

graph_edb

```
1 DROP TABLE test.t_tmp_craw_edb ;
2 CREATE TABLE test.t_tmp_craw_edb(id varchar, vul_msg jsonb);
3
4
5 --vul_node
6 CREATE TABLE test.t_tmp_node_vul_edb AS
7 SELECT t.vul_msg ->> 'edb_id' AS id ,
8        t.vul_msg ->> 'cve_id' AS aliases,
9        jsonb_build_object('discovery', NULL, 'identifier', t.vul_msg ->> 'author') AS SOURCE,
10       '{}'::jsonb AS description,
11       '{}'::jsonb AS weaknesses,
12       '{}'::jsonb AS severity,
13       jsonb_build_object('published', t.vul_msg ->> 'date', 'lastModified', NULL, 'datePublic', null) AS time_
14       ('{"impact_info":{"impacts":null, "impactScore":null},
15        "solution_info":null,
16        "exploit_info":{"exploitable":null, "exploits":null, "exploit_url":"" || ('https://www.exploit-db.com
17        "PoC_info":{"PoC_available":null, "PoC":null, "PoC_url":null},
18        "patch_info":{"patch_available":null, "patch_url":null},
19        "report_status":"" || CASE lower(t.vul_msg ->> 'verified') WHEN 'true' THEN 'verified' ELSE 'not ve
20 FROM test.t_tmp_craw_edb t
21
22 DELETE FROM test.dws_graph_node_vul WHERE vul_source = 'EDB';
23 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
24 INSERT INTO test.dws_graph_node_vul
25 SELECT nextval('cve_graph_seq') AS seq, tmp.*
26 FROM
27 (
28 SELECT DISTINCT id , aliases , "source", description ,weaknesses, severity, time_info, vul_status, 'EDB' AS
29 )tmp ;
30
31
32
33 CVE-cve 2019-5786
34 CVE-cve 2020-6418
35 CVE-cve : 2023-30367
36
```