) values(g\_name, g\_price);

END IF;

RETURN NEW;

END;

$TRIG\_FUNC$

LANGUAGE plpgsql

VOLATILE

SET search\_path = pract\_functions, public

COST 50;

CREATE or replace function ft\_delete\_sales()

RETURNS trigger

AS

$TRIG\_FUNC$

DECLARE

g\_name varchar(63);

g\_price numeric(12,2);

BEGIN

SELECT G.good\_name, G.good\_price\*OLD.sales\_qty INTO g\_name, g\_price FROM goods G where G.goods\_id = OLD.good\_id;

IF EXISTS(select from good\_sum\_mart T where T.good\_name = g\_name)

THEN

UPDATE good\_sum\_mart T SET sum\_sale = sum\_sale - g\_price where T.good\_name = g\_name;

DELETE FROM good\_sum\_mart T where T.good\_name = g\_name and (sum\_sale < 0 or sum\_sale = 0);

END IF;

RETURN NEW;

END;

$TRIG\_FUNC$

LANGUAGE plpgsql

VOLATILE

SET search\_path = pract\_functions, public

COST 50;

CREATE or replace function ft\_update\_sales()

RETURNS trigger

AS

$TRIG\_FUNC$

DECLARE

g\_name\_old varchar(63);

g\_price\_old numeric(12,2);

g\_name\_new varchar(63);

g\_price\_new numeric(12,2);

BEGIN

SELECT G.good\_name, G.good\_price\*OLD.sales\_qty INTO g\_name\_old, g\_price\_old FROM goods G where G.goods\_id = OLD.good\_id;

SELECT G.good\_name, G.good\_price\*NEW.sales\_qty INTO g\_name\_new, g\_price\_new FROM goods G where G.goods\_id = NEW.good\_id;

IF EXISTS(select from good\_sum\_mart T where T.good\_name = g\_name\_new)

THEN UPDATE good\_sum\_mart T SET sum\_sale = sum\_sale + g\_price\_new where T.good\_name = g\_name\_new;

ELSE INSERT INTO good\_sum\_mart (good\_name, sum\_sale) values(g\_name\_new, g\_price\_new);

END IF;

IF EXISTS(select from good\_sum\_mart T where T.good\_name = g\_name\_old)

THEN

UPDATE good\_sum\_mart T SET sum\_sale = sum\_sale - g\_price\_old where T.good\_name = g\_name\_old;

DELETE FROM good\_sum\_mart T where T.good\_name = g\_name\_old and (sum\_sale < 0 or sum\_sale = 0);

END IF;

RETURN NEW;

END;

$TRIG\_FUNC$

LANGUAGE plpgsql

VOLATILE

SET search\_path = pract\_functions, public

COST 50;

CREATE TRIGGER tr\_insert\_sales

AFTER INSERT

ON sales

FOR EACH ROW

EXECUTE PROCEDURE ft\_insert\_sales();

CREATE TRIGGER tr\_delete\_sales

AFTER DELETE

ON sales

FOR EACH ROW

EXECUTE PROCEDURE ft\_delete\_sales();

CREATE TRIGGER tr\_update\_sales

AFTER UPDATE

ON sales

FOR EACH ROW

EXECUTE PROCEDURE ft\_update\_sales();