```
********* QUERY **********
DROP TABLE IF EXISTS new_tags CASCADE;
**************************

DROP TABLE
********* QUERY **********
DROP TABLE IF EXISTS a_tags CASCADE;
**************************

DROP TABLE
********* QUERY **********
DROP TABLE IF EXISTS b_tags CASCADE;
**************************

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS new_tags AS
SELECT * FROM tags LIMIT 0;
**************************

SELECT 0
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS a_tags AS
SELECT * FROM tags LIMIT 0;
**************************

SELECT 0
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS b_tags AS
SELECT * FROM tags LIMIT 0;
**************************

SELECT 0
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
DROP FUNCTION IF EXISTS insert_a_tags;
**************************
```

```
DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION insert_a_tags() RETURNS TRIGGER AS
$$
        BEGIN
                IF lower(substring(NEW.tag FROM 1 FOR 1)) = 'a' THEN
                        INSERT INTO a_tags (pk, tag, parent)
                        VALUES (NEW.pk, NEW.tag, NEW.parent);
                END IF;
                RETURN NEW;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER trigger_a_tags BEFORE INSERT ON new_tags
FOR EACH ROW EXECUTE PROCEDURE insert_a_tags();
**************************

CREATE TRIGGER
********* QUERY **********
INSERT INTO new_tags (pk, tag, parent)
VALUES
        (1, 'fruits', null),
        (2, 'apple', 1);
**************************

INSERT 0 2
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |
  2 | apple  |      1
(2 rows)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk |  tag  | parent
----+-------+--------
  2 | apple |      1
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS insert_b_tags;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION insert_b_tags() RETURNS TRIGGER AS
$$
        BEGIN
                IF lower(left(NEW.tag, 1)) = 'b' THEN
```

```
                        INSERT INTO b_tags (pk, tag, parent)
                        VALUES (NEW.pk, NEW.tag, NEW.parent);
                        RETURN NULL;
                END IF;
                RETURN NEW;
        END;
$$
LANGUAGE 'plpgsql';
***************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER trigger_b_tags BEFORE INSERT ON new_tags
FOR EACH ROW EXECUTE PROCEDURE insert_b_tags();
***************************

CREATE TRIGGER
********* QUERY **********
INSERT INTO new_tags (pk, tag, parent)
VALUES (3, 'banana', 1);
***************************

INSERT 0 0
********* QUERY **********
SELECT * FROM new_tags;
***************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |
  2 | apple  |      1
(2 rows)

********* QUERY **********
SELECT * FROM b_tags;
***************************

 pk |  tag   | parent
----+--------+--------
  3 | banana |      1
(1 row)

********* QUERY **********
TRUNCATE new_tags;
***************************

TRUNCATE TABLE
********* QUERY **********
TRUNCATE a_tags;
***************************

TRUNCATE TABLE
********* QUERY **********
TRUNCATE b_tags;
***************************

TRUNCATE TABLE
********* QUERY **********
DROP TRIGGER IF EXISTS trigger_a_tags ON new_tags;
***************************
```

```
DROP TRIGGER
********* QUERY **********
DROP TRIGGER IF EXISTS trigger_b_tags ON new_tags;
**************************

DROP TRIGGER
********* QUERY **********
CREATE OR REPLACE FUNCTION insert_tags() RETURNS TRIGGER AS
$$
        BEGIN
                IF lower(substring(NEW.tag FROM 1 FOR 1)) = 'a' THEN
                        INSERT INTO a_tags (pk, tag, parent)
                        VALUES (NEW.pk, NEW.tag, NEW.parent);
                        RETURN NEW;
                ELSIF lower(left(NEW.tag, 1)) = 'b' THEN
                        INSERT INTO b_tags (pk, tag, parent)
                        VALUES (NEW.pk, NEW.tag, NEW.parent);
                        RETURN NULL;
                ELSE
                        RETURN NEW;
                END IF;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER trigger_tags BEFORE INSERT ON new_tags
FOR EACH ROW EXECUTE PROCEDURE insert_tags();
**************************

CREATE TRIGGER
