ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



ΣΚΑΡΟΣ ΓΕΩΡΓΙΟΣ – Π15128 ΚΑΤΩΠΟΔΗΣ ΑΝΑΣΤΑΣΙΟΣ ΑΛΕΞΙΟΣ – Π15189 ΣΑΓΙΕΝΤ ΙΩΣΗΦ - Π15123

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ: 9/6/2017

- (a) Με βάση τα παραπάνω δεδομένα, σχεδιάστε το σχεσιακό σχήμα της ΒΔ και υλοποιήστε το (CREATE TABLE) στο ΣΔΒΔ PostgreSQL. Το παραδοτέο του υποερωτήματος είναι το σχεσιακό σχήμα της ΒΔ και οι εντολές CREATE TABLE.
- (b) Εφαρμόστε τη θεωρία της κανονικοποίησης πάνω στο σχεσιακό σχήμα της BΔ που σχεδιάσατε και ελέγξτε εάν ακολουθεί την 3NF ή BCNF.
- (c) Εισάγετε τα παραπάνω δεδομένα στη BΔ (COPY) που δημιουργήσατε. Το παραδοτέο του υποερωτήματος είναι οι εντολές που χρησιμοποιήθηκαν για την εισαγωγή των δεδομένων καθώς και screenshots που θα δείχνουν τους πίνακες φορτωμένους με τα δεδομένα.

```
a)
CREATE TABLE DEPARTMENTS (
  id serial NOT NULL,
  name varchar(35),
  PRIMARY KEY (id)
);
CREATE TABLE DRUGS (
  id serial NOT NULL,
  name text,
  description text,
  PRIMARY KEY (id)
);
CREATE TYPE genderType AS ENUM ('Male', 'Female');
CREATE TABLE PATIENT (
  patientAMKA varchar(40) NOT NULL,
  userid varchar(45),
  password varchar(20),
  name varchar(20),
  surname varchar(20),
  gender genderType,
  PRIMARY KEY (patientAMKA)
CREATE TABLE DOCTOR (
  doctorAMKA varchar(40) NOT NULL,
  username varchar(45),
  password varchar(20),
  name varchar(20),
  surname varchar(20),
  speciality smallint,
  PRIMARY KEY (doctorAMKA),
  FOREIGN KEY (speciality) REFERENCES DEPARTMENTS(id)
CREATE TABLE MEDIACL_FOLDER (
  id integer NOT NULL,
  patient varchar(40),
```

```
cure text,
  drug id integer,
  PRIMARY KEY (id),
  FOREIGN KEY (patient) REFERENCES PATIENT(patientAMKA),
  FOREIGN KEY (drug_id) REFERENCES DRUGS(id)
);
CREATE TABLE APPOINTMENTS (
  id integer NOT NULL,
  t timestamp,
  patientAMKA varchar(40),
  doctorAMKA varchar(40),
  diagnosis varchar(20),
  PRIMARY KEY (id),
  FOREIGN KEY (patientAMKA) REFERENCES PATIENT(patientAMKA),
  FOREIGN KEY (doctorAMKA) REFERENCES DOCTOR(doctorAMKA)
);
```

b)DOCTOR

FD1: doctorAMKA ->username, password, name, surname, speciality CK : {doctorAMKA}

- Ο πίνακας είναι 1NF αφού όλα τα πεδία τιμών των χαρακτηριστικών είναι ατομικά.
- Ο πίνακας είναι **2NF** γιατί κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται από τμήμα του κλειδιού.
- Ο πίνακας είναι **3NF** αφού κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται μεταβατικά από ένα υποψήφιο κλειδί (δηλαδή μέσω άλλου χαρακτηριστικού το οποίο δεν συμμετέχει στο υποψήφιο κλειδί).
- Ο πίνακας είναι **BCNF** αφού για κάθε συναρτησιακή εξάρτηση α->β το α είναι κλειδί.

