

Προχωρημένα Θέματα Βάσεων Δεδομένων

Εργασία - Υλοποίηση χωρικής βάσης δεδομένων

ΘΕΜΑ

Υλοποίηση χωρικής βάσης δεδομένων για μια δημοτική κοινότητα

Ιπποκράτης Κοτσάνης

AM: 131

Θεσσαλονίκη, 2023

Περιεχόμενα

1. Κάντε μία σύντομη περιγραφή της εφαρμογής σας και των δυνατοτήτων της.
2. Σχεδιάστε το ER-diagram (διάγραμμα οντοτήτων-συσχετίσεων) κάνοντας χρήση και των κατάλληλων pictograms. Το διάγραμμα μπορεί να γίνει σε οποιοδήποτε σχεδιαστικό πρόγραμμα, modeler ή βοηθητικό tool θέλετε.
3. Μετατρέψτε το διάγραμμα οντοτήτων-συσχετίσεων σε σχεσιακό εφαρμόζοντας τους κανόνες μετατροπής.
4. Υλοποιήστε τους αντίστοιχους πίνακες και τις συνδέσεις τους από το σχεσιακό σας μοντέλο στην πλατφόρμα SQL που χρησιμοποιείτε (π.χ. PostgreSQL + PostGIS).
5. Εισάγετε μερικά ενδεικτικά δεδομένα στους πίνακες αυτούς από το σύνολο δεδομένων που δημιουργήσατε (ή χρησιμοποιείτε τα έτοιμα σύνολα δεδομένων που επιλέξατε).
6. Κατασκευάστε μια σειρά από τουλάχιστον 7 χρήσιμα μη-χωρικά ερωτήματα, εκφράστε τα σε SQL και καταγράψτε τα. Στα ερωτήματα αυτά θα πρέπει να υπάρχει τουλάχιστον ένα ερώτημα ομαδοποίησης (GROUP BY), τουλάχιστον ένα ερώτημα σύνδεσης (JOIN), τουλάχιστον ένα ερώτημα συνάθροισης (AVG, SUM, COUNT, MIN, MAX), τουλάχιστον ένα εμφωλευμένο ερώτημα, και τουλάχιστον ένα ερώτημα με δημιουργία και χρήση όψης (VIEW).
7. Κατασκευάστε μια σειρά από τουλάχιστον 9 χρήσιμα χωρικά ερωτήματα, εκφράστε τα σε SQL και καταγράψτε τα. Στα ερωτήματα αυτά θα πρέπει να υπάρχει τουλάχιστον ένα ερώτημα υπολογισμού εμβαδών (area), τουλάχιστον ένα ερώτημα υπολογισμού ή ελέγχου αποστάσεων (distance), τουλάχιστον δύο ερωτήματα χρήσης τοπολογικού τελεστή (π.χ. touch, intersect, cross, overlap, disjoint, within, contains) μεταξύ ζεύγους χωρικών αντικειμένων (spatial join), τουλάχιστον ένα ερώτημα self spatial join με χρήση τοπολογικού τελεστή, τουλάχιστον ένα εμφωλευμένο ερώτημα, και τουλάχιστον ένα ερώτημα με δημιουργία και χρήση όψης.
8. Εκτελέστε τα ερωτήματα αυτά στο περιβάλλον διαχείρισης της βάσης σας (π.χ. στο σύστημα PostgreSQL + PostGIS), και καταγράψτε τα αποτελέσματά τους.
9. Για τα μη χωρικά ερωτήματα εμφανίστε τα αποτελέσματα σε μορφή πινάκων ή screenshots, ενώ για τα χωρικά ερωτήματα οπτικοποιήστε την εκτέλεση τους χρησιμοποιώντας ένα κατάλληλο tool (π.χ. uDig, MapServer, κλπ.).
10. Παράρτημα κώδικα.
11. Βιβλιογραφική αναφορά.

