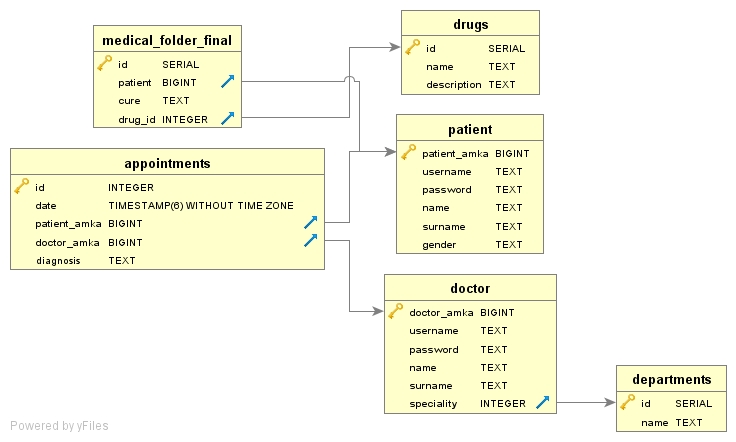
Μάθημα: «Βάσεις Δεδομένων (4ο Εξάμηνο)»

Π14053 | Παναγιώτα θωμοπουλου

# Ερώτημα 1

1. Σχεσιακό Σχήμα 

Εντολές Create:

CREATE TABLE public.appointments

(

id integer NOT NULL,

date timestamp without time zone NOT NULL,

patient\_amka bigint NOT NULL,

doctor\_amka bigint NOT NULL,

diagnosis text COLLATE pg\_catalog."default",

CONSTRAINT appointments\_pkey PRIMARY KEY (id),

CONSTRAINT doctor\_amka\_fkey FOREIGN KEY (doctor\_amka)

REFERENCES public.doctor (doctor\_amka) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT patient\_amka\_fkey FOREIGN KEY (patient\_amka)

REFERENCES public.patient (patient\_amka) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.appointments

OWNER to postgres;

=======================================================

CREATE TABLE public.departments

(

id integer NOT NULL DEFAULT nextval('departments\_id\_seq'::regclass),

name text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT departments\_pkey PRIMARY KEY (id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.departments

OWNER to postgres;

=======================================================

CREATE TABLE public.doctor

(

doctor\_amka bigint NOT NULL,

username text COLLATE pg\_catalog."default" NOT NULL,

password text COLLATE pg\_catalog."default" NOT NULL,

name text COLLATE pg\_catalog."default" NOT NULL,

surname text COLLATE pg\_catalog."default" NOT NULL,

speciality integer NOT NULL,

CONSTRAINT doctor\_pkey PRIMARY KEY (doctor\_amka),

CONSTRAINT speciality\_fkey FOREIGN KEY (speciality)

REFERENCES public.departments (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.doctor

OWNER to postgres;

=======================================================

CREATE TABLE public.drugs

(

id integer NOT NULL DEFAULT nextval('drugs\_id\_seq'::regclass),

name text COLLATE pg\_catalog."default" NOT NULL,

description text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT drugs\_pkey PRIMARY KEY (id)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.drugs

OWNER to postgres;

=======================================================

CREATE TABLE public.medical\_folder\_final

(

id integer NOT NULL DEFAULT nextval('medical\_folder\_final\_id\_seq'::regclass),

patient bigint NOT NULL,

cure text COLLATE pg\_catalog."default" NOT NULL,

drug\_id integer NOT NULL,

CONSTRAINT medical\_folder\_final\_pkey PRIMARY KEY (id),

CONSTRAINT drug\_id\_fkey FOREIGN KEY (drug\_id)

REFERENCES public.drugs (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION,

CONSTRAINT patient\_fkey FOREIGN KEY (patient)

REFERENCES public.patient (patient\_amka) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.medical\_folder\_final

OWNER to postgres;

=======================================================

CREATE TABLE public.patient

(

patient\_amka bigint NOT NULL,

username text COLLATE pg\_catalog."default" NOT NULL,

password text COLLATE pg\_catalog."default" NOT NULL,

name text COLLATE pg\_catalog."default" NOT NULL,

surname text COLLATE pg\_catalog."default" NOT NULL,

gender text COLLATE pg\_catalog."default" NOT NULL,

CONSTRAINT patient\_pkey PRIMARY KEY (patient\_amka)

)

WITH (

OIDS = FALSE

)

TABLESPACE pg\_default;

ALTER TABLE public.patient

OWNER to postgres;

b.

Η ΒΔ ακολουθεί την 3NF

c.

