

SQL



- יצירת יחסים והכנסת רשומות חדשות
- שליפת נתונים מתוך יחס אחד
- שליפת נתונים מתוך מספר יחסים
- פעולת הקבצה ופונקציות הקבצה
- פונקציות שימושיות
- ← טריגרים ופונקציות בפורמט PostgreSQL
- תרגול

טריגרים ופונקציות בפורמט PostgreSQL



כאשר משתמש מנסה לשנות נתונים בבסיס הנתונים שלנו נרצה לוודא שהפעולות שהוא עושה חוקיות.

בדיקות שכאלו יכולות להתבצע בצד של הלקוח, לדוגמה, נוכל להגדיר בממשק של המשתמש שאם הוא מזין כתובת דואר אלקטרונית ללא סימן הכרוכית (@) לא נקבל את הערך שהזין, או אם הוא מנסה להכניס מספר טלפון המכיל תווים השונים ממספרים, או מספר קצר או ארוך מידיי או מספר שלא מתחיל בתו '0' נדחה את הערך שניסה להזין. בדיקות מסוג זה לא קשורות לבסיס הנתונים ולא מתבצעות באמצעות SQL.

אך יש בדיקות שחייבות להתבצע בצד השרת. לדוגמה, אם משתמש מנסה ליצור רשומה חדשה בבסיס הנתונים שלנו (להירשם לאתר לדוגמה) עם כתובת מייל שכבר קיימת בבסיס הנתונים. או אם המשתמש מנסה לרשום תעודת זהות הקיימת ביחס. או בבסיס הנתונים של קופת החולים, אם משתמש מנסה להירשם לתור לבדיקה רפואית אצל רופא מסוים אך התור כבר תפוס.

לצורך כך נוכל ליצור טריגרים שיקפצו בכל פעם שיקרה משהו שנגדיר והם יזמנו פונקציה שתבדוק את התנאים שנגדיר ואם נגלה שהניסיון של המשתמש מפר את החוקים שקבענו נוכל לדחות אותו.

טריגר בפורמט PostgreSQL מורכב משני חלקים:

1. טריגר - כאן מוגדר מתי הטריגר קופץ ובאילו נסיבות הוא יקפיץ את הפונקציה,
2. פונקציה - כאן נבדקים כל התנאים שהגדרנו, אם הכל תקין השינוי בבסיס הנתונים מאושר, ואם לא, הוא נדחה.

תמיד רושמים את הפונקציה לפני הטריגר כדי שאם הטריגר ינסה לקרוא לפונקציה המהדר ידע שהיא קיימת.

טריגרים ופונקציות בפורמט PostgreSQL



טריגר ופונקציה בסיסית מורכבים מהחלקים שלהלן:

פסוק ליצירת הטריגר <code>trig_name</code>	<code>create or replace function trig_name()</code> returns trigger as\$\$
פתיחת אזור להכרזה על משתנים. אם אין שימוש במשתנים אין צורך ב- <code>declare</code>	<code>declare</code>
שמות המשתנים והסוג שלהם.	
<ul style="list-style-type: none">משתנה מסוג <code>numeric</code> משמש למספרים עשרוניים, יש להגדיר את כמות הספרות בסך הכל (בדוגמא-4) וכן את כמות הספרות אחרי הנקודה (בדוגמא-2), לדוגמא עבור המספר 123.58 יש להגדיר: <code>numeric(5,2)</code>.משתנה מסוג <code>varchar</code> משמש לקליטת מילים, יש להגדיר את אורך המילה המקסימלית האפשרית (בדוגמא-20).	<code>varName1 integer;</code> <code>varName2 boolean;</code> <code>varName3 numeric(4,2);</code> <code>varName4 varchar(20);</code>
התחלת הפונקציה עצמה. גוף הפונקציה יכול לכלול לולאות <code>for</code> , משפטי <code>if</code> , ופעולות נוספות כמו הכנסה של רשומה ליחס כלשהו: <code>insert into table_name values(A,B,C,D...)</code>	<code>begin</code>
התחלת משפט "אם". בתוך התנאי אפשר לבדוק האם מתקיים משהו בתוך יחס הקיים במערכת (באמצעות שאילתה רגילה), אפשר להשוות את אחד הערכים בשורה החדשה שמנסים להכניס לבסיס הנתונים, לדוגמא: <code>if(new.columnA in (select columnA from tale_name))</code>	<code>If(condition) then</code>
הקפצת הודעת שגיאה למשתמש	<code>raise notice 'error message for the user';</code>
החזרת ערך ריק למשתמש ומניעת הכנסת הרשומה החדשה לבסיס הנתונים	<code>return null;</code>
סיום משפט "אם"	<code>end if;</code>
הכנסת הרשומה החדשה לבסיס הנתונים	<code>return new;</code>
סיום הפונקציה	<code>end; \$\$ language plpgsql;</code>
יצירת הטריגר <code>T1</code> וקביעה מתי הוא יקרה (<code>before/after/instead of</code>) ועבור איזה סוגים של אירועים (הכנסה או עדכון של רשומה) ועל איזו טבלה הטריגר יופעל	<code>create or replace trigger T1 before insert or update on table_name</code>
האם לבצע את הטריגר לכל שורה בנפרד: <code>for each row</code> , או לכל ההוראה ביחד: <code>for each statement</code>	<code>for each row</code>
אם רוצים להפעיל את הטריגר רק במקרים מסוימים אפשר להשתמש ב- <code>when</code> (לא חובה).	<code>when (condition)</code>
זימון הפונקציה והפעלתה (אפשר לשלוח לפונקציה פרמטרים לשימושה בתוך הסוגריים)	<code>execute procedure trig_name();</code>

טריגרים ופונקציות בפורמט PostgreSQL



אם נניח שלכל רופא יש בכל יום רק תור אחד שבו הוא מקבל, ונרצה לוודא שאין שני קליינטים הנרשמים לאותו התאריך:

```
CREATE OR REPLACE FUNCTION trigf1() RETURNS TRIGGER AS $$
BEGIN
    IF (EXISTS (SELECT *
                FROM treatment
                WHERE new.did=did AND new.tdate=tdate))
    THEN
        RAISE NOTICE 'התור המבוקש תפוס, אנא נסה תאריך או רופא אחרים';
        RETURN NULL;
    END IF;
    RETURN NEW;
END; $$ LANGUAGE PLPGSQL;
```

```
CREATE OR REPLACE TRIGGER T1 BEFORE INSERT ON treatment
FOR EACH ROW
EXECUTE PROCEDURE trigf1());
```

```
doctor (did, dname, dfamily, dphone, dcity, dstreet, dhouse, dspeciality, ddatestart, dsalary)
patient (pid, pname, pfamily, pbirthdate)
clinic (cid, ccity, caddress, cname)
treatment (tdate, did, pid, cid)
```