1. Περιγραφή της εφαρμογής και των απαιτήσεων

Η βάση δεδομένων περιγράφει την οργάνωση μιας δημοτικής κοινότητας. Πιο συγκεκριμένα, συσχετίζει τους εργαζομένους ενός δήμου με τις οργανωτικές δομές και τα κτίρια στα οποία εργάζονται. Παράλληλα, η βάση περιλαμβάνει μη χωρικά δεδομένα όπως οι ηλεκτρονικοί υπολογιστές, η τηλεφωνία, τα UPS, οι servers και τα πληροφοριακά συστήματα που χρησιμοποιούνται στον δήμο. Επίσης, περιγράφει και χωρικά δεδομένα όπως οι λίμνες που βρίσκονται στον δήμο και πιθανή δραστηριότητα ψαρέματος από τους εργαζομένους αλλά και τους δρόμους και τις δημοτικές αθλητικές εγκαταστάσεις.

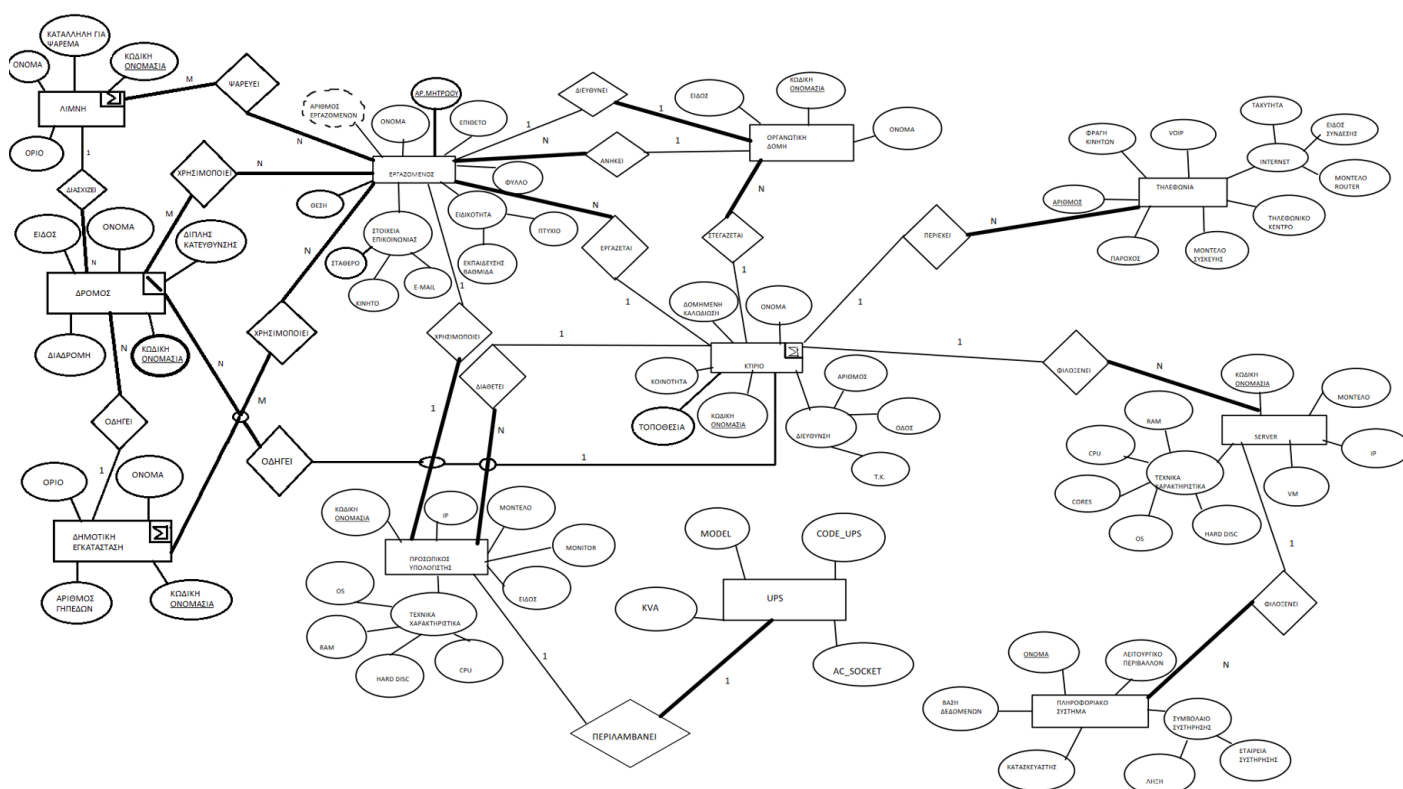
Παρακάτω παρουσιάζονται όλες οι οντότητες της χωρικής βάσης δεδομένων καθώς και οι απαιτήσεις του συστήματος.

