



Εργασία στο μάθημα Βάσεις Δεδομένων

Παραδοτέο 3

Ομάδα 37

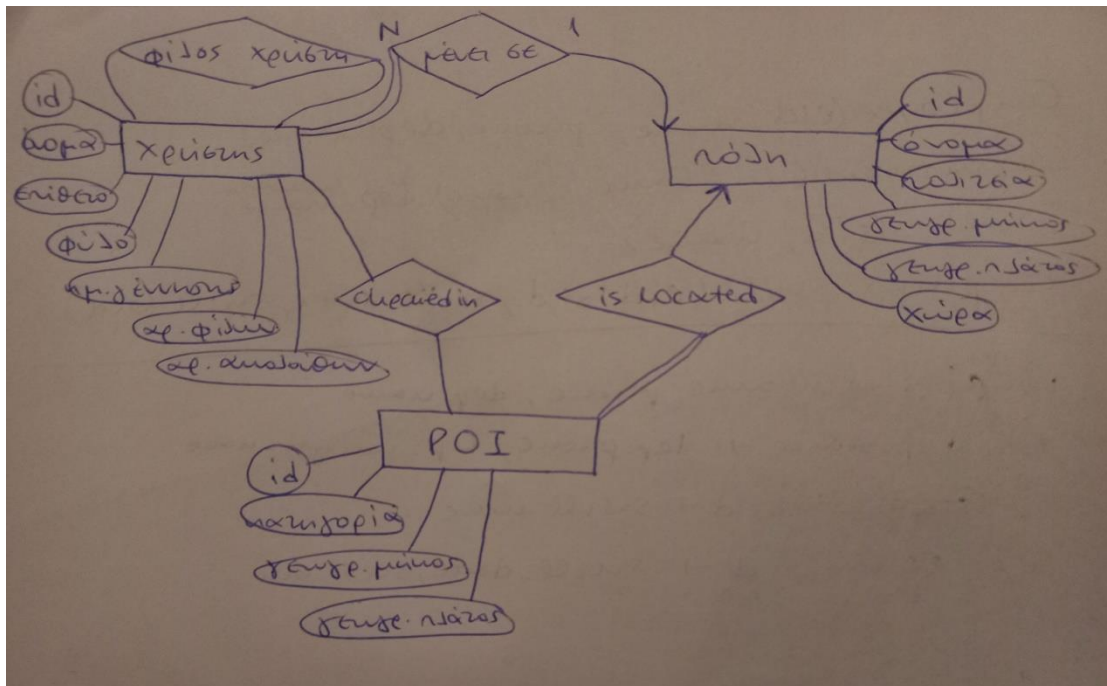
ΤΣΑΚΑΛΟΥ ΑΝΑΣΤΑΣΙΑ 2022201900226

PENTOYLA ΒΑΣΙΛΙΚΗ 2022201900197

1^ο Παραδοτέο

Ερώτημα 1

1) Σχεδίαση Μοντέλου Οντοτήτων-Συσχετίσεων



2) Αιτιολόγηση πληθικότητας και συμμετοχής των αντιστοιχίσεων

Αρχικά, έχουμε 3 οντότητες, τον “Χρήστη”, την “Πόλη”, και τα “Σημεία Ενδιαφέροντος”.

Στην οντότητα του Χρήστη, έχουμε προσθέσει τα χαρακτηριστικά που γράφει η εκφώνηση “id”, “όνομα”, “επίθετο”, “φύλο”, “ημερομηνία γεννήσεως”, “αριθμός φίλων”, και “αριθμός ακολούθων”.

Στην οντότητα της Πόλης, έχουμε προσάψει τα χαρακτηριστικά “id”, “όνομα”, “πολιτεία”, “χώρα”, “γεωγραφικό μήκος”, και “γεωγραφικό πλάτος”.

Στην οντότητα των Σημείων Ενδιαφέροντος, έχουμε προσθέσει τα γνωρίσματα “id”, “κατηγορία”, “γεωγραφικό μήκος”, και “γεωγραφικό πλάτος”.

Παρακάτω γράφουμε τι ισχύει στον κόσμο του συστήματος βάσεων δεδομένων μας, που προκύπτουν από την εκφώνηση, μαζί με διάφορες υποθέσεις και περιορισμούς που έχουμε προσθέσει εμείς.

Ο κάθε χρήστης συνδέεται (εδώ συγκεκριμένα, κατοικεί) σε μία πόλη ακριβώς σύμφωνα με την εκφώνηση, και έχουμε υποθέσει πως μία πόλη μπορεί να κατοικείται από έναν ή πολλούς χρήστες. Οπότε ο τύπος της πληθικότητας που περιγράφεται είναι ο N-1, επειδή ένας χρήστης μένει ακριβώς σε μία πόλη, αλλά μία πόλη μπορεί να κατοικείται από πολλούς χρήστες. Επίσης, αφού ο κάθε χρήστης πρέπει σίγουρα να κατοικεί σε μία πόλη, τότε είναι ολική η συμμετοχή της αντιστοίχισης του Χρήστη με την συσχέτιση Κατοικεί. Όμως, δεν είναι απαραίτητο πως η κάθε πόλη κατοικείται από έναν χρήστη. Μπορεί να υπάρχει μία πόλη στο σύστημα μας που να μην κατοικείται από κάποιον που να είναι χρήστης. Οπότε είναι μερική η αντιστοίχιση της Πόλης με την συσχέτιση Κατοικεί.