PATIENT

FD1 : patientAMKA -> userid, password, name, surname, gender <u>CK</u> : {patientAMKA}

- Ο πίνακας είναι 1NF αφού όλα τα πεδία τιμών των χαρακτηριστικών είναι ατομικά.
- Ο πίνακας είναι **2NF** γιατί κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται από τμήμα του κλειδιού.
- Ο πίνακας είναι **3NF** αφού κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται μεταβατικά από ένα υποψήφιο κλειδί (δηλαδή μέσω άλλου χαρακτηριστικού το οποίο δεν συμμετέχει στο υποψήφιο κλειδί).
- Ο πίνακας είναι **BCNF** αφού για κάθε συναρτησιακή εξάρτηση α->β το α είναι κλειδί.

APPOINTMENTS

FD1 : id -> t, patientAMKA, doctorAMKA, diagnosis CK : {id}

- Ο πίνακας είναι 1NF αφού όλα τα πεδία τιμών των χαρακτηριστικών είναι ατομικά.
- Ο πίνακας είναι **2NF** γιατί κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται από τμήμα του κλειδιού.
- Ο πίνακας είναι **3NF** αφού κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται μεταβατικά από ένα υποψήφιο κλειδί (δηλαδή μέσω άλλου χαρακτηριστικού το οποίο δεν συμμετέχει στο υποψήφιο κλειδί).
- Ο πίνακας είναι **BCNF** αφού για κάθε συναρτησιακή εξάρτηση α->β το α είναι κλειδί.

MEDICAL FOLDER

FD1 : id-> patientAMKA, cure, drug_id

<u>CK</u>: {id}

- Ο πίνακας είναι 1NF αφού όλα τα πεδία τιμών των χαρακτηριστικών είναι ατομικά.
- Ο πίνακας είναι **2NF** γιατί κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται από τμήμα του κλειδιού.
- Ο πίνακας είναι **3NF** αφού κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται μεταβατικά από ένα υποψήφιο κλειδί (δηλαδή μέσω άλλου χαρακτηριστικού το οποίο δεν συμμετέχει στο υποψήφιο κλειδί).
- Ο πίνακας είναι **BCNF** αφού για κάθε συναρτησιακή εξάρτηση α->β το α είναι κλειδί.

DRUGS

FD1: id -> name, description

<u>CK</u> : {id}

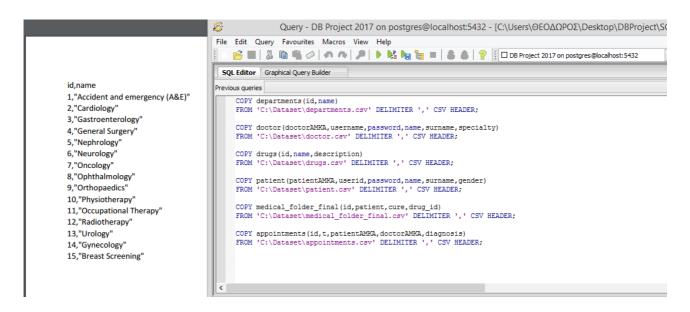
- Ο πίνακας είναι **1NF** αφού όλα τα πεδία τιμών των χαρακτηριστικών είναι ατομικά.
- Ο πίνακας είναι **2NF** γιατί κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται από τμήμα του κλειδιού.
- Ο πίνακας είναι **3NF** αφού κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται μεταβατικά από ένα υποψήφιο κλειδί (δηλαδή μέσω άλλου χαρακτηριστικού το οποίο δεν συμμετέχει στο υποψήφιο κλειδί).
- Ο πίνακας είναι **BCNF** αφού για κάθε συναρτησιακή εξάρτηση α->β το α είναι κλειδί.