- Ο δήμος είναι οργανωμένος σε οργανωτικές δομές. Κάθε οργανωτική δομή έχει ένα όνομα, κωδικό και τι είδος είναι. Κάθε οργανωτική δομή διευθύνεται από έναν εργαζόμενο.
- Κάθε οργανωτική δομή ανήκει σε ένα κτίριο το οποίο έχει όνομα, κοινότητα, κωδική ονομασία, διεύθυνση, δομημένη καλωδίωση και τοποθεσία(location). Το κτίριο διαθέτει τηλεφωνία, προσωπικούς υπολογιστές, εργαζόμενους και φιλοξενεί servers.
- Ο server έχει κωδική ονομασία, μοντέλο, διεύθυνση IP, αν λειτουργεί με VM, και ορισμένα τεχνικά χαρακτηριστικά (RAM, CPU, CORES, OS, HARD DISC). Ο server φιλοξενεί πληροφοριακά συστήματα.
- Η τηλεφωνία έχει αριθμό, πάροχο, μοντέλο συσκευής, φραγή κινητών, τηλεφωνικό κέντρο, Internet, και δυνατότητα ή μη VOIP.
- Στο πληροφοριακό σύστημα υπάρχουν πληροφορίες σχετικά με τι είδους βάσεις δεδομένων τρέχει, το συμβόλαιο συντήρησης, τον κατασκευαστή, το λειτουργικό περιβάλλον και το όνομα-κλειδί.
- Ο προσωπικός υπολογιστής έχει μια κωδική ονομασία (κλειδί), μια διεύθυνση IP, μοντέλο, monitor, είδος, και τέλος τα τεχνικά χαρακτηριστικά του. Ο κάθε προσωπικός υπολογιστής χρησιμοποιείται από έναν εργαζόμενο και το κάθε κτίριο διαθέτει προσωπικούς υπολογιστές.
- Ο εργαζόμενος έχει όνομα, επώνυμο, φύλο, ειδικότητα, θέση, κάποια στοιχεία επικοινωνίας και έναν αριθμό μητρώου-κλειδί. Ο εργαζόμενος μπορεί να ανήκει σε πολλαπλές οργανωτικές δομές.
- Το UPS έχει μια κωδική ονομασία, ένα όνομα μοντέλου, πόσο ισχύ έχει καθώς και πόσες πρίζες διαθέτει. Το κάθε UPS χρησιμοποιείται από έναν

προσωπικό υπολογιστή προκειμένου να προστατευτεί.