Ακόμα, περιγράφεται στην εκφώνηση πως ο κάθε Χρήστης μπορεί να έχει βρεθεί σε Σημεία Ενδιαφέροντος. Αυτόματα, αυτό σημαίνει πως πρέπει να υπάρχει η συσχέτιση έστω “checked in” που συνδέει τους Χρήστες με τα Σημεία Ενδιαφέροντος. Εφόσον ο κάθε χρήστης μπορεί να έχει βρεθεί σε σημείο ενδιαφέροντος, δηλαδή μπορεί και να μην έχει βρεθεί απαραίτητα, σημαίνει ότι έχουμε μερική συμμετοχή, καθώς ένα στιγμιότυπο από τους Χρήστες μπορεί να μην έχει κάνει check in σε κάποιο από τα Σημεία Ενδιαφέροντος που έχουμε στο σύστημα. Επίσης, υποθέτουμε πως όχι όλα τα σημεία ενδιαφέροντος έχουν επισκεφτεί σίγουρα από κάποιον χρήστη, και επομένως θα έχουμε και εδώ μερική συμμετοχή, από τα Σημεία Ενδιαφέροντος προς τη συσχέτιση checked in. Η checked in έχει τα χαρακτηριστικά day, month, year, hour, minutes, seconds. Όσον αφορά την πληθικότητα της συσχέτισης με τις οντότητες, είναι πολλά-προς-πολλά, καθώς ένας χρήστης μπορεί να έχει κάνει check in πολλά σημεία ενδιαφέροντος, και παράλληλα ένα σημείο ενδιαφέροντος να έχει γίνει checked in από πολλούς χρήστες.

Προχωρώντας, αναφέρεται πως οι χρήστες μπορούν να έχουν φίλους άλλους χρήστες. Για αυτό φτιάχνουμε την συσχέτιση “Φίλος χρήστη” που συνδέεται μόνο με τον “Χρήστη”. Η αντιστοίχιση είναι πολλά-προς-πολλά καθώς ο χρήστης μπορεί να έχει πολλούς φίλους, και ο κάθε φίλος του χρήστη το ίδιο.

Έπειτα, αναφορικά με την Πόλη, από την εκφώνηση έχουμε ότι η κάθε πόλη πρέπει να σχετίζεται με τα σημεία ενδιαφέροντος που βρίσκονται σε αυτήν. Οπότε έχουμε την συσχέτιση έστω “is located”, που συνδέεται με την οντότητα της πόλης και των σημείων ενδιαφέροντος. Η σύνδεση του σημείου ενδιαφέροντος με το is located μπορεί να εκφραστεί ως “Το σημείο ενδιαφέροντος βρίσκεται στην πόλη”, και η σύνδεση της πόλης με το is located “Στη πόλη βρίσκεται το σημείο ενδιαφέροντος”. Η πρώτη σύνδεση, αφορά ολική συμμετοχή καθώς όλα τα σημεία ενδιαφέροντος βρίσκονται σε μία πόλη, και η δεύτερη σύνδεση αφορά μερική συμμετοχή, γιατί μπορεί σε κάποια πόλη να μη βρίσκεται ένα σημείο ενδιαφέροντος, δηλαδή να μην υπάρχει. Επίσης, η πληθικότητα της είναι πολλά-προς-ένα, επειδή ένα σημείο ενδιαφέροντος μπορεί να υπάρχει σε μία μόνο πόλη, και σε μία πόλη να υπάρχουν πολλά σημεία ενδιαφέροντος.

Ερώτημα 2

1) Μοντέλο Οντοτήτων - Συσχετίσεων σε Σχεσιακό Μοντέλο

Αρκεί να πάρουμε τα ονόματα των οντοτήτων και να τα βάλουμε ως όνομα της σχέσης, και μετά να βάλουμε ως γνωρίσματα, τα γνωρίσματα που έχει η οντότητα.

Χρήστες(id, όνομα, επίθετο, φύλο, ημερ. Γέννησης, αριθμός φίλων, αριθμός ακολούθων)

Πόλεις(id, όνομα, πολιτεία, χώρα, γεωγρ. μήκος, γεωγρ. πλάτος)

POI(id, κατηγορία, γεωγρ. μήκος, γεωγρ. πλάτος)

2) Συναρτησιακές εξαρτήσεις

Στη σχέση Χρήστες(id, όνομα, επίθετο, φύλο, ημερ. Γέννησης, αριθμός φίλων, αριθμός ακολούθων):

id → όνομα, επίθετο, φύλο, ημερ. γέννησης, αριθμός φίλων, αριθμός ακολούθων

Στη σχέση Πόλεις(id, όνομα, πολιτεία, χώρα, γεωγρ. μήκος, γεωγρ. πλάτος):

id → όνομα, πολιτεία, χώρα, γεωγρ. μήκος, γεωγρ. πλάτος

πολιτεία → χώρα

γεωγρ. μήκος, γεωγρ. πλάτος → πολιτεία

Στη σχέση POI(id, κατηγορία, γεωγρ. μήκος, γεωγρ. πλάτος):

id → κατηγορία, γεωγρ. μήκος, γεωγρ. πλάτος

3) Τύπος κανονικής μορφής

Η σχέση Χρήστες(id, όνομα, επίθετο, φύλο, ημερ. Γέννησης, αριθμός φίλων, αριθμός ακολούθων) είναι σε BCNF καθώς το αριστερό γνώρισμα της (μοναδικής, εδώ συγκεκριμένα) συναρτησιακής εξάρτησής της είναι υπερκλειδί.

Η σχέση ΡΟΙ(id, κατηγορία, γεωγρ. μήκος, γεωγρ. πλάτος) είναι επίσης σε BCNF για τον ίδιο λόγο, το αριστερό γνώρισμα της (μόνης) συναρτησιακής της εξάρτησης, είναι υπερκλειδί.

Εξετάζουμε την σχέση Πόλεις(id, όνομα, πολιτεία, χώρα, γεωγρ. μήκος, γεωγρ. πλάτος). Πράγματι, η πρώτη συναρτησιακή εξάρτηση της σχέσης τηρεί τον κανόνα του BCNF, δηλαδή είναι υπερκλειδί το αριστερό γνώρισμα της. Όμως, πηγαίνοντας στην επόμενη συναρτησιακή εξάρτηση, διαπιστώνουμε ότι το αριστερό γνώρισμα δεν είναι υπερκλειδί (πολιτεία \rightarrow χώρα). Οπότε σίγουρα η σχέση δεν είναι σε BCNF.

Ελέγχουμε αν είναι σε 3NF:

Κοιτάζουμε αν το δεξί γνώρισμα της συναρτησιακής εξάρτησης είναι τμήμα υποψήφιου κλειδιού. Είναι τμήμα υποψήφιου κλειδιού, και από την πρώτη συναρτησιακή εξάρτηση έχουμε ότι το αριστερό γνώρισμα είναι υπερκλειδί. Οπότε προχωράμε στην τελευταία συναρτησιακή εξάρτηση, και βλέπουμε ότι και η πολιτεία είναι τμήμα υποψήφιου κλειδιού, δηλαδή το δεξί γνώρισμα. Άρα η σχέση είναι σε 3NF.

