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

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags (
        pk integer GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
        tag text,
        level integer DEFAULT 0);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE part_tags_level_0 (CHECK (level = 0)) INHERITS (part_tags
);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE part_tags_level_1 (CHECK (level = 1)) INHERITS (part_tags
);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE part_tags_level_2 (CHECK (level = 2)) INHERITS (part_tags
);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE part_tags_level_3 (CHECK (level = 3)) INHERITS (part_tags
);
**************************

CREATE TABLE
                                          Table "pg_temp_3.part_tags"
 Column |  Type   | Collation | Nullable |           Default            | S
torage  | Stats target | Description
--------+---------+-----------+----------+------------------------------+--
--------+--------------+-------------
 pk     | integer |           | not null | generated always as identity | p
lain    |              |
 tag    | text    |           |          |                              | e
xtended |              |
 level  | integer |           |          | 0                            | p
lain    |              |
Indexes:
    "part_tags_pkey" PRIMARY KEY, btree (pk)
Child tables: part_tags_level_0,
              part_tags_level_1,
              part_tags_level_2,
              part_tags_level_3
Access method: heap

********* QUERY **********
ALTER TABLE part_tags_level_0
ADD CONSTRAINT part_tags_level_0_pkey PRIMARY KEY (pk);
**************************
```

```
ALTER TABLE
********* QUERY **********
ALTER TABLE part_tags_level_1
ADD CONSTRAINT part_tags_level_1_pkey PRIMARY KEY (pk);
**************************

ALTER TABLE
********* QUERY **********
ALTER TABLE part_tags_level_2
ADD CONSTRAINT part_tags_level__pkey PRIMARY KEY (pk);
**************************

ALTER TABLE
********* QUERY **********
ALTER TABLE part_tags_level_3
ADD CONSTRAINT part_tags_level_3_pkey PRIMARY KEY (pk);
**************************

ALTER TABLE
********* QUERY **********
CREATE INDEX part_tags_level_0_tag ON part_tags_level_0 USING GIN (tag gin_
trgm_ops);
**************************

CREATE INDEX
********* QUERY **********
CREATE INDEX part_tags_level_1_tag ON part_tags_level_1 USING GIN (tag gin_
trgm_ops);
**************************

CREATE INDEX
********* QUERY **********
CREATE INDEX part_tags_level_2_tag ON part_tags_level_2 USING GIN (tag gin_
trgm_ops);
**************************

CREATE INDEX
********* QUERY **********
CREATE INDEX part_tags_level_3_tag ON part_tags_level_3 USING GIN (tag gin_
trgm_ops);
**************************

CREATE INDEX
********* QUERY **********
CREATE OR REPLACE FUNCTION insert_part_tags() RETURNS trigger AS
$$
        BEGIN
                IF NEW.level = 0 THEN
                        INSERT INTO part_tags_level_0 VALUES (NEW.*);
                ELSIF NEW.level = 1 THEN
                        INSERT INTO part_tags_level_1 VALUES (NEW.*);
                ELSIF NEW.level = 2 THEN
                        INSERT INTO part_tags_level_2 VALUES (NEW.*);
                ELSIF NEW.level = 3 THEN
                        INSERT INTO part_tags_level_3 VALUES (NEW.*);
                ELSE
                        RAISE EXCEPTION 'Error in part_tags, level out of r
ange';
                END IF;
                RETURN NULL;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER insert_part_tags_trigger BEFORE INSERT ON part_tags
FOR EACH ROW EXECUTE PROCEDURE insert_part_tags();
**************************

CREATE TRIGGER
********* QUERY **********
INSERT INTO part_tags (tag, level)
VALUES
        ('vegetables', 0),
        ('fruits', 0),
        ('orange', 1),
        ('apple', 1),
        ('red apple', 2);
**************************

INSERT 0 0
********* QUERY **********
SELECT * FROM part_tags;
**************************

 pk |    tag     | level
----+------------+-------
  1 | vegetables |     0
  2 | fruits     |     0
  3 | orange     |     1
  4 | apple      |     1
  5 | red apple  |     2
(5 rows)