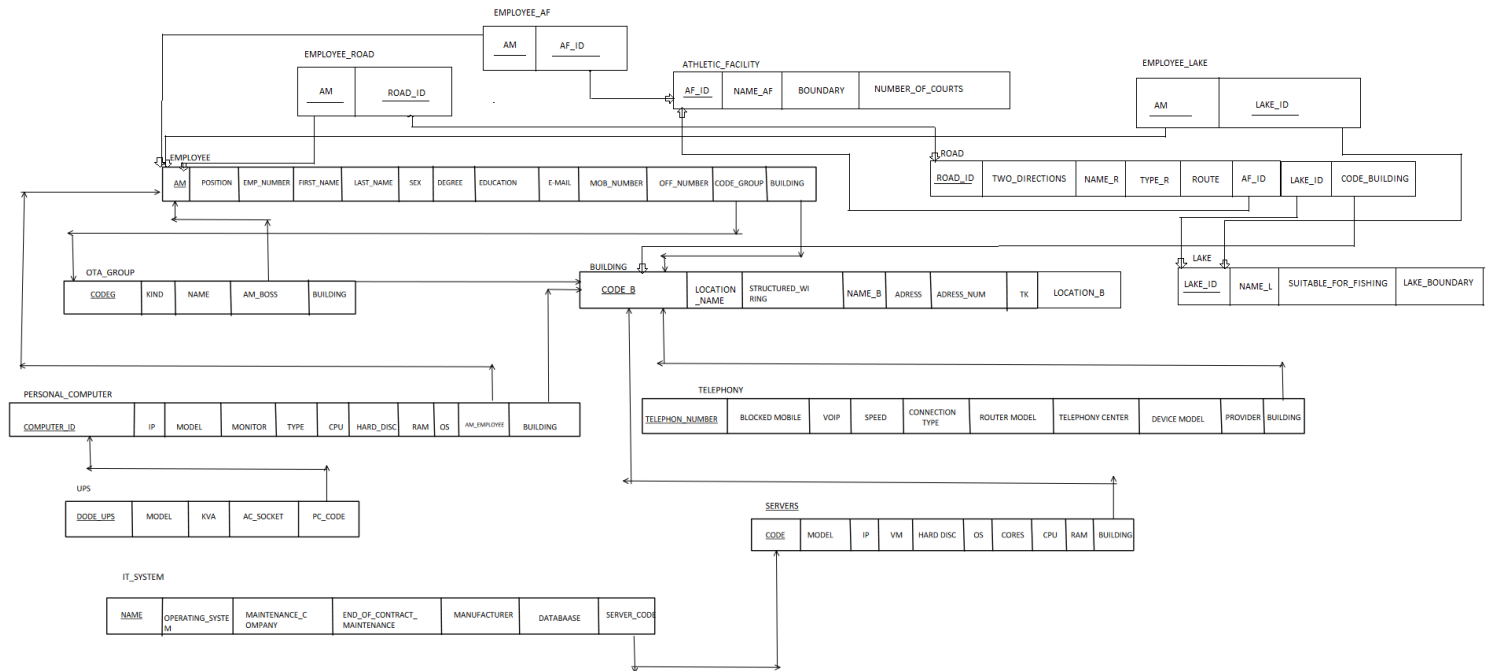
- Η λίμνη στον δήμο έχει όνομα, πληροφορίες σχετικά με την έκταση-οριοθέτηση της και αν είναι κατάλληλη για ψάρεμα. Οι εργαζόμενοι στο δήμο έχουν την δυνατότητα να ψαρεύουν στον δήμο. Ακόμα ένας δρόμος μπορεί να διασχίζει μόνο μία λίμνη (γέφυρα), ενώ μια λίμνη μπορεί να διασχίζεται από πολλούς διαφορετικούς δρόμους - γέφυρες.
- Ο δρόμος στον δήμο περιλαμβάνει όνομα, είδος όπως επαρχιακός, ταχείας κυκλοφορίας κτλ. καθώς και αν είναι διπλής κατεύθυνσης ή όχι. Επίσης, περιλαμβάνει πληροφορίες της διαδρομής του εντός του δήμου και μοναδική κωδική ονομασία. Οι εργαζόμενοι στον δήμο μπορούν να χρησιμοποιούν όλους τους δρόμους στον δήμο και οι δρόμοι φυσικά να χρησιμοποιούνται από όλους τους εργαζόμενους. Ακόμα, ο δρόμος που οδηγεί σε κάθε μία αθλητική εγκατάσταση είναι μοναδικός αλλά μια δημοτική εγκατάσταση μπορεί να είναι προσβάσιμη από πολλούς διαφορετικούς δρόμους.
- Η δημοτική αθλητική εγκατάσταση περιλαμβάνει όνομα, τον αριθμό των γηπέδων καθώς και πληροφορίες της έκτασης της. Όλοι οι εργαζόμενοι στον δήμο έχουν την δυνατότητα να τις χρησιμοποιούν για άθληση, αλλά και η κάθε αθλητική εγκατάσταση μπορεί να χρησιμοποιείται από πολλούς εργαζόμενους.