COPY public.departments(id,name)

FROM 'D:\Users\Penny\Desktop\Dataset\departments.csv' DELIMITER ',' CSV HEADER;

COPY public.doctor(doctor\_amka,username,password,name,surname,speciality)

FROM 'D:\Users\Penny\Desktop\Dataset\doctor.csv' DELIMITER ',' CSV HEADER;

COPY public.patient(patient\_amka,username,password,name,surname,gender)

FROM 'D:\Users\Penny\Desktop\Dataset\patient.csv' DELIMITER ',' CSV HEADER;

COPY public.drugs(id,name,description)

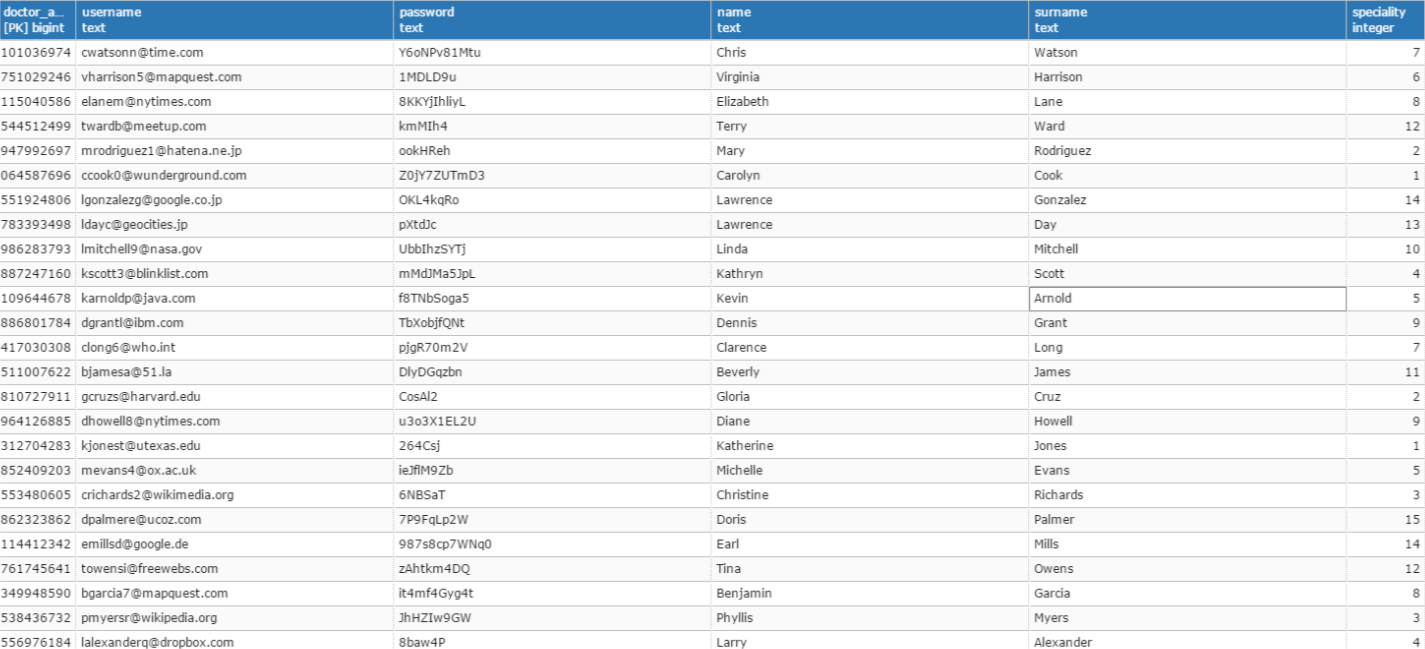
FROM 'D:\Users\Penny\Desktop\Dataset\drugs.csv' DELIMITER ',' CSV HEADER;