4) Μετατροπή σε BCNF ή 3NF

Η σχέση Πόλεις(id, όνομα, πολιτεία, χώρα, γεωγρ. μήκος, γεωγρ. πλάτος) μόνο δεν είναι σε BCNF, οπότε ακολουθούμε τον αλγόριθμο της Ανάλυσης:

Έχουμε: Πόλεις(id, όνομα, πολιτεία, χώρα, γεωγρ. μήκος, γεωγρ. πλάτος)

id \rightarrow όνομα, πολιτεία, χώρα, γεωγρ. μήκος, γεωγρ. πλάτος

(Έστω I \rightarrow A B C D E)

πολιτεία \rightarrow χώρα

(Έστω B \rightarrow C)

γεωγρ. μήκος, γεωγρ. πλάτος \rightarrow πολιτεία

(Έστω DE \rightarrow B)

Εκτελώ τον αλγόριθμο για την συναρτησιακή εξάρτηση B \rightarrow C

R1 = BC

R2 = R – (BC – B) = R – C = I A B C D E – C = I A B D E

Δηλαδή

$F1 = BC, F2 = IABDE$

$F1 \cup F2 = F$ οπότε τερματίζει ο αλγόριθμος της Ανάλυσης.

Εκτελώ τον αλγόριθμο για την συναρτησιακή εξάρτηση $DE \rightarrow B$

$R1 = BDE$

$R2 = R - (BDE - B) = R - DE = I A B C D E - DE = I A B C$

Δηλαδή

$F1 = BDE, F2 = IABC$

$F1 \cup F2 = F$ οπότε τερματίζει ο αλγόριθμος της Ανάλυσης.

Οπότε μετατρέψαμε την σχέση από 3NF σε BCNF.

5) Εντολές για την δημιουργία του πίνακα user

```
CREATE TABLE IF NOT EXISTS user (
```

```
    user_id varchar(30),
```

```
    name varchar(50),
```

```
    lastname varchar(50),
```

```
    gender varchar(20),
```

```
    birthday varchar(30),
```

```
    friendsno varchar(100),
```

```
    followersno varchar(100)
```

```
    PRIMARY KEY (user_id)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS city (
```

```
    city_id varchar(40),
```

```
city_name varchar(40),  
state varchar(40),  
country varchar(40),  
latitude varchar(100),  
longitude varchar(100),  
  
PRIMARY KEY (city_id)  
  
);
```

```
CREATE TABLE IF NOT EXISTS poi (  
  
    poi_id varchar(30),  
    category varchar(50),  
    lat varchar(50),  
    lon varchar(20),  
  
    PRIMARY KEY (poi_id)  
  
);
```

2^ο Παραδοτέο

Ερώτημα 3

1. Εισαγωγή

Αρχικά, έχουμε φτιάξει 5 .php αρχεία, που ένα μεγάλο μέρος του κώδικα προέρχεται από τα 3 .php αρχεία που βρίσκονται ανεβασμένα στο Teams. Η άσκηση αναφέρει να εισάγουμε στοιχεία για τον Χρήστη, Πόλη, και Σημεία Ενδιαφέροντος ξεχωριστά, οπότε έχουμε βάλει την επιλογή να διαλέξει ο χρήστης σε ποια σχέση θέλει να εισάγει μία νέα εγγραφή. Οπότε από το αρχείο php-insert.php, φτιάχνουμε τα php-insert-users.php, php-insert-cities.php, php-insert-fois.php. Αυτά θα είναι και τα αντικείμενα της ιστοσελίδας όταν ο χρήστης επιλέγει την εισαγωγή στη σχέση του Χρήστη, Πόλης, ή Σημεία Ενδιαφέροντος αντίστοιχα. Παίρνουμε τα .php αρχεία καθώς και το .css αρχείο, και τα μεταφέρουμε μέσω SFTP στον υπολογιστή του Χίλωνα (hilon.dit.uop.gr), στον φάκελο της ομάδας μας, και μετά στον φάκελο public_html.

Μετά, φτιάχνουμε τις σχέσεις users, cities, fois, που είναι οι 3 βασικές οντότητες από το Παραδοτέο 1, και βάζουμε τα γνωρίσματα που αντιστοιχούν στο καθένα, σύμφωνα με το διάγραμμα που είχαμε φτιάξει.

Έπειτα, πρέπει να φτιάξουμε τις συσχετίσεις των σχέσεων. Φτιάχνουμε τις checkins, users_relations, located, lives. Είναι όλες οι συσχετίσεις που βλέπουμε και στο διάγραμμα. Η users_relations αφορά την αυτοσυσχέτιση του χρήστη με άλλους χρήστες, δηλαδή έστω φίλους του, όπως είχαμε ορίσει στο διάγραμμα.

```
CREATE TABLE users (  
    user_id_real SERIAL PRIMARY KEY,  
    user_id INTEGER,  
    first_name varchar(40),  
    last_name varchar(40),  
    gender varchar(10),  
    birth_date date,
```



```
followers_no INTEGER
```

```
);
```

```
CREATE TABLE cities (
```

```
city_id_real SERIAL PRIMARY KEY,
```

```
city_name varchar(40),
```

```
state_name varchar(40),
```

```
country_name varchar(40),
```

```
latitude REAL,
```

```
longitude REAL
```

```
);
```

```
CREATE TABLE pois (
```

```
pid_real SERIAL PRIMARY KEY,
```

```
pid varchar(50),
```

```
category varchar(50),
```

```
latitude REAL,
```

```
longitude REAL
```

```
);
```

```
CREATE TABLE users_relations (
```

```
user1 INTEGER,
```

```
user2 INTEGER,
```

```
FOREIGN KEY (user1) REFERENCES users(user_id_real) ON DELETE CASCADE,
```

```
FOREIGN KEY (user2) REFERENCES users(user_id_real) ON DELETE CASCADE
```

```
);
```

```
CREATE TABLE checkins (  
  ch_user_id_real INTEGER,  
  ch_pid_real INTEGER,  
  day INTEGER,  
  month INTEGER,  
  year INTEGER,  
  hours INTEGER,  
  min INTEGER,  
  sec INTEGER,  
  FOREIGN KEY (ch_user_id_real) REFERENCES users(user_id_real) ON DELETE CASCADE,  
  FOREIGN KEY (ch_pid_real) REFERENCES pois(pid_real) ON DELETE CASCADE  
);
```

```
CREATE TABLE located (  
  city INTEGER,  
  poi INTEGER,  
  FOREIGN KEY (city) REFERENCES cities(city_id_real) ON DELETE CASCADE,  
  FOREIGN KEY (poi) REFERENCES pois(pid_real) ON DELETE CASCADE  
);
```

```
CREATE TABLE lives (  
  resident_id INTEGER,  
  lives_city_id INTEGER,  
  FOREIGN KEY (resident_id) REFERENCES users(user_id_real) ON DELETE CASCADE,  
  FOREIGN KEY (lives_city_id) REFERENCES cities(city_id_real) ON DELETE CASCADE  
);
```