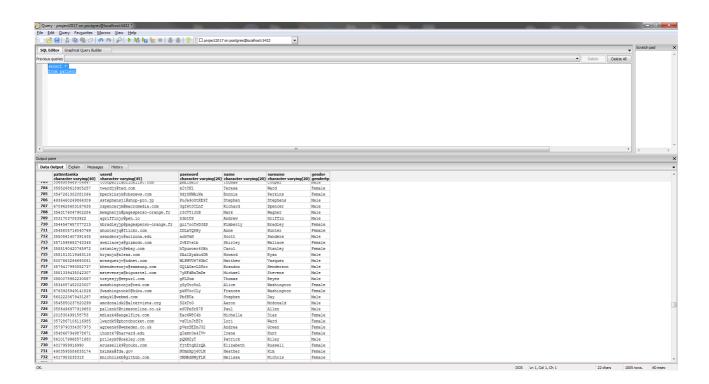
DEPARTMENTS

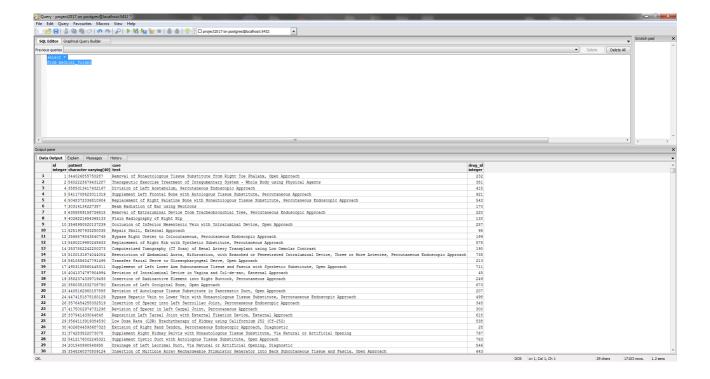
FD1: id -> name

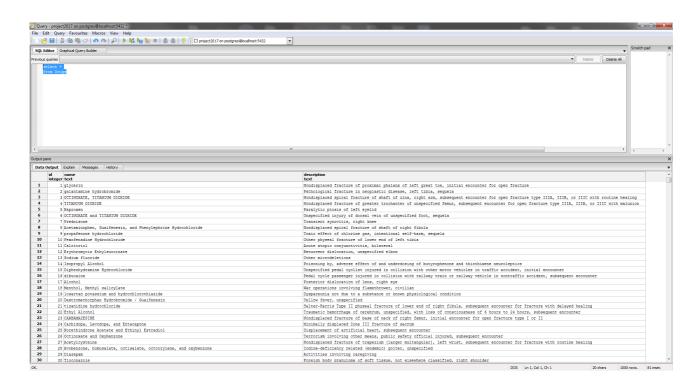
<u>CK</u>: {id}

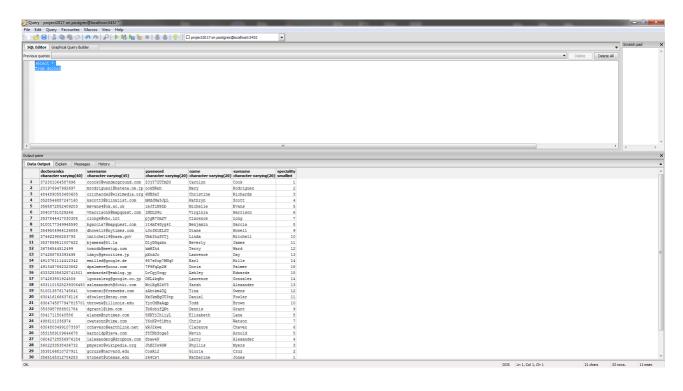
- Ο πίνακας είναι 1NF αφού όλα τα πεδία τιμών των χαρακτηριστικών είναι ατομικά.
- Ο πίνακας είναι **2NF** γιατί κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται από τμήμα του κλειδιού.
- Ο πίνακας είναι **3NF** αφού κανένα χαρακτηριστικό από αυτά που δεν συμμετέχουν στο κλειδί δεν προσδιορίζεται μεταβατικά από ένα υποψήφιο κλειδί (δηλαδή μέσω άλλου χαρακτηριστικού το οποίο δεν συμμετέχει στο υποψήφιο κλειδί).
- Ο πίνακας είναι **BCNF** αφού για κάθε συναρτησιακή εξάρτηση α->β το α είναι κλειδί.