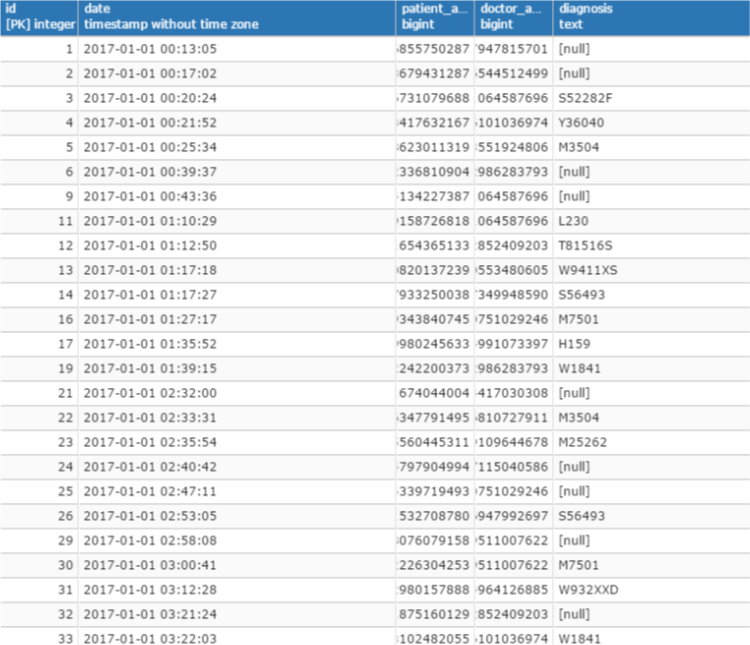
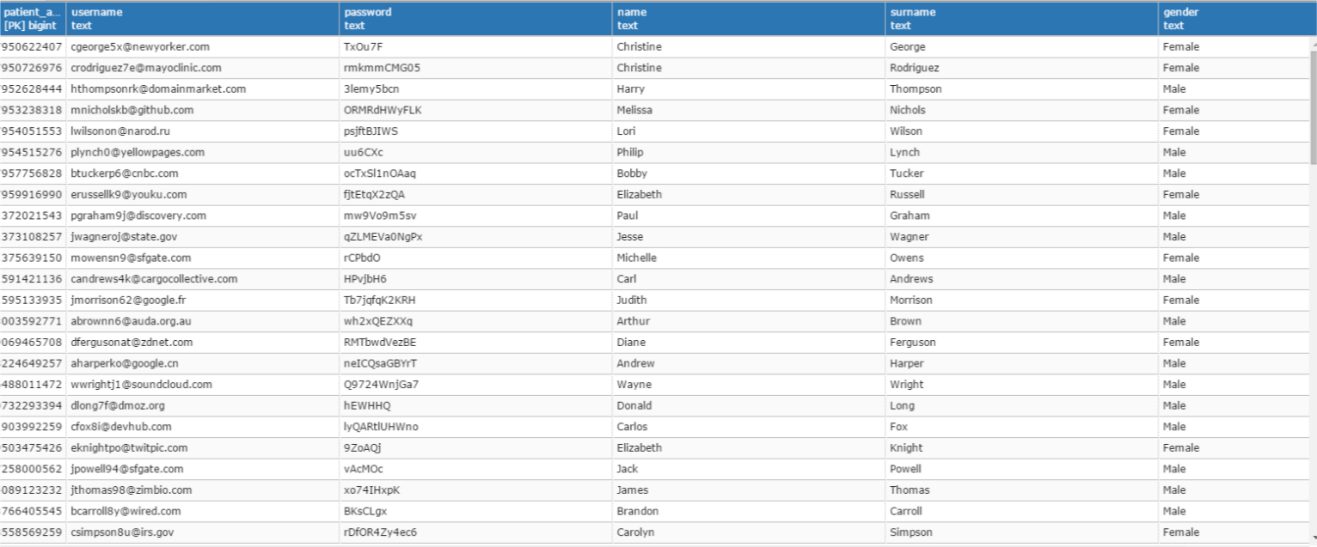
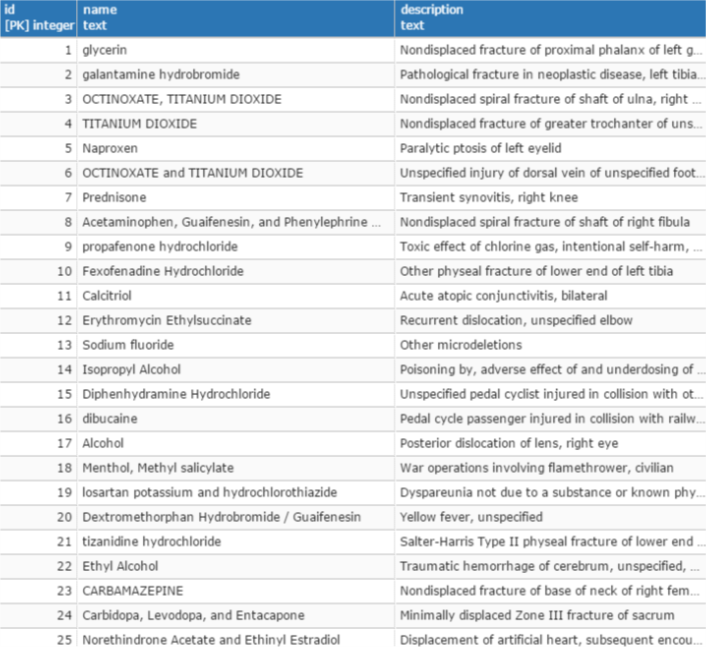
COPY public.appointments(id,date,patient\_amka,doctor\_amka,diagnosis)