2. Εισαγωγή από αρχείο κειμένου

Προσωρινές σχέσεις

Βλέπουμε τα 5 .csv αρχεία και τα περιεχόμενα τους, και βάσει αυτών θα φτιάξουμε τις 5 προσωρινές σχέσεις. Θα βάλουμε ως γνωρίσματα, τα πεδία/στήλες που βλέπουμε στο περιεχόμενο τους. Δεν θα χρειαστεί να βάλουμε σε καμία σχέση πρωτεύον κλειδί, εφόσον είναι προσωρινές και θα βάλουμε μετά, όταν γίνουν τελικές, τα πρωτεύοντα κλειδιά τους και τα references στις συσχετίσεις που χρειάζονται.

```
CREATE TABLE users_temp(  
  
    user_id INTEGER,  
  
    first_name varchar(40),  
  
    last_name varchar(40),  
  
    gender varchar(10),  
  
    birth_date date,  
  
    city_name varchar(40),  
  
    state_name varchar(40),  
  
    country_name varchar(40),  
  
    followers_no INTEGER  
  
);
```

```
CREATE TABLE cities_temp(  
  
    city_name varchar(40),  
  
    state_name varchar(40),  
  
    country_name varchar(40),  
  
    latitude REAL,  
  
    longitude REAL
```

```
);
```

```
CREATE TABLE pois_temp(  
    pid varchar(50),  
    category varchar(50),  
    city_name varchar(40),  
    state_name varchar(40),  
    country_name varchar(40),  
    latitude REAL,  
    longitude REAL  
);
```

```
CREATE TABLE checkins_temp(  
    ch_user_id INTEGER,  
    ch_pid varchar(50),  
    day INTEGER,  
    month INTEGER,  
    year INTEGER,  
    hours INTEGER,  
    min INTEGER,  
    sec INTEGER  
);
```

```
CREATE TABLE users_relations_temp(  
    id1 INTEGER,
```

id2 INTEGER

);

Τώρα, για μεταφερθούν τα δεδομένα από τα .csv αρχεία στις προσωρινές σχέσεις, χρησιμοποιούμε τις εντολές \copy που παρέχει η psql. Κάνουμε:

```
\copy checkins_temp FROM /home/Data/2021-22/checkins.csv WITH DELIMITER ','
```

```
\copy users_relations_temp FROM /home/Data/2021-22/users_relations.csv WITH  
DELIMITER ','
```

```
\copy users_temp FROM /home/Data/2021-22/users.csv WITH DELIMITER ','
```

```
\copy cities_temp FROM /home/Data/2021-22/cities.csv WITH DELIMITER ','
```

```
\copy pois_temp FROM /home/Data/2021-22/pois.csv WITH DELIMITER ','
```

Και βγάζει μήνυμα επιτυχίας τον αριθμό καταχωρήσεων/εγγραφών που έχει η σχέση τελικά.

Μετά, για να γίνουν τελικές οι σχέσεις, αρκεί να εισάγουμε τις εντολές INSERT INTO, SELECT, και FROM. Οπότε εισάγουμε:

```
INSERT INTO users(user_id, first_name, last_name, gender, birth_date, followers_no)
```

```
SELECT user_id, first_name, last_name, gender, birth_date, followers_no FROM  
users_temp;
```

```
INSERT INTO cities(city_name, state_name, country_name, latitude, longitude)
```

```
SELECT city_name, state_name, country_name, latitude, longitude FROM cities_temp;
```

```
INSERT INTO pois(pid, category, latitude, longitude)
```

```
SELECT pid, category, latitude, longitude FROM pois_temp;
```

```
INSERT INTO checkins(ch_user_id_real,ch_pid_real,day,month,year,hours,min,sec)
```

```
SELECT users.user_id_real, pois.pid_real, checkins_temp.day, checkins_temp.month,  
checkins_temp.year, checkins_temp.hours, checkins_temp.min, checkins_temp.sec
```

```
FROM users, pois, checkins_temp  
  
WHERE checkins_temp.ch_user_id=users.user_id  
  
AND checkins_temp.ch_pid=pois.pid;
```

```
INSERT INTO located(city, poi)  
  
SELECT cities.city_id_real, pois.pid_real  
  
FROM pois, pois_temp, cities  
  
WHERE cities.city_name=pois_temp.city_name  
  
AND cities.state_name=pois_temp.state_name  
  
AND cities.country_name=pois_temp.country_name  
  
AND pois.pid=pois_temp.pid;
```

```
INSERT INTO lives(resident_id, lives_city_id)  
  
SELECT users.user_id_real, cities.city_id_real  
  
FROM cities_temp,cities,users,users_temp  
  
WHERE cities_temp.city_name=cities.city_name  
  
AND cities_temp.state_name=cities.state_name  
  
AND users_temp.user_id=users.user_id  
  
AND users_temp.city_name=cities.city_name  
  
AND cities_temp.city_name=cities.city_name;
```

3^ο Παραδοτέο

1. Παρουσίαση σημείων ενδιαφέροντος.

