How to store the backup of any table row when ever anything change in a particular table row.

1. Create an archive table using following query.

CREATE TABLE IF NOT EXISTS archive.archives

(

    id bigint NOT NULL DEFAULT nextval('archive.archives\_id\_seq'::regclass),

    table\_name character varying COLLATE pg\_catalog."default",

    table\_id bigint,

    json\_data json,

    created\_by integer,

    created\_by\_unit integer,

    created\_at timestamp without time zone,

    ip character varying COLLATE pg\_catalog."default",

    action character varying COLLATE pg\_catalog."default",

    tg\_action character varying COLLATE pg\_catalog."default",

    CONSTRAINT archives\_pkey PRIMARY KEY (id)

)

1. Create a function to store data in archive table.

DECLARE

action1 TEXT;

begin

IF new.deleted\_at IS NOT NULL THEN

action1 :='delete';

ELSE

action1 :='update';

END IF;

INSERT INTO archive.archives(table\_name, table\_id, json\_data, created\_by, created\_by\_unit, created\_at, ip, action, tg\_action)

VALUES (TG\_TABLE\_SCHEMA || '.' || TG\_TABLE\_NAME, old.id, to\_json(old.\*), new.updated\_by, new.updated\_by\_unit, now(), new.updated\_ip, action1, TG\_OP);

RETURN NEW;

end

1. Add trigger into a table. Table in which you’re adding trigger that table must have columns – updated\_by(timestamp), updated\_by\_unit(bigint), updated\_ip(character varying). Else whenever you’ll add record or updated record error will be occur.

CREATE TRIGGER store\_data\_in\_archives

    BEFORE DELETE OR UPDATE

    ON master.branch

    FOR EACH ROW

    EXECUTE FUNCTION master.store\_data\_in\_archives();