FROM 'D:\Users\Penny\Desktop\Dataset\appointments.csv' DELIMITER ',' CSV HEADER;

COPY public.medical\_folder\_final(id,patient,cure,drug\_id)

FROM 'D:\Users\Penny\Desktop\Dataset\medical\_folder\_final.csv' DELIMITER ',' CSV HEADER;





# Ερώτημα 2

Α.

SELECT res.count, drugs.name

FROM

((SELECT drug\_id, COUNT(\*)

FROM medical\_folder\_final

GROUP BY drug\_id

ORDER BY COUNT DESC

LIMIT 1)

UNION ALL

(SELECT drug\_id, COUNT(\*)

FROM medical\_folder\_final

GROUP BY drug\_id

ORDER BY COUNT ASC

LIMIT 1)) as res

INNER JOIN drugs

ON res.drug\_id = drugs.id;

Β.

SELECT diagnosis

FROM appointments

WHERE date > CURRENT\_DATE - interval '7' day;

C.

SELECT res.\*

FROM (SELECT \*

FROM appointments

WHERE EXTRACT(month FROM date) = EXTRACT(month FROM CURRENT\_DATE)) as res

INNER JOIN doctor

ON res.doctor\_amka = doctor.doctor\_amka

WHERE doctor.name = 'X name' AND doctor.surname = 'X surname'

D.

SELECT pat3.\*

FROM

((SELECT patient\_amka, COUNT(\*)

FROM

(SELECT DISTINCT patient\_amka, doctor\_amka

FROM appointments) as pat

GROUP BY patient\_amka) as pat2

INNER JOIN patient

ON pat2.patient\_amka = patient.patient\_amka) as pat3

WHERE count > 1

E.

SELECT departments.name

FROM departments

INNER JOIN

(SELECT dep.id, COUNT(\*)

FROM appointments

INNER JOIN doctor as doc

ON appointments.doctor\_amka = doc.doctor\_amka

INNER JOIN departments as dep

ON doc.speciality = dep.id

GROUP BY dep.id

ORDER BY COUNT DESC

LIMIT 1) as res

ON res.id = departments.id

F.

SELECT COUNT(res.patient\_amka)/15

FROM (SELECT DISTINCT patient\_amka, doctor.speciality

FROM appointments

INNER JOIN doctor

ON appointments.doctor\_amka = doctor.doctor\_amka

WHERE appointments.diagnosis IS NOT null) as res

# Ερώτημα 3

Α.

CREATE FUNCTION public.update\_medical\_folder()

RETURNS trigger

LANGUAGE 'plpgsql'

COST 100.0

VOLATILE NOT LEAKPROOF

AS $BODY$

DECLARE

trig\_patient bigint;

trig\_cure text;

trig\_drug\_id integer;

BEGIN

IF (TG\_OP = 'INSERT') THEN

trig\_patient = new.patient;

trig\_cure = new.cure;

trig\_drug\_id = new.drug\_id;

INSERT INTO medical\_folder\_final(patient, cure, drug\_id)

VALUES (trig\_patient, trig\_cure, trig\_drug\_id);

END IF;

END;

CREATE TRIGGER update\_medical\_folder\_final

AFTER INSERT

ON public.medical\_folder\_final

FOR EACH ROW

WHEN appointments.diagnosis IS NOT null

EXECUTE PROCEDURE public.update\_medical\_folder();

Β.

CREATE FUNCTION public.cursors()

RETURNS table

LANGUAGE 'sql'

AS $function$

DECLARE

curs CURSOR FOR

SELECT diagnosis

FROM appointments

WHERE date > CURRENT\_DATE - interval '7' day;

BEGIN

RETURN curs;

END;

$function$;

ALTER FUNCTION public.cursors()

OWNER TO postgres;

# Ερώτημα 4

**<?php**

**class** TableRows **extends** RecursiveIteratorIterator {

**function** \_\_construct($it) {

**parent**::*\_\_construct*($it, **self**::***LEAVES\_ONLY***);

}

**function** *current*() {

**return "<td style=''>"** . **parent**::*current*(). **"</td>"**;

}

**function** beginChildren() {

**echo "<tr>"**;

}

**function** endChildren() {

**echo "</tr>"** . **"\n"**;

}

}

$myPDO = **new** PDO(**'pgsql:host=localhost; dbname=Databases17'**, **'postgres'**, **'123456'**);