Εμφανίζουμε το γεωγραφικό πλάτος, γεωγραφικό μήκος, κατηγορία, πόλη, και χώρα των σημείων ενδιαφέροντος. Συνδέουμε τη συσχέτιση located με την pois και την cities.

Query:

```
SELECT DISTINCT pois.latitude, pois.longitude, pois.category,  
cities.city_name, cities.country_name  
  
FROM pois, cities, located  
  
WHERE pois.pid_real=located.poi  
  
AND cities.city_id_real=located.city;
```

Σε log file (1.txt) βρίσκονται όλες οι εγγραφές που απαντούν στο παραπάνω query.

2. Παρουσίαση χρηστών.

1. Query:

```
SELECT users.first_name, users.last_name, users.gender, users.followers_no  
  
FROM users;
```

2. Query:

```
SELECT cities.country_name, cities.state_name, cities.city_name, users.last_name,  
users.first_name  
  
FROM users, cities, lives  
  
WHERE lives.resident_id=users.user_id_real  
  
AND lives.lives_city_id=cities.city_id_real  
  
ORDER BY cities.city_name;
```

Στα log files 2a.txt και 2b.txt βρίσκονται τα αποτελέσματα της ερώτησης.

5. Απαντήσεις σε ερωτήματα.

3. Παρουσιάστε τα ονόματα και τα επίθετα των χρηστών που έχουν επισκεφτεί κάποιο σημείο ενδιαφέροντος κατηγορίας Χ.

Έχουμε βάλει να εμφανίζει το όνομα και επίθετο των χρηστών που έχουν επισκεφτεί για παράδειγμα το Πάρκο. Εμφανίζουμε και την κατηγορία, το πάρκο στην προκειμένη περίπτωση, για να φαίνεται πως πράγματι είναι η κατηγορία που ζητάει η συνθήκη.

Query:

```
SELECT DISTINCT users.first_name, users.last_name,
               pois.category
FROM pois, checkins, users
WHERE  pois.category='Πάρκο'
AND    checkins.ch_pid_real=pois.pid_real;
AND    checkins.ch_user_id_real=users.user_id_real;
```

Απάντηση του query:

first_name	last_name	category
Margaret	Newman	Πάρκο
Eva	Simpson	Πάρκο
Frances	Gibson	Πάρκο
Bailey	Campbell	Πάρκο
Evie	Mitchell	Πάρκο
Delaney	Watson	Πάρκο
Elena	Gill	Πάρκο
Remington	Butler	Πάρκο

Talia	Porter	Πάρκο
Eliana	Webster	Πάρκο
Remington	Reed	Πάρκο
Muhammad	Anderson	Πάρκο
Madelyn	Fleming	Πάρκο
Amy	Meyer	Πάρκο
Theodore	Hanson	Πάρκο
Carson	Holmes	Πάρκο
Brooke	Hanson	Πάρκο
Samantha	White	Πάρκο
Tessa	Briggs	Πάρκο
Leonardo	Ortiz	Πάρκο
Grant	Brooks	Πάρκο
Clara	Clark	Πάρκο
Jonah	Allen	Πάρκο
Kane	Brooks	Πάρκο
Liam	Wood	Πάρκο

Η ολόκληρη απάντηση είναι σε log file (5-3.txt) στον hilon.dit.uop.gr στην ομάδα μας.

5. Παρουσιάστε τα σημεία ενδιαφέροντος με περισσότερες από 1000 επισκέψεις, ταξινομημένα με βάση τον αριθμό των επισκέψεών τους.

Εμφανίζουμε τους serial κωδικούς των σημείων ενδιαφέροντος, για επαλήθευση. Επιπλέον, παρουσιάζουμε την κατηγορία των σημείων ενδιαφέροντος καθώς και το πλήθος επισκέψεων, για να φαίνεται ότι είναι πάνω από 1000 συνολικές επισκέψεις από χρήστες.

Query:

```
SELECT pois.pid_real, pois.category, COUNT(pois.pid_real)
FROM checkins, pois
WHERE checkins.ch_user_id_real=pois.pid_real
GROUP BY pois.pid_real, pois.category
HAVING COUNT(pois.pid_real)>1000
ORDER BY COUNT(pois.pid_real) desc;
```

Απάντηση του query:

pid_real	category	count
9728	Εστιατόριο Τάκο	2026
4185	Εστιατόριο Θαλασσινών	1435
2960	Μπαρ	1224
3292	Κατάστημα Με Αντίκες	1062

(4 rows)

6. Παρουσιάστε την πόλη που ζουν οι περισσότεροι χρήστες.

Παρουσιάζουμε το όνομα της πόλης με τους περισσότερους κατοίκους και το πλήθος των κατοίκων της.

Query:

```
SELECT DISTINCT cities.city_name, COUNT(lives.lives_city_id)
FROM users, lives, cities
WHERE users.user_id_real=lives.resident_id
AND cities.city_id_real=lives.lives_city_id
GROUP BY cities.city_name
ORDER BY COUNT(lives.lives_city_id) desc
LIMIT 1;
```

Απάντηση του query:

```
city_name | count
-----+-----
New York  |  3808
(1 row)
```

7. Παρουσιάστε τον αριθμό επισκέψεων ανά χρήστη.

Ως στοιχεία του χρήστη θεωρούμε το όνομα και το επώνυμο, καθώς και το serial id που ξεχωρίζει τον κάθε χρήστη, οπότε εμφανίζουμε αυτά τα χαρακτηριστικά, καθώς και τον αριθμό επισκέψεων ανά χρήστη, σε φθίνουσα σειρά. Το πλήθος των ίδιων id στην checkins σχέση είναι το πλήθος των επισκέψεων, οπότε ταξινομούμε με βάση το πλήθος αυτού του γνωρίσματος.