********* QUERY **********
SELECT * FROM ONLY part_tags;
**************************

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

********* QUERY **********
SELECT * FROM part_tags_level_0;
**************************

 pk |    tag     | level
----+------------+-------
  1 | vegetables |     0
  2 | fruits     |     0
(2 rows)

********* QUERY **********
SELECT * FROM part_tags_level_1;
**************************

 pk |  tag   | level
```

```
----+--------+-------
  3 | orange |     1
  4 | apple  |     1
(2 rows)

********* QUERY **********
SELECT * FROM part_tags_level_2;
**************************

 pk |    tag    | level
----+-----------+-------
  5 | red apple |     2
(1 row)

********* QUERY **********
DELETE FROM part_tags WHERE tag = 'apple';
**************************

DELETE 1
********* QUERY **********
SELECT * FROM part_tags;
**************************

 pk |    tag     | level
----+------------+-------
  1 | vegetables |     0
  2 | fruits     |     0
  3 | orange     |     1
  5 | red apple  |     2
(4 rows)

********* QUERY **********
SELECT * FROM part_tags_level_1;
**************************

 pk |  tag   | level
----+--------+-------
  3 | orange |     1
(1 row)

********* QUERY **********
UPDATE part_tags
SET tag = 'apple'
WHERE pk = 3;
**************************

UPDATE 1
********* QUERY **********
SELECT * FROM part_tags;
**************************

 pk |    tag     | level
----+------------+-------
  1 | vegetables |     0
  2 | fruits     |     0
  3 | apple      |     1
  5 | red apple  |     2
(4 rows)

