```
CREATE TABLE product (
  doc_id INT,
  details JSONB
);


INSERT INTO product VALUES (1, '{"CPU":4, "RAM":16, "HDD":2}');
INSERT INTO product VALUES (2, '{"CPU":3, "RAM":8, "HDD":1}');


doc_id |            details
----+---------------------------------
  1 | {"CPU": 4, "HDD": 2, "RAM": 16}
  2 | {"CPU": 3, "HDD": 1, "RAM": 8}




CREATE TABLE replace (
  id INT,
  doc_id INT,
  old_key text,
  new_key text
);
INSERT INTO replace VALUES (1, 1, 'CPU', 'RAM');
INSERT INTO replace VALUES (2, 1, 'HDD', 'SSD');
INSERT INTO replace VALUES (3, 2, 'HDD', 'SSD');

id | doc_id | old_key | new_key
----+--------+---------+---------
  1 |      1 | CPU     | RAM
  2 |      1 | HDD     | SSD
  3 |      2 | HDD     | SSD
```

```
id |            details
----+--------------------------------
  1 | {"SSD": 2, "RAM": 20}
  2 | {"CPU": 3, "SSD": 1, "RAM": 8}
```

Posted by BlackEric on Fri, 01 May 2026 18:28:23 GMT
View Forum Message <> Reply to Message

```
WITH upd AS (
  SELECT
    p.doc_id,
    jsonb_object_agg(


      CASE
        WHEN r.new_key IS NOT NULL AND p.details ? r.new_key
          THEN ( (p.details ->> key)::int + (p.details ->> r.new_key)::int )::text::jsonb
        ELSE value
      END
    ) AS new_details
  FROM product p
  LEFT JOIN replace r
         ON p.doc_id = r.doc_id AND (p.details ? r.old_key)
  CROSS JOIN LATERAL jsonb_each(p.details)
  WHERE r.old_key IS NULL OR key = r.old_key
  GROUP BY p.doc_id
)
UPDATE product AS p
SET details = u.new_details
FROM upd u
WHERE p.doc_id = u.doc_id;
```