Query:

```
SELECT users.user_id_real, users.first_name, users.last_name,  
COUNT(checkins.ch_user_id_real)  
  
FROM users, checkins  
  
WHERE users.user_id_real=checkins.ch_user_id_real  
  
GROUP BY users.user_id_real  
  
ORDER BY COUNT(checkins.ch_user_id_real) desc;
```

Απάντηση του query:

user_id_real	first_name	last_name	count
9728	Nathan	Ross	2026
4185	Collins	Nelson	1435
2960	Octavia	Klein	1224
3292	Madelyn	Fleming	1062
6204	Matthew	Hamilton	1000
5484	Joshua	Briggs	960
2217	Saylor	Young	921
4928	Harrison	Richardson	909
5898	Edison	Hanson	891
2872	Justin	Thompson	849
9969	Kaiden	Morris	816
4404	Jayden	Fleming	762
7050	Melissa	Mitchell	702
415	Madilyn	Torres	679
4319	Emery	King	664
8275	Erik	Lambert	664
9300	Maya	Thompson	634

9133		Leon		Edwards		628
6427		Lucia		Castillo		604
4694		Waylon		Porter		603
5929		Maximus		Wright		583
7479		Jack		Price		558
3959		Aubree		Rogers		516
2958		Kaiden		Davis		509
6820		Aurora		Bailey		503
1958		Riley		Hughes		496
4572		Addison		Norris		480
6076		Matthew		Hale		480
7689		Dominic		Perez		475
6863		Grace		Cooper		464
3991		Milani		Mitchell		462
3415		Arya		Davis		451
4571		Dylan		Edwards		450
5783		Dahlia		Thomas		444
5953		Hazel		Hamilton		429
815		Felix		Gill		429
2969		Sarah		Norris		429
651		Jayden		Taylor		420
4579		Ashton		Mckinney		417
5852		Brooklyn		Houston		414
2128		Ian		Watson		403
5998		Kayson		Fuller		402
7226		Zara		Davidson		401
7987		Muhammad		Anderson		401
9416		Dylan		Hammond		399

Όλες οι εγγραφές φαίνονται στο log file 5-7.txt.

8. Παρουσιάστε τις πόλεις που έχουν σημεία ενδιαφέροντος που ανήκουν στην κατηγορία Χ.

Παρουσιάζουμε τις πόλεις με τα σημεία ενδιαφέροντος που ανήκουν στην κατηγορία έστω 'Γραφείο'. Ακόμα, εμφανίζουμε το id του σημείου ενδιαφέροντος, για να μπορούμε να τα ξεχωρίσουμε, καθώς και το id της πόλης. Κάνουμε ταξινόμηση ώστε να φαίνεται ξεκάθαρα ποια πόλη έχει ποια σημεία ενδιαφέροντος, καθώς μία πόλη (το ίδιο id) μπορεί να αντιστοιχίζεται σε πολλά σημεία ενδιαφέροντος. Οπότε ταξινομούμε με βάση το id της πόλης.

Query:

```
SELECT DISTINCT cities.city_id_real, pois.pid_real,
cities.city_name, pois.category
FROM cities, pois, located
WHERE pois.pid_real = located.poi
AND located.city=cities.city_id_real
AND pois.category='Γραφείο'
ORDER BY cities.city_id_real;
```

Απάντηση του query:

city_id_real	pid_real	city_name	category
1	61645	A-1 Trailer Park	Γραφείο
9	10812	Adobe Mountain Trailer Park	Γραφείο
11	3919	Agnew	Γραφείο
11	27901	Agnew	Γραφείο
11	28078	Agnew	Γραφείο
11	30101	Agnew	Γραφείο
11	33168	Agnew	Γραφείο

11		45275		Agnew		Γραφείο
11		48453		Agnew		Γραφείο
11		55471		Agnew		Γραφείο
11		58017		Agnew		Γραφείο
11		60410		Agnew		Γραφείο
11		62961		Agnew		Γραφείο
11		64880		Agnew		Γραφείο
11		72225		Agnew		Γραφείο
12		69554		Aguascalientes		Γραφείο
18		38662		Alamo Placita		Γραφείο
18		41385		Alamo Placita		Γραφείο
24		54533		Alexandria		Γραφείο
30		24833		Alondra Park		Γραφείο
30		33665		Alondra Park		Γραφείο
43		74285		Arbutus		Γραφείο
44		64981		Ardmore		Γραφείο
46		30649		Arlington		Γραφείο
46		72740		Arlington		Γραφείο
47		8537		Arlington		Γραφείο
47		41468		Arlington		Γραφείο
47		42613		Arlington		Γραφείο
47		48547		Arlington		Γραφείο
47		51891		Arlington		Γραφείο
47		52755		Arlington		Γραφείο
47		53114		Arlington		Γραφείο
47		53155		Arlington		Γραφείο
47		74994		Arlington		Γραφείο
47		75935		Arlington		Γραφείο
47		80140		Arlington		Γραφείο

Έχουμε το log file 5-8.txt που δείχνει όλα τα δεδομένα του αποτελέσματος.

9. Παρουσιάστε τα ονόματα των πόλεων που βρίσκονται μέσα στο ορθογώνιο που ορίζεται από τα σημεία (x1,y1) και (x2,y2).

Θα επιλέξουμε την σχέση των cities, από την οποία θα δείξουμε το όνομα της πόλης, το γεωγραφικό πλάτος, και γεωγραφικό μήκος. Η συνθήκη θα είναι αν το γεωγρ. πλάτος είναι ανάμεσα στις τιμές του 5 και 30, και αντίστοιχα αν το γεωγρ. μήκος είναι ανάμεσα στις τιμές του 10 και 25 (ενδεικτικές τιμές). Εμφανίζουμε επίσης το id των πόλεων για να μπορούμε να ψάξουμε πιο εύκολα κάθε εγγραφή που φαίνεται να ορίζεται στα σημεία αυτά, και να επαληθεύσουμε.