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

```
SELECT * FROM part_tags_level_1;
*************************

 pk |  tag  | level
----+-------+-------
  3 | apple |     1
(1 row)

********* QUERY **********
CREATE OR REPLACE FUNCTION update_part_tags() RETURNS trigger AS
$$
        BEGIN
                IF NEW.level <> OLD.level THEN
                        DELETE FROM part_tags WHERE pk = OLD.pk;
                        INSERT INTO part_tags (tag, level) VALUES (NEW.tag,
 NEW.level);
                END IF;
                RETURN NULL;
        END;
$$
LANGUAGE 'plpgsql';
*************************

CREATE FUNCTION
********* QUERY **********
CREATE TRIGGER update_part_tags_trigger BEFORE UPDATE ON part_tags_level_0
FOR EACH ROW EXECUTE PROCEDURE update_part_tags();
*************************

CREATE TRIGGER
********* QUERY **********
CREATE TRIGGER update_part_tags_trigger BEFORE UPDATE ON part_tags_level_1
FOR EACH ROW EXECUTE PROCEDURE update_part_tags();
*************************

CREATE TRIGGER
********* QUERY **********
CREATE TRIGGER update_part_tags_trigger BEFORE UPDATE ON part_tags_level_2
FOR EACH ROW EXECUTE PROCEDURE update_part_tags();
*************************

CREATE TRIGGER
********* QUERY **********
CREATE TRIGGER update_part_tags_trigger BEFORE UPDATE ON part_tags_level_3
FOR EACH ROW EXECUTE PROCEDURE update_part_tags();
*************************

CREATE TRIGGER
********* QUERY **********
UPDATE part_tags
SET level = 1, tag = 'apple'
WHERE pk = 5;
*************************

UPDATE 0
********* QUERY **********
SELECT * FROM part_tags;
*************************

 pk |    tag     | level
```

```
----+------------+-------
  1 | vegetables |     0
  2 | fruits     |     0
  3 | apple      |     1
  6 | apple      |     1
(4 rows)

********* QUERY **********
SELECT * FROM part_tags_level_1;
**************************

 pk |  tag  | level
----+-------+-------
  3 | apple |     1
  6 | apple |     1
(2 rows)

********* QUERY **********
SELECT * FROM part_tags_level_2;
**************************

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

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

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags (
        pk serial,
        level integer NOT NULL DEFAULT 0,
        tag text NOT NULL,
        CONSTRAINT part_tags_pkey PRIMARY KEY (pk, level)
)
PARTITION BY LIST (LEVEL);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_level_0 PARTITION OF part_tags FO
R VALUES IN (0);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_level_1 PARTITION OF part_tags FO
R VALUES IN (1);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_level_2 PARTITION OF part_tags FO
R VALUES IN (2);
**************************

CREATE TABLE
********* QUERY **********
```

```
CREATE TEMP TABLE IF NOT EXISTS part_tags_level_3 PARTITION OF part_tags FO
R VALUES IN (3);
*************************

CREATE TABLE
********* QUERY **********
CREATE INDEX part_tags_tag ON part_tags USING GIN (tag gin_trgm_ops);
*************************

CREATE INDEX
                     Partitioned table "pg_temp_3.part_tags"
 Column |  Type   | Collation | Nullable |                Default

--------+---------+-----------+----------+----------------------------------
------
 pk     | integer |           | not null | nextval('part_tags_pk_seq'::regc
lass)
 level  | integer |           | not null | 0
 tag    | text    |           | not null |
Partition key: LIST (level)
Indexes:
    "part_tags_pkey" PRIMARY KEY, btree (pk, level)
    "part_tags_tag" gin (tag gin_trgm_ops)
Number of partitions: 4 (Use \d+ to list them.)

                       Table "pg_temp_3.part_tags_level_0"
 Column |  Type   | Collation | Nullable |                Default

--------+---------+-----------+----------+----------------------------------
------
 pk     | integer |           | not null | nextval('part_tags_pk_seq'::regc
lass)
 level  | integer |           | not null | 0
 tag    | text    |           | not null |
Partition of: part_tags FOR VALUES IN (0)
Indexes:
    "part_tags_level_0_pkey" PRIMARY KEY, btree (pk, level)
    "part_tags_level_0_tag_idx" gin (tag gin_trgm_ops)

********* QUERY **********
INSERT INTO part_tags (tag, level)
VALUES
        ('vegetables', 0),
        ('fruits', 0),
        ('orange', 1),
        ('apple', 1),
        ('red apple', 2);
*************************

INSERT 0 5
********* QUERY **********
SELECT * FROM part_tags;
*************************

 pk | level |    tag
----+-------+------------
  1 |     0 | vegetables
  2 |     0 | fruits
  3 |     1 | orange
  4 |     1 | apple
  5 |     2 | red apple
(5 rows)

********* QUERY **********
SELECT * FROM ONLY part_tags;
**************************

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

********* QUERY **********
SELECT * FROM part_tags_level_0;
**************************

 pk | level |    tag
----+-------+------------
  1 |     0 | vegetables
  2 |     0 | fruits
(2 rows)

********* QUERY **********
SELECT * FROM part_tags_level_1;
**************************

 pk | level |  tag