2. Διάγραμμα οντοτήτων-συσχετίσεων (ER)



*zoom in to see details

3. Μετατροπή διαγράμματος οντοτήτων-συσχετίσεων σε σχεσιακό



*zoom in to see details

4. Υλοποίηση των αντίστοιχων πινάκων και των συνδέσεων τους από το σχεσιακό μοντέλο στην πλατφόρμα SQL (PostgreSQL + PostGIS)

Αρχικά θα γίνει υλοποίηση των 7 μη χωρικών πινάκων:

Table EMPLOYEE

```
CREATE TABLE EMPLOYEE (
  AM int NOT NULL,
  POSITION_EMP varchar(15) DEFAULT NULL,
  EMP_NUMBER int DEFAULT NULL,
  FIRST_NAME varchar(50) DEFAULT NULL,
  LAST_NAME varchar(50) DEFAULT NULL,
  SEX varchar(10) DEFAULT NULL,
  DEGREE_EMP varchar(50) DEFAULT NULL,
  EDUCATION varchar(50) DEFAULT NULL,
  EMAIL varchar(50) DEFAULT NULL,
  MOB_NUMBER int DEFAULT NULL,
  OFF_NUMBER int DEFAULT NULL,
  CODE_GROUP varchar DEFAULT NULL,
  BUILDING varchar(50) DEFAULT NULL,
  PRIMARY KEY (AM));
```

Table OTA_GROUP

```
CREATE TABLE OTA_GROUP (
```

```
CODEG varchar(10) NOT NULL,  
KIND_G varchar(15) DEFAULT NULL,  
NAME_OG varchar(15) DEFAULT NULL,  
AM_BOSS int DEFAULT NULL,  
BUILDING varchar(10) DEFAULT NULL,  
PRIMARY KEY (CODEG),  
CONSTRAINT OTA_GROUP_ibfk_1 FOREIGN KEY (AM_BOSS) REFERENCES EMPLOYEE(AM));
```

```
ALTER TABLE EMPLOYEE ADD FOREIGN KEY(CODE_GROUP)REFERENCES  
OTA_GROUP(CODEG);
```

Table PERSONAL COMPUTER

```
CREATE TABLE PERSONAL_COMPUTER (  
COMPUTER_ID varchar(35) NOT NULL,  
IP varchar(15) DEFAULT NULL,  
MODEL varchar(35) DEFAULT NULL,  
MONITOR varchar(35) DEFAULT NULL,  
TYPE_PC varchar(35) DEFAULT NULL,  
CPU varchar(35) DEFAULT NULL,  
HARD_DISC varchar(35) DEFAULT NULL,  
RAM varchar(35) DEFAULT NULL,  
OS varchar(15) DEFAULT NULL,  
AM_EMPLOYEE int DEFAULT NULL,  
BUILDING varchar(15) DEFAULT NULL,  
PRIMARY KEY (COMPUTER_ID),  
CONSTRAINT PERSONAL_COMPUTER_ibfk_1 FOREIGN KEY (AM_EMPLOYEE) REFERENCES  
EMPLOYEE(AM));
```

Table TELEPHONY

```
CREATE TABLE TELEPHONY(  
TELEPHON_NUMBER BIGINT,  
BLOCKED_MOBILE CHAR,  
VOIP CHAR,  
SPEED_Mbs INTEGER,  
CONNECTION_TYPE VARCHAR(15),  
ROUTER_MODEL VARCHAR(20),  
TELEPHONY_CENTER VARCHAR(15),  
PROVIDER VARCHAR(10),  
BUILDING VARCHAR(15),  
DEVICE_MODEL VARCHAR(15),  
PRIMARY KEY(TELEPHON_NUMBER));
```