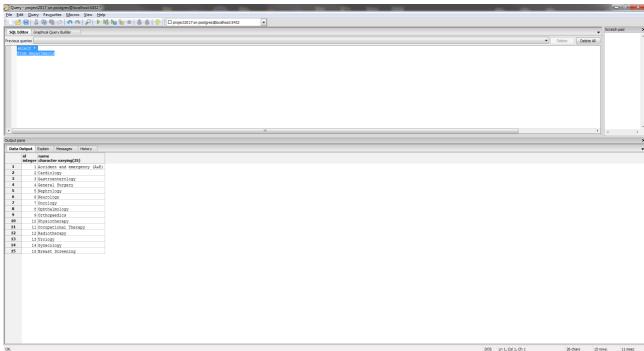


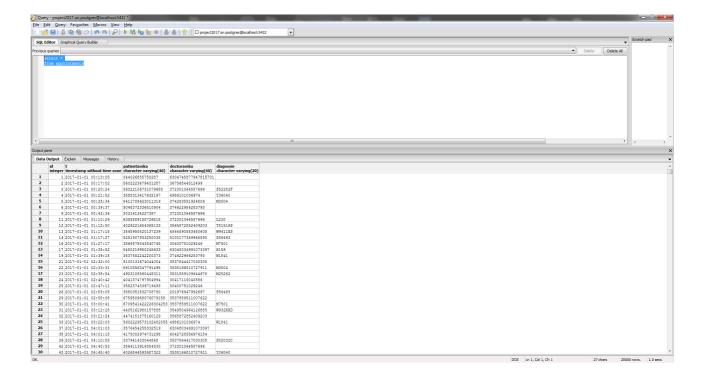












- (a) Ποια είναι τα φάρμακα (μέγιστο και ελάχιστο πλήθος) που συνταγογραφούνται. Στο αποτέλεσμα πρέπει να εμφανίζονται ποσότητα και όνομα φαρμάκου.
- (b) Ποιες είναι οι διαγνώσεις της τελευταίας εβδομάδας.
- (c) Ποια είναι τα ραντεβού του γιατρού 'Χ' τον τελευταία μήνα.
- (d) Ποιοι είναι οι ασθενείς που τους έχουν δει παραπάνω από ένας γιατροί.
- (e) Από ποιο τμήμα έχουν εξεταστεί οι περισσότεροι ασθενείς.
- (f) Ποιος είναι ο μέσος αριθμός των ασθενών που έχουν εξεταστεί (δηλαδή, υπάρχει διάγνωση) ανά τμήμα .

ΛΥΣΗ

a)

b)

SELECT diagnosis

```
FROM appointments
             WHERE t >
             (SELECT max(t)
             FROM appointments)-interval'7 days'
             AND diagnosis is not null
             ORDER BY 1
c)
      SELECT doctoramka, patientamka, t
      FROM Appointments
      WHERE t >
             (select max(t)
             from appointments)-interval'1 month'
      And doctoramka='30400751029246';
d)
      SELECT patientamka, count(doctoramka)
      FROM (SELECT DISTINCT patientamka, doctoramka
             FROM appointments
             WHERE diagnosis is not null
             order by 1)ASA
      GROUP BY patientamka
      HAVING count(doctoramka) > 1
      ORDER BY COUNT(doctoramka);
e)
SELECT id , name, count(patientamka) AS unique_patients
FROM(
             SELECT departments.id ,departments.name ,appointments.patientamka ,count(*)
      FROM appointments
      INNER JOIN doctor ON appointments.doctoramka=doctor.doctoramka
      INNER JOIN departments ON doctor.speciality=departments.id
      GROUP BY departments.id ,departments.name ,appointments.patientamka
      ORDER BY 1) as b
GROUP BY id, name
ORDER BY 3 DESC
LIMIT 1
```

```
select id , name , nonull::float/ withnull as avr_per_dep
from
      (SELECT id, name, count(patientamka) AS nonull
                   SELECT departments.id ,departments.name ,appointments.patientamka
      FROM(
,count(*)
             FROM appointments
             INNER JOIN doctor ON appointments.doctoramka=doctor.doctoramka
             INNER JOIN departments ON doctor.speciality=departments.id
             WHERE appointments.diagnosis is not NULL
             GROUP BY departments.id ,departments.name ,appointments.patientamka
             ORDER BY 1) AS A
      GROUP BY id, name
      ORDER BY 1 )AS A
INNER JOIN
      (SELECT id, name, count(patientamka) AS withnull
      FROM(
                   SELECT departments.id ,departments.name ,appointments.patientamka
,count(*)
             FROM appointments
             INNER JOIN doctor ON appointments.doctoramka=doctor.doctoramka
             INNER JOIN departments ON doctor.speciality=departments.id
             GROUP BY departments.id ,departments.name ,appointments.patientamka
             ORDER BY 1) AS B
      GROUP BY id, name
      ORDER BY 1 ) AS B USING (id,name);
```