$myPDO->setAttribute(PDO::***ATTR\_ERRMODE***, PDO::***ERRMODE\_EXCEPTION***);

$sql=**"SELECT res.count, drugs.name FROM ((SELECT drug\_id, COUNT(\*) FROM medical\_folder\_final GROUP BY drug\_id ORDER BY COUNT DESC LIMIT 1) UNION ALL(SELECT drug\_id, COUNT(\*) FROM medical\_folder\_final GROUP BY drug\_id ORDER BY COUNT ASC LIMIT 1)) as res INNER JOIN drugs ON res.drug\_id = drugs.id;"**;

$result = $myPDO->prepare($sql);

$result->execute();

**?>**

<**div class="container"** >

<**table**>

**<?php**

$result->setFetchMode(PDO::***FETCH\_ASSOC***);

**foreach**(**new** TableRows(**new** RecursiveArrayIterator($result->fetchAll())) **as** $k=>$v) {

**echo** $v;

}

$myPDO= **null**;

**?>**

</**table**>

<**table**>

**<?php**

$sql=**"SELECT diagnosis FROM appointments WHERE date > CURRENT\_DATE - interval '7' day; "**;

$result = $myPDO->prepare($sql);

$result->execute();

$result->setFetchMode(PDO::***FETCH\_ASSOC***);

**foreach**(**new** TableRows(**new** RecursiveArrayIterator($result->fetchAll())) **as** $k=>$v) {

**echo** $v;

}

$myPDO= **null**;

**?>**

</**table**>

<**table**>

**<?php**

$sql=**"SELECT res.\* FROM (SELECT \* FROM appointments WHERE EXTRACT(month FROM date) = EXTRACT(month FROM CURRENT\_DATE)) as res INNER JOIN doctor ON res.doctor\_amka = doctor.doctor\_amka WHERE doctor.name = 'X name' AND doctor.surname = 'X surname'"**;

$result = $myPDO->prepare($sql);

$result->execute();

$result->setFetchMode(PDO::***FETCH\_ASSOC***);

**foreach**(**new** TableRows(**new** RecursiveArrayIterator($result->fetchAll())) **as** $k=>$v) {

**echo** $v;

}

$myPDO= **null**;

**?>**

</**table**>

<**table**>

**<?php**

$sql=**" SELECT pat3.\* FROM ((SELECT patient\_amka, COUNT(\*) FROM (SELECT DISTINCT patient\_amka, doctor\_amka FROM appointments) as pat GROUP BY patient\_amka) as pat2 INNER JOIN patient ON pat2.patient\_amka = patient.patient\_amka) as pat3 WHERE count > 1"**;

$result = $myPDO->prepare($sql);

$result->execute();

$result->setFetchMode(PDO::***FETCH\_ASSOC***);

**foreach**(**new** TableRows(**new** RecursiveArrayIterator($result->fetchAll())) **as** $k=>$v) {

**echo** $v;

}

$myPDO= **null**;

**?>**

</**table**>

<**table**>

**<?php**

$sql=**" SELECT departments.name FROM departments INNER JOIN (SELECT dep.id, COUNT(\*) FROM appointments INNER JOIN doctor as doc ON appointments.doctor\_amka = doc.doctor\_amka INNER JOIN departments as dep ON doc.speciality = dep.id GROUP BY dep.id ORDER BY COUNT DESC LIMIT 1) as res ON res.id = departments.id"**;

$result = $myPDO->prepare($sql);

$result->execute();

$result->setFetchMode(PDO::***FETCH\_ASSOC***);

**foreach**(**new** TableRows(**new** RecursiveArrayIterator($result->fetchAll())) **as** $k=>$v) {

**echo** $v;

}

$myPDO= **null**;

**?>**

</**table**>

<**table**>

**<?php**

$sql=**"SELECT COUNT(res.patient\_amka)/15 FROM (SELECT DISTINCT patient\_amka, doctor.speciality FROM appointments INNER JOIN doctor ON appointments.doctor\_amka = doctor.doctor\_amka WHERE appointments.diagnosis IS NOT null) as res "**;

$result = $myPDO->prepare($sql);

$result->execute();

$result->setFetchMode(PDO::***FETCH\_ASSOC***);

**foreach**(**new** TableRows(**new** RecursiveArrayIterator($result->fetchAll())) **as** $k=>$v) {

**echo** $v;

}

$myPDO= **null**;

**?>**

</**table**>

</**div**>