

***** QUERY *****

```
CREATE TEMP TABLE IF NOT EXISTS temp_tags AS
SELECT * FROM tags;
```

SELECT 3

***** QUERY *****

```
SELECT * FROM temp_tags;
```

pk	tag	parent
1	fruits	NULL
2	vegetables	NULL
4	apple	1

(3 rows)

***** QUERY *****

```
UPDATE temp_tags
SET tag = 'orange'
WHERE pk = 4;
```

UPDATE 1

***** QUERY *****

```
SELECT * FROM temp_tags;
```

pk	tag	parent
1	fruits	NULL
2	vegetables	NULL
4	orange	1

(3 rows)

***** QUERY *****

```
SELECT * FROM temp_tags;
```

pk	tag	parent
1	fruits	NULL
2	vegetables	NULL
4	orange	1

(3 rows)

***** QUERY *****

```
CREATE TEMP TABLE IF NOT EXISTS a_tags (
    pk integer NOT NULL PRIMARY KEY,
    tag text,
    parent integer);
```

CREATE TABLE

Table "pg_temp_3.a_tags"				
Column	Type	Collation	Nullable	Default
pk	integer		not null	
tag	text			
parent	integer			

Indexes:

"a_tags_pkey" PRIMARY KEY, btree (pk)

***** QUERY *****

```
CREATE OR REPLACE RULE r_tags1 AS ON INSERT TO temp_tags
WHERE NEW.tag ILIKE 'a%'
DO ALSO
```

```
    INSERT INTO a_tags (pk, tag, parent)
    VALUES (NEW.pk, NEW.tag, NEW.parent);
```

CREATE RULE

***** QUERY *****

```
INSERT INTO temp_tags (pk, tag)
VALUES (11, 'apple');
```

INSERT 0 1

***** QUERY *****

```
SELECT * FROM temp_tags;
```

pk	tag	parent
1	fruits	NULL
2	vegetables	NULL
4	orange	1
11	apple	NULL

(4 rows)

***** QUERY *****

```
SELECT * FROM a_tags;
```

pk	tag	parent
11	apple	NULL

(1 row)

***** QUERY *****

```
CREATE TEMP TABLE IF NOT EXISTS b_tags (
    pk integer NOT NULL PRIMARY KEY,
    tag text,
    parent integer);
```

CREATE TABLE

Table "pg_temp_3.b_tags"				
Column	Type	Collation	Nullable	Default
pk	integer		not null	
tag	text			
parent	integer			

Indexes:

"b_tags_pkey" PRIMARY KEY, btree (pk)

***** QUERY *****

```
CREATE OR REPLACE RULE r_tags2 AS ON INSERT TO temp_tags
WHERE NEW.tag ILIKE 'b%'
DO INSTEAD
```

```

INSERT INTO b_tags (pk, tag, parent)
VALUES (NEW.pk, NEW.tag, NEW.parent);
*****

```

```

CREATE RULE
***** QUERY *****
INSERT INTO temp_tags (pk, tag)
VALUES (12, 'banana');
*****

```

```

INSERT 0 0
***** QUERY *****
SELECT * FROM temp_tags;
*****

```

pk	tag	parent
1	fruits	NULL
2	vegetables	NULL
4	orange	1
11	apple	NULL

(4 rows)

```

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

```

pk	tag	parent
12	banana	NULL

(1 row)

```

***** QUERY *****
CREATE OR REPLACE RULE r_tags3 AS ON INSERT TO temp_tags
WHERE NEW.tag ILIKE 'c%'
DO INSTEAD NOTHING;
*****

```

```

CREATE RULE
***** QUERY *****
INSERT INTO temp_tags (pk, tag)
VALUES (13, 'cedor');
*****