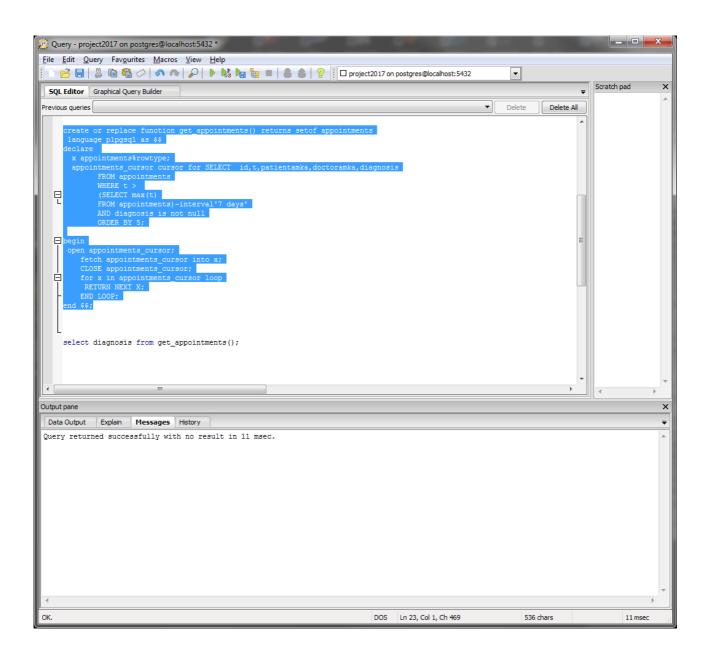
Αφού ολοκληρώσετε το Ερώτημα 1, απαντήστε στα παρακάτω:

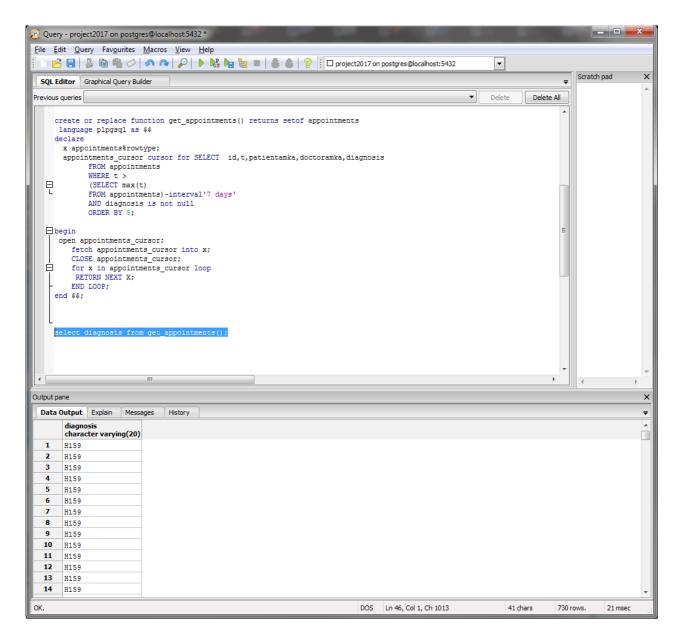
- (a) Σε αυτό το ερώτημα θα δημιουργήστε ένα trigger καθώς και τη συνάρτηση που θα εκτελείται όταν ενεργοποιείται το trigger. Κάθε φορά που θα πραγματοποιείται η εξέταση ενός ασθενούς και υπάρχει διάγνωση, θα ενημερώνεται αυτόματα και ο ιατρικός του φάκελος.
- (b) Απαντήστε σε ένα από τα υποερωτήματα από το ερώτημα 2 (όποιο επιθυμείτε) χρησιμοποιώντας cursors. Συγκεκριμένα, θα φτιάξετε μια συνάρτηση, η οποία θα διαβάζει το αποτέλεσμα από το query με έναν cursor και θα επιστρέφει το αποτέλεσμα σαν πίνακα. Καλέστε τη συνάρτηση μία φορά και δείξτε το αποτέλεσμα με screenshot.