Table UPS

```
CREATE TABLE UPS(  

```

```
CODE_UPS int NOT NULL,  
MODEL varchar(35) DEFAULT NULL,  
KVA int DEFAULT NULL,  
AC_SOCKET int DEFAULT NULL,  
PC_CODE varchar(35) DEFAULT NULL,  
PRIMARY KEY (CODE_UPS),  
CONSTRAINT UPS_ibfk_1 FOREIGN KEY (PC_CODE) REFERENCES  
PERSONAL_COMPUTER(COMPUTER_ID));
```

Table SERVERS

```
CREATE TABLE SERVERS(  
CODE int,  
MODEL VARCHAR(25),  
IP VARCHAR(50),  
HARD_DISK VARCHAR(50),  
OS VARCHAR(50),  
CORES INT,  
CPU VARCHAR(50),  
RAM_GB INT,  
BUILDING VARCHAR(50),  
PRIMARY KEY(CODE));
```

Table IT_SYSTEM

```
CREATE TABLE IT_SYSTEM (  
NAME_IT varchar(30) NOT NULL,  
OPERATING_SYSTEM varchar(30) DEFAULT NULL,  
MAINTENANCE_COMPANY varchar(20) DEFAULT NULL,  
END_OF_CONTRACT_MAINTENANCE varchar(20) DEFAULT NULL,  
MANUFACTURER varchar(20) DEFAULT NULL,  
DATABASE_IT varchar(20) DEFAULT NULL,  
SERVER_CODE int DEFAULT NULL,  
PRIMARY KEY (NAME_IT),  
CONSTRAINT IT_SYSTEM_ibfk_1 FOREIGN KEY (SERVER_CODE) REFERENCES  
SERVERS(CODE));
```

Υλοποίηση των 4 χωρικών πινάκων:

Αρχικά τρέχουμε τον παρακάτω κώδικα SQL για να προσθέσουμε το postgis extension, προκειμένου να εισάγουμε spatial columns στα tables της βάσης.

```
CREATE EXTENSION postgis;
```

Table BUILDING

```
CREATE TABLE BUILDING (  
  CODE_B varchar(10) NOT NULL,  
  LOCATION_NAME varchar(15) DEFAULT NULL,  
  NAME_B varchar(15) DEFAULT NULL,  
  ADRESS varchar(15) DEFAULT NULL,  
  ADRESS_NUM INT DEFAULT NULL,  
  TK INT DEFAULT NULL,  
  LOCATION_B GEOMETRY DEFAULT NULL,  
  STRUCTUREDWIRING BOOLEAN DEFAULT NULL,  
  PRIMARY KEY (CODE_B));
```

Table LAKE

```
CREATE TABLE LAKE (  
  LAKE_ID INT NOT NULL,  
  NAME_L varchar(15) DEFAULT NULL,  
  LAKE_BOUNDARY GEOMETRY DEFAULT NULL,  
  SUITABLE_FOR_FISHING BOOLEAN DEFAULT NULL,  
  PRIMARY KEY (LAKE_ID));
```

Table ROAD

```
CREATE TABLE ROAD (  
  ROAD_ID INT NOT NULL,  
  NAME_R varchar(50) DEFAULT NULL,  
  TYPE_R varchar(50) DEFAULT NULL,  
  ROUTE GEOMETRY DEFAULT NULL,  
  AF_ID INT DEFAULT NULL,  
  LAKE_ID INT NOT NULL,  
  TWO_DIRECTIONS BOOLEAN DEFAULT NULL,  
  CODE_BUILDING varchar(50) DEFAULT NULL,  
  PRIMARY KEY (ROAD_ID),  
  FOREIGN KEY (AF_ID) REFERENCES ATHLETIC_FACILITY (AF_ID),  
  FOREIGN KEY (LAKE_ID) REFERENCES LAKE (LAKE_ID),  
  FOREIGN KEY (CODE_BUILDING) REFERENCES BUILDING (CODE_B));
```

Table ATHLETIC FACILITY

```
CREATE TABLE ATHLETIC_FACILITY (  
  AF_ID INT NOT NULL,  
  NAME_AF varchar(15) DEFAULT NULL,  
  BOUNDARY GEOMETRY DEFAULT NULL,  
  NUMBER_OF_COURTS INT DEFAULT NULL,  
  PRIMARY KEY (AF_ID));
```