********* QUERY **********
INSERT INTO new_tags
VALUES
        (1, 'fruits', null),
        (2, 'apple', 1),
        (3, 'banana', 1);
**************************

INSERT 0 2
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |
  2 | apple  |      1
(2 rows)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk |  tag  | parent
----+-------+--------
  2 | apple |      1
```

```
(1 row)

********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  3 | banana |      1
(1 row)

********* QUERY **********
DROP TABLE IF EXISTS new_tags CASCADE;
**************************

DROP TABLE
********* QUERY **********
DROP TABLE IF EXISTS a_tags CASCADE;
**************************

DROP TABLE
********* QUERY **********
DROP TABLE IF EXISTS b_tags CASCADE;
**************************

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS new_tags AS
SELECT * FROM tags LIMIT 0;
**************************

SELECT 0
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS a_tags AS
SELECT * FROM tags LIMIT 0;
**************************

SELECT 0
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS b_tags AS
SELECT * FROM tags LIMIT 0;
**************************

SELECT 0
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)
```

```
********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
DROP FUNCTION IF EXISTS copy_tags;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION copy_tags() RETURNS TRIGGER AS
$$
        BEGIN
                IF TG_OP = 'INSERT' THEN
                        IF lower(substring(NEW.tag FROM 1 FOR 1)) = 'a' THE
N
                                INSERT INTO a_tags
                                VALUES (NEW.pk, NEW.tag, NEW.parent);
                        ELSIF lower(left(NEW.tag, 1)) = 'b' THEN
                                INSERT INTO b_tags
                                VALUES (NEW.pk, NEW.tag, NEW.parent);
                        END IF;
                        RETURN NEW;
                END IF;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER copy_tags_insert BEFORE INSERT ON new_tags
FOR EACH ROW EXECUTE PROCEDURE copy_tags();
**************************

CREATE TRIGGER
********* QUERY **********
INSERT INTO new_tags
VALUES
        (1, 'fruits', null),
        (2, 'apple', 1),
        (3, 'banana', 1);
**************************

INSERT 0 3
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |
  2 | apple  |      1
  3 | banana |      1
(3 rows)
```

```
********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk |  tag  | parent
----+-------+--------
  2 | apple |      1
(1 row)

********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  3 | banana |      1
(1 row)

********* QUERY **********
CREATE OR REPLACE FUNCTION copy_tags() RETURNS TRIGGER AS
$$
        BEGIN
                IF TG_OP = 'INSERT' THEN
                        IF lower(left(NEW.tag, 1)) = 'a' THEN
                                INSERT INTO a_tags
                                VALUES (NEW.pk, NEW.tag, NEW.parent);
                        ELSIF lower(substring(NEW.tag FROM 1 FOR 1)) = 'b'
THEN
                                INSERT INTO b_tags
                                VALUES (NEW.pk, NEW.tag, NEW.parent);
                        END IF;
                        RETURN NEW;
                END IF;
                IF TG_OP = 'DELETE' THEN
                        IF lower(substring(OLD.tag FROM 1 FOR 1)) = 'a' THE
N
                                DELETE FROM a_tags WHERE pk = OLD.pk;
                        ELSIF lower(left(OLD.tag, 1)) = 'b' THEN
                                DELETE FROM b_tags WHERE pk = OLD.pk;
                        END IF;
                        RETURN OLD;
                END IF;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER copy_tags_delete AFTER DELETE ON new_tags
FOR EACH ROW EXECUTE PROCEDURE copy_tags();
**************************

CREATE TRIGGER
********* QUERY **********
DELETE FROM new_tags WHERE pk IN (2, 3);
**************************

DELETE 2
********* QUERY **********
```

```
SELECT * FROM new_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |
(1 row)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