```

```

INSERT 0 0
***** QUERY *****
SELECT pk, tag, parent, 'tags' AS tablename FROM temp_tags
UNION ALL
SELECT pk, tag, parent, 'a_tags' AS tablename FROM a_tags
UNION ALL
SELECT pk, tag, parent, 'b_tags' AS tablename FROM b_tags
ORDER BY tablename, tag;
*****

```

pk	tag	parent	tablename
11	apple	NULL	a_tags
12	banana	NULL	b_tags
11	apple	NULL	tags
1	fruits	NULL	tags

4	orange	1	tags
2	vegetables	NULL	tags

(6 rows)

***** QUERY *****

DROP TABLE IF EXISTS new_tags, new_a_tags, new_b_tags;

DROP TABLE

***** QUERY *****

CREATE TEMP TABLE IF NOT EXISTS new_tags AS

SELECT * FROM tags LIMIT 0;

SELECT 0

***** QUERY *****

ALTER TABLE new_tags

ALTER COLUMN pk SET NOT NULL;

ALTER TABLE

***** QUERY *****

ALTER TABLE new_tags

ADD CONSTRAINT new_tags_pk PRIMARY KEY (pk);

ALTER TABLE

***** QUERY *****

CREATE TEMP TABLE IF NOT EXISTS new_a_tags AS

SELECT * FROM tags LIMIT 0;

SELECT 0

***** QUERY *****

ALTER TABLE new_a_tags

ALTER COLUMN pk SET NOT NULL;

ALTER TABLE

***** QUERY *****

ALTER TABLE new_a_tags

ADD CONSTRAINT new_a_tags_pk PRIMARY KEY (pk);

ALTER TABLE

***** QUERY *****

CREATE TEMP TABLE IF NOT EXISTS new_b_tags AS

SELECT * FROM tags LIMIT 0;

SELECT 0

***** QUERY *****

ALTER TABLE new_b_tags

ALTER COLUMN pk SET NOT NULL;

ALTER TABLE

***** QUERY *****

ALTER TABLE new_b_tags

ADD CONSTRAINT new_b_tags_pk PRIMARY KEY (pk);

ALTER TABLE

***** QUERY *****

DROP RULE IF EXISTS r_insert_a ON new_tags;

DROP RULE

***** QUERY *****

DROP RULE IF EXISTS r_insert_b ON new_tags;

DROP RULE

***** QUERY *****

CREATE OR REPLACE RULE r_insert_a AS ON INSERT TO new_tags
WHERE NEW.tag ILIKE 'a%'

DO ALSO

INSERT INTO new_a_tags (pk, tag, parent)
VALUES (NEW.pk, NEW.tag, NEW.parent);

CREATE RULE

***** QUERY *****

CREATE OR REPLACE RULE r_insert_b AS ON INSERT TO new_tags
WHERE NEW.tag ILIKE 'b%'

DO ALSO

INSERT INTO new_b_tags (pk, tag, parent)
VALUES (NEW.pk, NEW.tag, NEW.parent);

CREATE RULE

***** QUERY *****

INSERT INTO new_tags

VALUES

(1, 'fruits', NULL),
(2, 'apple', 1),
(3, 'orange', 1),
(4, 'banana', 1);

INSERT 0 4

***** QUERY *****

SELECT * FROM new_tags;

pk	tag	parent
1	fruits	
2	apple	1
3	orange	1
4	banana	1

(4 rows)

***** QUERY *****

SELECT * FROM new_a_tags;