Table EMPLOYEE_LAKE

```
CREATE TABLE EMPLOYEE_LAKE (  
    AM INT NOT NULL,  
    LAKE_ID INT NOT NULL);
```

Table EMPLOYEE_AF

```
CREATE TABLE EMPLOYEE_AF (  
    AM INT NOT NULL,  
    AF_ID INT NOT NULL);
```

Table EMPLOYEE_ROAD

```
CREATE TABLE EMPLOYEE_ROAD (  
    AM INT NOT NULL,  
    ROAD_ID INT NOT NULL);
```

Στην συνέχεια θα προσθέσουμε τα ξένα κλειδιά των παραπάνω πινάκων προκειμένου να φτιάξουμε τα relations, όπως περιγράφονται στο βήμα 3.

```
ALTER TABLE EMPLOYEE_LAKE ADD PRIMARY KEY (AM,LAKE_ID);  
ALTER TABLE EMPLOYEE_AF ADD PRIMARY KEY (AM,AF_ID);  
ALTER TABLE EMPLOYEE_ROAD ADD PRIMARY KEY (AM,ROAD_ID);  
ALTER TABLE EMPLOYEE ADD FOREIGN KEY(BUILDING)REFERENCES BUILDING(CODE_B);  
ALTER TABLE ROAD ADD FOREIGN KEY(CODE_BUILDING)REFERENCES BUILDING(CODE_B);  
ALTER TABLE OTA_GROUP ADD FOREIGN KEY(BUILDING)REFERENCES BUILDING(CODE_B);  
ALTER TABLE PERSONAL_COMPUTER ADD FOREIGN KEY(BUILDING)REFERENCES  
BUILDING(CODE_B);  
ALTER TABLE SERVERS ADD FOREIGN KEY(BUILDING)REFERENCES BUILDING(CODE_B);  
ALTER TABLE TELEPHONY ADD FOREIGN KEY(BUILDING)REFERENCES BUILDING(CODE_B);  
ALTER TABLE EMPLOYEE_LAKE ADD FOREIGN KEY(LAKE_ID)REFERENCES LAKE(LAKE_ID);  
ALTER TABLE ROAD ADD FOREIGN KEY(LAKE_ID)REFERENCES LAKE(LAKE_ID);  
ALTER TABLE ROAD ADD FOREIGN KEY(AF_ID)REFERENCES ATHLETIC_FACILITY(AF_ID);  
ALTER TABLE EMPLOYEE_ROAD ADD FOREIGN KEY(ROAD_ID)REFERENCES ROAD(ROAD_ID);  
ALTER TABLE EMPLOYEE_AF ADD FOREIGN KEY(AF_ID)REFERENCES  
ATHLETIC_FACILITY(AF_ID);  
ALTER TABLE EMPLOYEE_ROAD ADD FOREIGN KEY(AM)REFERENCES EMPLOYEE(AM);  
ALTER TABLE EMPLOYEE_AF ADD FOREIGN KEY(AM)REFERENCES EMPLOYEE(AM);  
ALTER TABLE EMPLOYEE_LAKE ADD FOREIGN KEY(AM)REFERENCES EMPLOYEE(AM);
```

5. Εισαγωγή ενδεικτικών δεδομένων στους πίνακες

Αρχικά θα γίνει εισαγωγή δεδομένων στους μη χωρικούς πίνακες:

Στον πίνακα EMPLOYEE θα αλλάξουμε το type των columns mob_number και off_number από int σε bigint διότι οι τηλεφωνικοί αριθμοί είναι πολλών ψηφίων.

```
ALTER TABLE public.employee  
    ALTER COLUMN mob_number TYPE bigint;
```

```
ALTER TABLE public.employee  
    ALTER COLUMN off_number TYPE bigint;
```

