

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

יצירת יחסים והכנסת רשומות חדשות



SQL מאפשרת ליצור טבלאות חדשות באמצעות פסוק המתחיל בפקודה **CREATE TABLE**. כאשר יוצרים טבלה חדשה מגדירים את העמודות שיהיו בטבלה. עבור כל עמודה מגדירים את סוג המידע שיהיה אפשר להכניס בתאים שבה. ב-SQL יש סוגים רבים של מידע, אך אנו בקורס נתמקד בנפוצים יותר לשימוש. סוגי המידע הינם מחרוזות, מספרים שלמים, מספרים עם שברים, ערכים בוליאניים (אמת או שקר), תאריך ועוד.

כאשר יוצרים טבלה חדשה יש להגדיר את המפתח של הטבלה. אם המפתח מורכב מתכונה אחת אפשר להגדיר אותו בצמוד להגדרה של התכונה (מיד אחרי סוג המידע), אך אם המפתח מורכב ממספר תכונות מגדירים אותו בסוף המשפט ליצירת הטבלה - **PRIMARY KEY**.

אפשר לדרוש אילוצים שונים על הערכים שבטבלה, לדוגמה:

אם הטבלה מכילה מפתח זר מטבלה אחרת יש להגדיר זאת בעת יצירת הטבלה - **FOREIGN KEY (column name) REFERENCES table name**.

אם יש תכונות מסוימות שאנחנו רוצים לחייב את המשתמש להזין בהן נתונים כלשהם כאשר הוא מכניס רשומה חדשה ליחס אפשר להגדיר את הדרישה הזו בעת יצירת הטבלה - **NOT NULL**.

אם יש תכונה מסוימת שאנו רוצים להגדיר בה ערך כלשהו בתור ברירת מחדל נשתמש באילוץ **DEFAULT** מיד אחרי סוג המידע של התכונה (לדוגמה: אפשר לכתוב **DEFAULT NULL**, ואז כאשר נכניס שורה חדשה לטבלה, הערך בעמודה זו יהיה ריק).

אם נרצה לוודא שערך של תכונה כלשהי עונה על תנאי מסוים, נוכל להשתמש באילוץ **CHECK** בסיום הגדרת התכונות.

(לדוגמה: אם נרצה שהתכונה גיל תהיה בין 18 ל-65, נוכל לכתוב אילוץ אחד: **CHECK (age >= 18 and age <= 65)**

או שני אילוצים: **CHECK (age <= 65)**, **CHECK (age >= 18)**).

אם נרצה לדרוש שערך של תכונה כלשהי יהיה ייחודי למרות שהתכונה הזו לא מפתח, נשתמש ב-**UNIQUE**.

הרצה של קוד הכולל משפט ליצירת טבלה גורמת למהדר לנסות ליצור את הטבלה הזו ועכשיו אתם מנסים ליצור אותה שוב (אולי בגלל שתיקנתם משהו בטבלה) המהדר יקפיץ לכם שגיאה כיוון שהטבלה כבר קיימת בבסיס הנתונים, ולכן חייבים למחוק את

הטבלה לפני שיוצרים אותה שוב באמצעות פסוק **DROP TABLE**

יצירת יחסים והכנסת רשומות חדשות



כפי שראינו, נהוג לקרוא לתכונות בשמות הכוללים את האות הראשונה של הישות ואת שם התכונה, לדוגמה: לתעודת הזהות של רופא נקרא did - ה-d הראשונה עבור doctor וה-id עבור תעודת הזהות, או לדוגמה לשם הפרטי של הרופא נקרא dname.

צרו את הטבלאות הבאות (מפתחות מסומנים בקו תחתון):

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)

ליצירת היחסים נשתמש בפקודה create table. נשים לב כי רוב התכונות ביחס "טיפול" הם מפתחות זרים.

ראשית נמחק את כל הטבלאות האלו אם הן קיימות בבסיס הנתונים (אם ננסה למחוק טבלה שלא קיימת נקבל שגיאה ולכן אנחנו מבקשים למחוק רק אם היא קיימת) בכדי למנוע מצב שבו אנחנו מנסים ליצור טבלה שכבר קיימת. כאשר מוחקים טבלאות שתלויות אחת בשנייה עלולה לקפוז שגיאה כיוון שאין אפשרות למחוק טבלה שיש בה ערכים שטבלה אחרת משתמשת בהם. בכדי למנוע את השגיאה יש לרשום CASCADE בסוף הפקודה שמשמעותו להשמיד את הטבלה ואת כל הרשומות שתלויות בה בטבלאות אחרות:

DROP TABLE IF EXISTS doctor CASCADE;