ΛΥΣΗ

a)

```
CREATE OR REPIACE FUNCTION medical_folder_update() RETURNS trigger AS
DECLARE oldId integer;
BEGIN
      IF (new.diagnosis)is not null then
             SELECT id INTO oldId FROM medical folder ORDER BY 1 DESC LIMIT 1;
             INSERT INTO medical_folder(id,patient,cure,drug_id)
             VALUES(oldId+1,new.patientamka,new.diagnosis,null);
             RETURN new:
      END IF;
      END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER mdupdate AFTER INSERT ON appointments
FOR EACH ROW EXECUTE PROCEDURE medical folder update();
INSERT INTO appointments (id,t,patientamka,doctoramka,diagnosis) VALUES
  ('28013', null, '3568345415910816', '201976947992697', 'lick');
b)
create or replace function get appointments() returns setof appointments
language plpgsql as $$
declare
 x appointments%rowtype;
 appointments_cursor cursor for SELECT_id,t,patientamka,doctoramka,diagnosis
      FROM appointments
      WHERE t >
      (SELECT max(t)
      FROM appointments)-interval'7 days'
      AND diagnosis is not null
      ORDER BY 5;
begin
open appointments_cursor;
  fetch appointments_cursor into x;
  CLOSE appointments cursor;
  for x in appointments_cursor loop
  RETURN NEXT X;
  END LOOP;
end $$:
select diagnosis from get_appointments();
```





(a) Υλοποιήστε προγραμματιστικά έναν client σε οποιαδήποτε γλώσσα προγραμματισμού γνωρίζετε (π.χ. Java) χρησιμοποιώντας την κατάλληλη βιβλιοθήκη σύνδεσης με την PostgreSQL (π.χ. JDBC). Ο client θα συνδέεται στο ΣΔΒΔ της PostgreSQL, θα εκτελεί τα queries του Ερωτήματος 2, και θα εμφανίζει τα αποτελέσματα στον χρήστη (είτε σε terminal είτε με γραφικά)

ΛΥΣΗ

a)

```
import psycopg2
con=psycopg2.connect("dbname='project2017' user='postgres' host='localhost' password='admin'")
cur = con.cursor()
cur.execute("""select id , name , nonull::float/ withnull as avr_per_dep
from
      (SELECT id, name, count(patientamka) AS nonull
                    SELECT departments.id ,departments.name ,appointments.patientamka
      FROM(
,count(*)
             FROM appointments
             INNER JOIN doctor ON appointments.doctoramka=doctor.doctoramka
             INNER JOIN departments ON doctor.speciality=departments.id
             WHERE appointments.diagnosis is not NULL
             GROUP BY departments.id ,departments.name ,appointments.patientamka
             ORDER BY 1) AS A
      GROUP BY id, name
      ORDER BY 1 )AS A
INNER JOIN
      (SELECT id, name, count(patientamka) AS withnull
      FROM(
                    SELECT departments.id ,departments.name ,appointments.patientamka
,count(*)
             FROM appointments
             INNER JOIN doctor ON appointments.doctoramka=doctor.doctoramka
             INNER JOIN departments ON doctor.speciality=departments.id
             GROUP BY departments.id ,departments.name ,appointments.patientamka
             ORDER BY 1) AS B
      GROUP BY id, name
      ORDER BY 1 ) AS B USING (id,name);
""")
rows = cur.fetchall()
for r in rows:
  print r
```