pk	tag	parent
2	apple	1

(1 row)

```
***** QUERY *****
SELECT * FROM new_b_tags;
*****
```

pk	tag	parent
4	banana	1

(1 row)

```
***** QUERY *****
DROP RULE IF EXISTS r_delete_a ON new_tags;
*****
```

```
DROP RULE
***** QUERY *****
DROP RULE IF EXISTS r_delete_b ON new_tags;
*****
```

```
DROP RULE
***** QUERY *****
CREATE OR REPLACE RULE r_delete_a AS ON DELETE TO new_tags
WHERE OLD.tag ILIKE 'a%'
DO ALSO
    DELETE FROM new_a_tags WHERE OLD.pk = pk;
*****
```

```
CREATE RULE
***** QUERY *****
CREATE OR REPLACE RULE r_delete_b AS ON DELETE TO new_tags
WHERE OLD.tag ILIKE 'b%'
DO ALSO
    DELETE FROM new_b_tags WHERE OLD.pk = pk;
*****
```

```
CREATE RULE
***** QUERY *****
DELETE FROM new_tags WHERE tag = 'apple';
*****
```

```
DELETE 1
***** QUERY *****
DELETE FROM new_tags WHERE tag = 'banana';
*****
```

```
DELETE 1
***** QUERY *****
SELECT * FROM new_tags;
*****
```

pk	tag	parent
1	fruits	
3	orange	1

(2 rows)

```
***** QUERY *****
SELECT * FROM new_a_tags;
*****
```

```

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

```

```

***** QUERY *****
SELECT * FROM new_b_tags;
*****

```

```

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

```

```

***** QUERY *****
DROP FUNCTION IF EXISTS move_record;
*****

```

```

DROP FUNCTION
***** QUERY *****
CREATE OR REPLACE FUNCTION move_record (
    new_pk integer,
    new_tag text,
    new_parent integer,
    old_pk integer,
    old_tag text) RETURNS VOID AS
$$
    BEGIN
        IF left(lower(new_tag), 1) IN ('a', 'b') THEN
            DELETE FROM new_tags WHERE pk = OLD_pk;
            INSERT INTO new_tags VALUES (new_pk, new_tag, new_p
arent);
        END IF;
    END;
$$
LANGUAGE 'plpgsql';
*****

```

```

CREATE FUNCTION
***** QUERY *****
DROP RULE IF EXISTS r_insert ON new_tags;
*****

```

```

DROP RULE
***** QUERY *****
CREATE OR REPLACE RULE r_insert AS ON UPDATE TO new_tags
DO ALSO
    SELECT move_record(
        NEW.pk,
        NEW.tag,
        NEW.parent,
        OLD.pk,
        OLD.tag);
*****

```

```

CREATE RULE
***** QUERY *****
UPDATE new_tags SET tag = 'apple' WHERE tag = 'orange';
*****

```

```

move_record

```

(1 row)

UPDATE 0

***** QUERY *****

SELECT * FROM new_tags;

pk	tag	parent
1	fruits	
3	apple	1

(2 rows)

***** QUERY *****

SELECT * FROM new_a_tags;

pk	tag	parent
3	apple	1

(1 row)

***** QUERY *****

SELECT * FROM new_b_tags;

pk	tag	parent
----	-----	--------

(0 rows)

***** QUERY *****

UPDATE new_tags SET tag = 'banana' WHERE tag = 'apple';

move_record

(1 row)

UPDATE 0

***** QUERY *****

SELECT * FROM new_tags;

pk	tag	parent
1	fruits	
3	banana	1

(2 rows)

***** QUERY *****

SELECT * FROM new_a_tags;

pk	tag	parent
----	-----	--------

(0 rows)


```
***** QUERY *****
SELECT * FROM new_b_tags;
*****
```

pk	tag	parent
3	banana	1

(1 row)

```
***** QUERY *****
UPDATE new_tags SET tag = 'apple' WHERE tag = 'banana';
*****
```

```
move_record
-----
```

(1 row)

```
UPDATE 0
***** QUERY *****
SELECT * FROM new_tags;
*****
```

pk	tag	parent
1	fruits	
3	apple	1

(2 rows)

```
***** QUERY *****
SELECT * FROM new_a_tags;
*****
```

pk	tag	parent
3	apple	1

(1 row)

```
***** QUERY *****
SELECT * FROM new_b_tags;
*****
```

pk	tag	parent
----	-----	--------

(0 rows)