DROP TABLE IF EXISTS patient CASCADE;

DROP TABLE IF EXISTS clinic CASCADE;

DROP TABLE IF EXISTS treatment CASCADE;

יצירת יחסים והכנסת רשומות חדשות



כעת נרשום את הפקודות ליצירת הטבלאות:

```
CREATE TABLE doctor (did INT PRIMARY KEY, dname VARCHAR(30), dfamily VARCHAR(30), dphone INTEGER, dcity VARCHAR(30), dstreet VARCHAR(30), dhouse int, dspeciality VARCHAR(30), ddatestart DATE, dsalary NUMERIC(10,2));
```

```
CREATE TABLE patient (pid INT PRIMARY KEY, pname VARCHAR(30), pfamily VARCHAR(30), pbirthdate DATE);
```

```
CREATE TABLE clinic (cid INT, ccity VARCHAR(30), caddress VARCHAR(30), cname VARCHAR(30), PRIMARY KEY (cid));
```

```
CREATE TABLE treatment (tdate DATE, did INT, pid INT, cid INT, PRIMARY KEY (tdate, did, pid, cid), FOREIGN KEY (did) REFERENCES doctor, FOREIGN KEY (pid) REFERENCES patient, FOREIGN KEY (cid) REFERENCES clinic);
```

יצירת יחסים והכנסת רשומות חדשות



הכניסו את הנתונים הבאים לטבלאות שיצרתם:

doctor									
did	dname	dfamily	dphone	dcity	dstreet	dhouse	dspeciality	ddatestart	dsalary
111	yossi	levi	0255555	jerusalem	ha-agas	1	family	1/8/1948	15000
222	moshe	cohen	0344444	tel aviv	bugrashov	5	gynecologist	4/8/2011	11000
555	rivka	levi	0344885	tel aviv	rotshild	4	gynecologist	5/4/2005	10000
444	sara	rubin	0357589	jaffa	erlich	32	proctologist	8/7/1976	17000
777	gabi	nagel	0489757	haifa	hatichon	66	dermatologist	15/11/1987	16000
333	sara	shnaider	0397578	tel aviv	alenbi	3	family	17/1/2013	18000

patient			
pid	pname	pfamily	pbirthdate
123	ofer	ben yaakov	20/11/1988
124	reut	barkai	3/5/2015
125	revital	aochanoona	6/2/2001
126	yogev	rappel	8/3/1970
127	shaked	ben moshe	12/12/1988
128	roni	lavi	27/1/1993

יצירת יחסים והכנסת רשומות חדשות



clinic			
cid	ccity	caddress	cname
911	jerusalem	jaffa 15	refuaa mashlima jlm
912	tel aviv	dizingof 64	dizingof center
913	jaffa	shderot yerushalaim 1	smile jaffa
914	jaffa	kibuts galuyot 5	orthopedics jaffa
915	haifa	mivtsa yonatan 22	mental health
916	jerusalem	nakdimon 8	hamerkaz lakashis

treatment			
<u>tdate</u>	<u>did</u>	<u>pid</u>	<u>cid</u>
17/1/2022	111	123	911
17/1/2022	555	126	916
17/1/2022	111	128	913
18/1/2022	444	126	912
18/1/2022	444	127	913
19/1/2022	333	127	916
19/1/2022	111	124	914
19/1/2022	222	125	914
19/1/2022	777	123	912
20/1/2022	222	125	911
20/1/2022	555	128	915
21/1/2022	444	124	916

יצירת יחסים והכנסת רשומות חדשות

```
INSERT INTO doctor VALUES (111,'yossi','levi',0255555,'jerusalem','ha-agas',1,'falmily','1948-08-01',15000),
(222,'moshe','cohen',0344444,'tel aviv','bugrashov',5,'gynecologist','2011-08-04',11000),
(555,'rivka','levi',0344885,'tel aviv','rotshild',4,'gynecologist','2005-04-05',10000),
(444,'sara','rubin',0357589,'jerusalem','erlich',32,'proctologist','1976-07-08',17000),
(777,'gabi','nagel',0489757,'jerusalem','hatichon',66,'dermatologist','1987-11-15',16000),
(333,'sara','shnaider',0397578,'tel aviv','alenbi',3,'falmily','2013-1-17',18000);
```

[illegible]

יצירת יחסים והכנסת רשומות חדשות



```
INSERT INTO clinic VALUES (911,'jerusalem','jaffa 15','refuaa mashlima jlm'),  
                             (912,'tel aviv','dizingof 64','dizingof center'),  
                             (913,'jaffa','shderot yerushalaim 1','smile jaffa'),  
                             (914,'jaffa','kibuts galuyot 5','orthopedics jaffa'),  
                             (915,'haifa','mivtsa yonatan 22','mental health'),  
                             (916,'jerusalem','nakdimon 8','hamerkaz lakashis');
```


יצירת יחסים והכנסת רשומות חדשות



אם נרצה לעדכן רשומות שכבר קיימות בבסיס הנתונים נוכל להשתמש בפקודת update. אם נשתמש בפקודה סתם כך ללא תנאי where הפקודה תעדכן את כל השורות הקיימות ביחס, אם נרצה לעדכן רק שורות ספציפיות נוכל להגדיר אותן באמצעות where.