----+-------+--------
  3 |     1 | orange
  4 |     1 | apple
(2 rows)

********* QUERY **********
SELECT * FROM part_tags_level_2;
**************************

 pk | level |    tag
----+-------+-----------
  5 |     2 | red apple
(1 row)

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

DROP TABLE
********* QUERY **********
DROP TABLE IF EXISTS part_tags_date_05_2020;
**************************

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags (
        pk serial,
        insert_date date NOT NULL DEFAULT now()::date,
        tag text NOT NULL,
        level integer NOT NULL DEFAULT 0,
        CONSTRAINT part_tags_pkey PRIMARY KEY (pk, insert_date)
)
PARTITION BY RANGE (insert_date);
**************************
```

```
CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_date_01_2020 PARTITION OF part_ta
gs FOR VALUES FROM ('2020-01-01') TO ('2020-01-31');
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_date_02_2020 PARTITION OF part_ta
gs FOR VALUES FROM ('2020-02-01') TO ('2020-02-28');
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_date_03_2020 PARTITION OF part_ta
gs FOR VALUES FROM ('2020-03-01') TO ('2020-03-31');
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_date_04_2020 PARTITION OF part_ta
gs FOR VALUES FROM ('2020-04-01') TO ('2020-04-30');
**************************

CREATE TABLE
********* QUERY **********
CREATE INDEX part_tags_tag ON part_tags USING GIN (tag gin_trgm_ops);
**************************

CREATE INDEX
                       Partitioned table "pg_temp_3.part_tags"
   Column    |  Type   | Collation | Nullable |               Default

-------------+---------+-----------+----------+----------------------------
-----------
 pk          | integer |           | not null | nextval('part_tags_pk_seq':
:regclass)
 insert_date | date    |           | not null | now()::date
 tag         | text    |           | not null |
 level       | integer |           | not null | 0
Partition key: RANGE (insert_date)
Indexes:
    "part_tags_pkey" PRIMARY KEY, btree (pk, insert_date)
    "part_tags_tag" gin (tag gin_trgm_ops)
Number of partitions: 4 (Use \d+ to list them.)

                       Table "pg_temp_3.part_tags_date_01_2020"
   Column    |  Type   | Collation | Nullable |               Default

-------------+---------+-----------+----------+----------------------------
-----------
 pk          | integer |           | not null | nextval('part_tags_pk_seq':
:regclass)
 insert_date | date    |           | not null | now()::date
 tag         | text    |           | not null |
 level       | integer |           | not null | 0
Partition of: part_tags FOR VALUES FROM ('2020-01-01') TO ('2020-01-31')
Indexes:
    "part_tags_date_01_2020_pkey" PRIMARY KEY, btree (pk, insert_date)
    "part_tags_date_01_2020_tag_idx" gin (tag gin_trgm_ops)

********* QUERY **********
INSERT INTO part_tags (tag, insert_date, level)
VALUES
        ('vegetables', '2020-01-01', 0),
        ('fruits', '2020-01-01', 0),
        ('orange', '2020-02-01', 1),
        ('apple', '2020-03-010', 1),
        ('red apple', '2020-04-01', 2);
**************************

INSERT 0 5
********* QUERY **********
SELECT * FROM part_tags;
**************************

 pk | insert_date |    tag     | level
----+-------------+------------+-------
  1 | 2020-01-01  | vegetables |     0
  2 | 2020-01-01  | fruits     |     0
  3 | 2020-02-01  | orange     |     1
  4 | 2020-03-10  | apple      |     1
  5 | 2020-04-01  | red apple  |     2
(5 rows)

********* QUERY **********
SELECT * FROM ONLY part_tags;
**************************

 pk | insert_date | tag | level
----+-------------+-----+-------
(0 rows)

********* QUERY **********
SELECT * FROM part_tags_date_01_2020;
**************************

 pk | insert_date |    tag     | level
----+-------------+------------+-------
  1 | 2020-01-01  | vegetables |     0
  2 | 2020-01-01  | fruits     |     0
(2 rows)

********* QUERY **********
SELECT * FROM part_tags_date_02_2020;
**************************

 pk | insert_date |  tag   | level
----+-------------+--------+-------
  3 | 2020-02-01  | orange |     1
(1 row)

********* QUERY **********
SELECT * FROM part_tags_date_03_2020;
**************************

 pk | insert_date |  tag  | level
----+-------------+-------+-------
  4 | 2020-03-10  | apple |     1
```

```
(1 row)

********* QUERY **********
SELECT * FROM part_tags_date_04_2020;
**************************

 pk | insert_date |    tag    | level
----+-------------+-----------+-------
  5 | 2020-04-01  | red apple |     2
(1 row)

********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS part_tags_date_05_2020 PARTITION OF part_ta
gs FOR VALUES FROM ('2020-05-01') TO ('2020-05-30');
**************************

CREATE TABLE
                                           Partitioned table "pg_temp_3.par
t_tags"
   Column    |  Type   | Collation | Nullable |                Default
           | Storage  | Stats target | Description
-------------+---------+-----------+----------+----------------------------
-----------+----------+--------------+-------------