To query:

```
SELECT cities.city_id_real, cities.city_name,  
cities.latitude, cities.longitude  
  
FROM cities  
  
WHERE cities.latitude>5  
  
AND cities.latitude<30  
  
AND cities.longitude>10  
  
AND cities.longitude<95;
```

Η απάντηση του query:

city_id_real	city_name	latitude	longitude
3	Abu Dhabi	24.4538	54.3774
6	Ad Damman	26.4368	50.104
13	Ajman	25.3937	55.4451
14	Al-Ahmadi	29.0891	48.061
29	Al-Muharraq	26.2466	50.6164

49		Ar Rifa		26.1315		50.5503
91		Bangalore		12.9768		77.5901
295		Chennai		13.0837		80.2702
306		Chetpet		12.9224		80.0201
362		Colombo		6.93875		79.8541
427		Delhi		28.6517		77.2219
445		Doha		25.2856		51.5264
459		Dubai		25.2653		55.2925
651		Giza		29.9871		31.2118
697		Gurgaon		28.4283		77.0027
724		Hawalli		29.3363		48.0199
780		Hyderabad		17.3606		78.4741
807		Jeddah		21.581		39.1654
832		Khadki		21.2281		79.7835
855		Kuwait		29.3797		47.9736
966		Manama		26.2235		50.5822
1088		Mumbai		19.076		72.8774
1092		Muscat		23.5882		58.3829
1105		Nerkunram		12.69		79.8531
1115		New Delhi		28.6139		77.209
1136		Noida		28.5708		77.3271
1226		Pallavaram		12.9655		80.1451
1233		Panvel		18.9895		73.1222
1276		Pimpri		21.2645		77.6831
1327		Pune		18.5214		73.8545
1382		Riyadh		24.6389		46.716

(36 rows)

10. Να βρείτε τους χρήστες που επισκέφθηκαν την Ελλάδα, το καλοκαίρι (Ιούνιο, Ιούλιο και Αύγουστο) του 2012.

Εμφανίζουμε το πραγματικό id, όνομα, και επώνυμο των χρηστών που επισκέφτηκαν την Ελλάδα το καλοκαίρι του 2012. Ενώνουμε την συσχέτιση checkins με τη σχέση pois και users, καθώς και την συσχέτιση located με pois και cities. Οι επιπλέον συνθήκες αφορούν τους μήνες και τον χρόνο του καλοκαιριού που έγινε η επίσκεψη, και η χώρα που είναι η Ελλάδα εδώ συγκεκριμένα.

Query:

```
SELECT DISTINCT users.user_id_real, users.first_name,
users.last_name

FROM users, cities, pois, located, checkins

WHERE checkins.ch_user_id_real=users.user_id_real

AND pois.pid_real=checkins.ch_pid_real

AND pois.pid_real=located.poi

AND cities.city_id_real=located.city

AND checkins.month>5

AND checkins.month<9

AND checkins.year=2012

AND cities.country_name='Greece';
```

Απάντηση query:

```
user_id_real | first_name | last_name
-----+-----+-----
          9139 | Elijah    | Bates
          7675 | Cristian  | Evans
          3535 | Aylin     | Campbell
          4643 | Harley   | Mckinney
```

11. Παρουσιάστε τα ονόματα και τα επίθετα των χρηστών που βρίσκονται στις χώρες με όνομα Χ ή Υ.

Παρουσιάζουμε το πραγματικό id, όνομα, και επίθετο των χρηστών που μένουν έστω στην Γαλλία ή στο Ηνωμένο Βασίλειο. Ενώνουμε τις συσχετίσεις lives με users και cities, και μετά κάνουμε UNION ένα παρόμοιο σκετ SQL εντολών, με την διαφορά το όνομα της χώρας αυτή τη φορά.

Query:

```
(SELECT DISTINCT users.user_id_real, users.first_name,
users.last_name

FROM users, cities, lives

WHERE users.user_id_real=lives.resident_id

AND cities.city_id_real=lives.lives_city_id

AND cities.country_name='France')

UNION

(SELECT DISTINCT users.user_id_real, users.first_name,
users.last_name

FROM users, cities, lives

WHERE users. user_id_real=lives.resident_id

AND cities.city_id_real=lives.lives_city_id

AND cities.country_name='United Kingdom');
```

Απάντηση στο query:

user_id_real	first_name	last_name
2519	Theo	King
5717	Kehlani	Miller
6642	Tyler	Mitchell
2828	Violet	Boucher
8015	Kyle	Phillips
8875	Madison	Clarke
7342	Jason	Jones
265	Ella	Sanders
2936	Catherine	Allen
1150	Justin	Allen
4621	Rebecca	Bell
4558	Rylan	Briggs
8412	Freya	Hamilton
8619	Harrison	Reed
6938	Benjamin	Cook
6015	Cynthia	Cook
3054	Lisa	Harris
4144	Hadley	Lambert
5138	Mateo	Richardson
7689	Dominic	Perez
630	Mia	Cook
8517	Eric	Shaw
5926	Luke	Johnston
2640	Londyn	Howard
6889	Isaac	Murphy

6630	Dylan	Brooks
9091	Linda	Robinson
9553	Benjamin	Campbell
4418	Kimberly	Robinson
7626	Bryce	Gross
4241	Madelyn	Watkins
533	Dorothy	Edwards
2381	Margot	Lindsey
5893	Samantha	Lewis
8588	Sebastian	Lavoie
8689	Angelina	Fleming
6478	Douglas	Morgan
6141	Ashley	Gutierrez
9255	Rebecca	Kelly
1271	Riley	Campbell
9077	Jaxton	Douglas
9934	Liam	Jackson
7318	Cecilia	Adams
8205	Nolan	Morgan
6297	Cody	Hughes
8969	Grant	Simpson
5151	Cali	Larson
7554	Jerry	Robinson
5157	Ian	May
5862	Amina	Campbell
9192	Aaron	Campbell
689	Robert	Wood
5508	Roman	Castillo

(53 rows)

14. Για κάθε σημείο ενδιαφέροντος παρουσιάστε τον αριθμό των επισκέψεων που έχουν γίνει καθώς και την χώρα στην οποία βρίσκεται αν ο αριθμός των επισκέψεων είναι μεταξύ 200 και 300.

Εμφανίζουμε το πραγματικό id των σημείων ενδιαφέροντος για να γίνεται επαλήθευση, και το όνομα της χώρας καθώς και το πλήθος επισκέψεων, εφόσον είναι μεταξύ των 200 και 300. Κάνουμε την ένωση της located με pois και cities, καθώς και checkins με pois. Δεν χρειαζόμαστε τους users για αυτό το query καθώς ο αριθμός επισκέψεων προκύπτει μόνο από την checkins, και η χώρα από την located και cities.