לדוגמה, הרופא בעל תעודת הזהות 111 שינה את שמו לרמי לוי מתוך תקווה שהשינוי יגדיל את משכורתו. עדכנו את פרטיו:

```
UPDATE doctor SET dname='rami', dfamily='levi' WHERE did=111;
```

אם נרצה למחוק רשומה כלשהי מהיחס, נשתמש בפקודה delete. גם כאן, אם לא נשתמש בתנאי where כל השורות ביחס יימחקו(!) וגם כאן, אם התנאי where כולל בתוכו כמה רשומות, כל הרשומות שעונות לתנאי יימחקו. לדוגמה, אם נמחק את כל הרשומות בהן שם המשפחה שווה 'לוי', הרי שיימחקו מספר שורות מהיחס, כיוון שקיימות מספר שורות העונות על התנאי.

לדוגמה, הרופאים שהתחילו לעבוד לפי 1950 יצאו לפנסיה ועזבו את המרפאה. מחקו את הרשומות של רופאים אלו:

```
DELETE FROM doctor WHERE ddatestart < '1950-01-01';
```

אם נריץ את הפקודה לעיל נקבל שגיאה כיוון שהמפתח של "רופא" נמצא כמפתח זר ביחס "טיפול" ולכן נצטרך להגדיר ביחס "טיפול" שכאשר מוחקים רשומה כלשהי מהיחס "רופא", כל השורות ביחס "טיפול" שקשורות לאותו רופא יימחקו:

```
CREATE TABLE treatment (tdate DATE, did INT, pid INT, cid INT, PRIMARY KEY (tdate, did, pid, cid), FOREIGN KEY (did) REFERENCES doctor ON DELETE CASCADE, FOREIGN KEY (pid) REFERENCES patient, FOREIGN KEY (cid) REFERENCES clinic);
```

יצירת יחסים והכנסת רשומות חדשות



אם נרצה לשנות את הטבלה עצמה, לדוגמה אם נרצה להוסיף עמודה לטבלה או למחוק עמודה או לשנות את השם של עמודה כלשהי, או להוסיף אילוף על עמודה נעשה זאת באמצעות הפקודה ALTER:

ALTER TABLE table_name ADD COLUMN column_name datatype;	הוספת עמודה	שינוי עמודות
ALTER TABLE table_name DROP COLUMN column_name;	מחיקת עמודה	
ALTER TABLE table_name RENAME COLUMN old_name TO new_name;	שינוי שם עמודה	
ALTER TABLE table_name ALTER COLUMN column_name SET NOT NULL;	הוספת אילוף שעמודה לא תהיה ריקה	
ALTER TABLE table_name ALTER COLUMN column_name DROP NOT NULL;	מחיקת אילוף שעמודה לא תהיה ריקה	
ALTER TABLE table_name ALTER COLUMN column_name SET DEFAULT 'value';	הגדרת ערך ברירת מחדל לעמודה	
ALTER TABLE table_name ALTER COLUMN column_name DROP DEFAULT;	מחיקת ערך ברירת מחדל לעמודה	
ALTER TABLE table_name ALTER COLUMN column_name TYPE new_datatype;	שינוי סוג מידע של עמודה	
ALTER TABLE table_name ADD CONSTRAINT constraint_name PRIMARY KEY (column);	הוספת אילוף מפתח ראשי	הוספת והסרת אילוצים
ALTER TABLE table_name ADD CONSTRAINT constraint_name UNIQUE (column);	הוספת אילוף ערך ייחודי	
ALTER TABLE table_name ADD CONSTRAINT constraint_name FOREIGN KEY (column) REFERENCES other_table (other_column);	הוספת אילוף מפתח זר	
ALTER TABLE table_name ADD CONSTRAINT constraint_name CHECK (expression);	הוספת אילוף בדיקת ערך	
ALTER TABLE table_name DROP CONSTRAINT constraint_name;	מחיקת אילוף	
ALTER TABLE old_name RENAME TO new_name;	שינוי שם טבלה	

יצירת יחסים והכנסת רשומות חדשות