 pk          | integer |           | not null | nextval('part_tags_pk_seq':
:regclass) | plain    |              |
 insert_date | date    |           | not null | now()::date
           | plain    |              |
 tag         | text    |           | not null |
           | extended |              |
 level       | integer |           | not null | 0
           | plain    |              |
Partition key: RANGE (insert_date)
Indexes:
    "part_tags_pkey" PRIMARY KEY, btree (pk, insert_date)
    "part_tags_tag" gin (tag gin_trgm_ops)
Partitions: part_tags_date_01_2020 FOR VALUES FROM ('2020-01-01') TO ('2020
-01-31'),
            part_tags_date_02_2020 FOR VALUES FROM ('2020-02-01') TO ('2020
-02-28'),
            part_tags_date_03_2020 FOR VALUES FROM ('2020-03-01') TO ('2020
-03-31'),
            part_tags_date_04_2020 FOR VALUES FROM ('2020-04-01') TO ('2020
-04-30'),
            part_tags_date_05_2020 FOR VALUES FROM ('2020-05-01') TO ('2020
-05-30')

                       Table "pg_temp_3.part_tags_date_05_2020"
   Column    |  Type   | Collation | Nullable |                Default

-------------+---------+-----------+----------+----------------------------
-----------
 pk          | integer |           | not null | nextval('part_tags_pk_seq':
:regclass)
 insert_date | date    |           | not null | now()::date
 tag         | text    |           | not null |
 level       | integer |           | not null | 0
Partition of: part_tags FOR VALUES FROM ('2020-05-01') TO ('2020-05-30')
Indexes:
    "part_tags_date_05_2020_pkey" PRIMARY KEY, btree (pk, insert_date)
    "part_tags_date_05_2020_tag_idx" gin (tag gin_trgm_ops)
```

```
********* QUERY **********
ALTER TABLE part_tags
DETACH PARTITION part_tags_date_05_2020;
**************************

ALTER TABLE
                                           Partitioned table "pg_temp_3.par
t_tags"
   Column    |  Type   | Collation | Nullable |                Default
           | Storage  | Stats target | Description
-------------+---------+-----------+----------+-----------------------------
-----------+----------+--------------+-------------
 pk          | integer |           | not null | nextval('part_tags_pk_seq':
:regclass) | plain    |              |
 insert_date | date    |           | not null | now()::date
           | plain    |              |
 tag         | text    |           | not null |
           | extended |              |
 level       | integer |           | not null | 0
           | plain    |              |
Partition key: RANGE (insert_date)
Indexes:
    "part_tags_pkey" PRIMARY KEY, btree (pk, insert_date)
    "part_tags_tag" gin (tag gin_trgm_ops)
Partitions: part_tags_date_01_2020 FOR VALUES FROM ('2020-01-01') TO ('2020
-01-31'),
            part_tags_date_02_2020 FOR VALUES FROM ('2020-02-01') TO ('2020
-02-28'),
            part_tags_date_03_2020 FOR VALUES FROM ('2020-03-01') TO ('2020
-03-31'),
            part_tags_date_04_2020 FOR VALUES FROM ('2020-04-01') TO ('2020
-04-30')

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

DROP TABLE
********* QUERY **********
DROP TABLE IF EXISTS table_b;
**************************

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS table_a (
        pk integer NOT NULL PRIMARY KEY,
        tag text,
        parent integer);
**************************

CREATE TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS table_b () INHERITS (table_a);
**************************

CREATE TABLE
********* QUERY **********
ALTER TABLE table_b
ADD CONSTRAINT table_b_pk PRIMARY KEY (pk);
```

```
*************************

ALTER TABLE
********* QUERY **********
INSERT INTO table_a
VALUES (1, 'fruits', 0);
*************************

INSERT 0 1
********* QUERY **********
INSERT INTO table_b
VALUES (2, 'orange', 0);
*************************

INSERT 0 1
********* QUERY **********
SELECT * FROM table_a;
*************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |      0
  2 | orange |      0
(2 rows)

********* QUERY **********
SELECT * FROM table_b;
*************************

 pk |  tag   | parent
----+--------+--------
  2 | orange |      0
(1 row)

********* QUERY **********
SELECT * FROM ONLY table_a;
*************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |      0
(1 row)

********* QUERY **********
UPDATE table_a
SET tag = 'apple'
WHERE pk = 2;
*************************

UPDATE 1
********* QUERY **********
SELECT * FROM table_a;
*************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |      0
  2 | apple  |      0
(2 rows)
```

```
********* QUERY **********
SELECT * FROM table_b;
**************************

 pk |  tag  | parent
----+-------+--------
  2 | apple |      0
(1 row)

********* QUERY **********
DELETE FROM table_a WHERE pk = 2;
**************************

DELETE 1
********* QUERY **********
SELECT * FROM table_a;
**************************

 pk |  tag   | parent
----+--------+--------
  1 | fruits |      0
(1 row)

********* QUERY **********
SELECT * FROM table_b;
**************************

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