DROP TRIGGER copy_tags_insert ON new_tags CASCADE;
**************************

DROP TRIGGER
********* QUERY **********
DROP TRIGGER copy_tags_delete ON new_tags CASCADE;
**************************

DROP TRIGGER
********* QUERY **********
TRUNCATE new_tags;
**************************

TRUNCATE TABLE
********* QUERY **********
TRUNCATE a_tags;
**************************

TRUNCATE TABLE
********* QUERY **********
TRUNCATE b_tags;
**************************

TRUNCATE TABLE
********* QUERY **********
CREATE OR REPLACE FUNCTION copy_tags() RETURNS TRIGGER AS
$$
        BEGIN
                IF TG_OP = 'INSERT' THEN
                        IF lower(left(NEW.tag, 1)) = 'a' THEN
                                INSERT INTO a_tags
                                VALUES (NEW.pk, NEW.tag, NEW.parent);
                        ELSIF lower(left(NEW.tag, 1)) = 'b' THEN
                                INSERT INTO b_tags
                                VALUES (NEW.pk, NEW.tag, NEW.parent);
```

```
                        END IF;
                        RETURN NEW;
                END IF;
                IF TG_OP = 'DELETE' THEN
                        IF lower(left(OLD.tag, 1)) = 'a' THEN
                                DELETE FROM a_tags WHERE OLD.pk = pk;
                        ELSIF lower(left(OLD.tag, 1)) = 'b' THEN
                                DELETE FROM b_tags WHERE OLD.pk = pk;
                        END IF;
                        RETURN OLD;
                END IF;
                IF TG_OP = 'UPDATE' THEN
                        IF lower(left(OLD.tag, 1)) IN ('a', 'b') THEN
                                DELETE FROM a_tags WHERE OLD.pk = pk;
                                DELETE FROM b_tags WHERE OLD.pk = pk;
                                DELETE FROM new_tags WHERE OLD.pk = pk;
                                INSERT INTO new_tags VALUES (NEW.pk, NEW.ta
g, NEW.parent);
                        END IF;
                        RETURN NEW;
                END IF;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER copy_tags_insert BEFORE INSERT ON new_tags
FOR EACH ROW EXECUTE PROCEDURE copy_tags();
**************************

CREATE TRIGGER
********* QUERY **********
CREATE TRIGGER copy_tags_delete AFTER DELETE ON new_tags
FOR EACH ROW EXECUTE PROCEDURE copy_tags();
**************************

CREATE TRIGGER
********* QUERY **********
CREATE TRIGGER copy_tags_update AFTER UPDATE ON new_tags
FOR EACH ROW EXECUTE PROCEDURE copy_tags();
**************************

CREATE TRIGGER
********* QUERY **********
INSERT INTO new_tags
VALUES
        (1, 'fruits', null),
        (2, 'apple', 1),
        (3, 'banana', 1);
**************************

INSERT 0 3
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk |  tag   | parent
----+--------+--------
```

```
  1 | fruits |
  2 | apple  |      1
  3 | banana |      1
(3 rows)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk |  tag  | parent
----+-------+--------
  2 | apple |      1
(1 row)

********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  3 | banana |      1
(1 row)

********* QUERY **********
UPDATE new_tags
SET tag = 'apricot'
WHERE pk = 3;
**************************

UPDATE 1
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk |   tag   | parent
----+---------+--------
  1 | fruits  |
  2 | apple   |      1
  3 | apricot |      1
(3 rows)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk |   tag   | parent
----+---------+--------
  2 | apple   |      1
  3 | apricot |      1
(2 rows)

********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk | tag | parent
----+-----+--------
(0 rows)

********* QUERY **********
```

```
UPDATE new_tags
SET tag = 'banana'
WHERE pk = 3;
**************************

UPDATE 1
********* QUERY **********
SELECT * FROM new_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |
  2 | apple  |      1
  3 | banana |      1
(3 rows)

********* QUERY **********
SELECT * FROM a_tags;
**************************

 pk |  tag  | parent
----+-------+--------
  2 | apple |      1
(1 row)

********* QUERY **********
SELECT * FROM b_tags;
**************************

 pk |  tag   | parent
----+--------+--------
  3 | banana |      1
(1 row)
```