Query:

```
SELECT pois.pid_real, pois.category, cities.country_name, COUNT(pois.pid_real)
FROM cities, pois, located, checkins
WHERE pois.pid_real=checkins.ch_pid_real
AND located.city=cities.city_id_real
AND located.poi=pois.pid_real
GROUP BY pois.pid_real, pois.category, cities.country_name
HAVING COUNT(pois.pid_real)>=200
AND COUNT(pois.pid_real)<=300;
```

Απάντηση query:

pid_real	category	country_name	count
1004	Πάρκο	United States	260
2234	Πολυκατάστημα	United States	242
2305	Πάρκο	United States	238
3065	Συνεδριακό Κέντρο	United States	202
3767	Κινηματογράφος	United States	232

4301		Κατάστημα Ηλεκτρονικών		United States		249
5317		Αεροδρόμιο		United States		222
6231		Πλατεία		United States		239
6924		Στάδιο Μπέιζμπολ		United States		292
8041		Σταθμός Τραίνου		United States		219
9546		Γέφυρα		United States		227
10330		Αεροδρόμιο		United States		231
10624		Σταθμός Τραίνου		United States		268
11204		Αεροδρόμιο		United States		242
11606		Σταθμός Τραίνου		United States		200
12461		Μετρό		United States		211
15005		Αεροδρόμιο		United States		213
15667		Αεροδρόμιο		United States		256
15950		Αεροδρόμιο		United States		235
16676		Αεροδρόμιο		United States		240
17891		Αεροδρόμιο		United States		230
19923		Αεροδρόμιο		United States		233
40118		Διαμέρισμα		United States		208
42479		Μουσικός Χώρος		United States		276
47512		Αεροδρόμιο		United States		219
84408		Σταθμός Τραίνου		United States		241

(26 rows)

15. Να παρουσιάσετε την περισσότερη και την λιγότερο δημοφιλή κατηγορία σημείων ενδιαφέροντος μαζί με τον αριθμό των επισκέψεων που έχουν πραγματοποιηθεί.

Αρχικά, κάνουμε τις συσχετίσεις checkins με pois. Η λιγότερο δημοφιλής κατηγορία είναι αυτή με το μικρότερο άθροισμα ίδιων id των pois, καθώς θα έχουν εντοπιστεί τις λιγότερες φορές (δηλαδή 1 το λιγότερο) στην checkins σχέση. Αντίστοιχα, η πιο δημοφιλής είναι αυτή με το μεγαλύτερο άθροισμα αυτών. Κάνουμε ταξινόμηση με όριο

το 1 και στις δύο περιπτώσεις ώστε να εμφανίσουμε την λιγότερο και περισσότερο δημοφιλή κατηγορία κάθε φορά.

Query:

```
(SELECT DISTINCT pois.category, SUM(pois.pid_real)
FROM pois, checkins
WHERE checkins.ch_pid_real=pois.pid_real
GROUP BY pois.category
ORDER BY SUM(pois.pid_real) asc
LIMIT 1)
UNION
(SELECT DISTINCT pois.category, SUM(pois.pid_real)
FROM pois, checkins
WHERE checkins.ch_pid_real=pois.pid_real
GROUP BY pois.category
ORDER BY SUM(pois.pid_real) desc
LIMIT 1);
```

Απάντηση query:

category	sum
Αφγανικό Εστιατόριο	19703
Οικία (προσωπική)	1513548464

(2 rows)

17. Να βρείτε τους χρήστες που έχουν επισκεφθεί την Ελλάδα και την Ισπανία, αλλά όχι την Ιταλία.

Query:

```
(SELECT DISTINCT users.user_id_real, users.first_name, users.last_name
FROM users, cities, pois, located, checkins
WHERE users.user_id_real=checkins.ch_user_id_real
AND pois.pid_real=checkins.ch_pid_real
AND cities.country_name='Greece'
AND cities.city_id_real=located.city
AND pois.pid_real=located.poi)
UNION
(SELECT DISTINCT users.user_id_real, users.first_name, users.last_name
FROM users, cities, pois, located, checkins
WHERE users.user_id_real=checkins.ch_user_id_real
AND pois.pid_real=checkins.ch_pid_real
AND cities.country_name='Spain'
AND cities.city_id_real=located.city
AND pois.pid_real=located.poi)
EXCEPT
(SELECT DISTINCT users.user_id_real, users.first_name, users.last_name
FROM users, cities, pois, located, checkins
WHERE users.user_id_real=checkins.ch_user_id_real
AND pois.pid_real=checkins.ch_pid_real
AND cities.country_name='Italy'
AND cities.city_id_real=located.city
AND pois.pid_real=located.poi);
```

Απάντηση query:

user_id_real	first_name	last_name
8001	Faith	Rogers
8300	Amy	Meyer
4418	Kimberly	Robinson
8616	Alice	Watson
6828	Raelyn	Hale
6304	Bowen	Larson
5417	Kyler	Hamilton
4980	Victoria	Bailey
6083	Adelynn	Baker
2440	Bhanupriya	Patla
3205	Margot	Barrett
4159	Ryleigh	Norman
9796	Selena	Wood
5178	Anaya	Roberts
352	Oscar	Perez
3204	Delaney	Patterson
3210	Wyatt	Long
5517	Greyson	Ortiz
9571	Camila	Allen
3598	Kevin	Patterson
1153	Dexter	Peterson
6368	Carol	Shaw
7965	Esme	Harrison
7715	Troy	Davidson
387	Tyler	Hamilton

6656	Brynn	James
854	Kayson	Allen
2139	Angelo	Hoffman
3535	Aylin	Campbell
2494	Calvin	Cox
7012	Gabriel	Clark
1079	Autumn	Richardson
1092	Rory	Mccarty
8517	Eric	Shaw
503	Vincent	Miller
247	Preston	Kim
7820	Arielle	Fowler
8244	Elijah	Gray
5583	Cooper	Hoffman
1106	Abigail	Roy
3029	Cayden	Hanson
7675	Cristian	Evans
2674	Magnolia	Simpson
209	Leilani	Houston
9886	Ashton	Morales
6359	Marcus	Robinson
3650	Madelyn	Wood
708	Marley	Ramos
5633	Ryder	Lewis