Λόγω των constraints που έχουν δημιουργηθεί ήδη στην βάση, δημιουργούνται προβλήματα κατά την εισαγωγή δεδομένων. Για τον λόγο αυτό θα απενεργοποιήσουμε προσωρινά όλα τα constraints των πινάκων της βάση και αφού ολοκληρωθεί η εισαγωγή των δεδομένων θα γίνει ξανά η ενεργοποίηση τους.

```
-- Disable all constraints for database  
SET session_replication_role = 'replica';
```

```
-- Enable all constraints for database  
SET session_replication_role = 'origin';
```

EMPLOYEE inserts:

```
INSERT INTO EMPLOYEE  
VALUES(1,'BOSS',1,'JOHN','PAPADOPOULOS','M','PE','EECE','Johnp@gmail.com',692715597,  
2310647899,'TPE','DIM.PAN');
```

```
INSERT INTO EMPLOYEE  
VALUES(8,'HEAD_DEPARTMENT',2,'MIKE','ANDREADIS','M','TE','ACCOUNTING','mikeand@g  
mail.com',6938399455,2310639899,'OIK','KEP.ASV');
```

```
INSERT INTO EMPLOYEE  
VALUES(18,'CLERK',3,'IOANNA','MICHOU','F','PE','ECON','MICI@gmail.com',6938367455,231  
0549899,'OIK','DIM.PAN');
```

```
INSERT INTO EMPLOYEE  
VALUES(32,'CLERK',4,'KOSTAS','CHRISTOU','M','PE','CS','kch@gmail.com',6938357455,23105  
43899,'TPE','DIM.PAN');
```

OTA_GROUP inserts:

```
INSERT INTO OTA_GROUP VALUES('TPE','DIVISION','INFORMATICS',1,'DIM.PAN');  
INSERT INTO OTA_GROUP VALUES('NOM','OFFICE','MAJOR',100,'DIM.PAN');  
INSERT INTO OTA_GROUP VALUES('OIK','DEPARTMENT','ACCOUNTING',15,'DIM.PAN');
```

PERSONAL COMPUTER inserts:

```
INSERT INTO PERSONAL_COMPUTER(COMPUTER_ID , IP , MODEL , MONITOR , TYPE_PC ,  
CPU , HARD_DISC , RAM , OS , AM_EMPLOYEE , BUILDING ) values ('JOHN PAPADOPOULOS  
PC','216.58.216.164','HP 250 G7','15.6','laptop','i7-7300HQ','512GB SSD','8GB  
DDR4','Windows 10',1,'DIM.PAN');
```

```
INSERT INTO PERSONAL_COMPUTER(COMPUTER_ID , IP , MODEL , MONITOR , TYPE_PC ,  
CPU , HARD_DISC , RAM , OS , AM_EMPLOYEE , BUILDING ) values ('MIKE ANDREADIS  
PC','216.58.216.165','HP 250 G6','15.6','laptop','i5-5200U','512GB SSD','16GB  
DDR4','Windows 10',8,'DIM.PAN');
```

```
INSERT INTO PERSONAL_COMPUTER(COMPUTER_ID , IP , MODEL , MONITOR , TYPE_PC ,  
CPU , HARD_DISC , RAM , OS , AM_EMPLOYEE , BUILDING ) values ('IOANNA MICHOU  
PC','216.58.216.166','Dell Inspiron 5501','17','laptop','i5-7200U','512GB SSD','32GB  
DDR4','Linux Mint 19',18,'DIM.PAN');
```

TELEPHONY inserts:

```
INSERT INTO TELEPHONY VALUES('2310647899','Y','Y',20,'IEEE 802.11ac','Xiaomi Mi  
4A','Xortiatl','COSMOTE','DIM.PAN','Alcatel TMAX 10');
```

```
INSERT INTO TELEPHONY VALUES('2310639899','N','Y',20,'IEEE 802.11ac','TP-LINK TD-  
W8961N','Xortiatl','COSMOTE','DIM.PAN','Alcatel TMAX 10');
```

```
INSERT INTO TELEPHONY VALUES('2310549899','Y','N',20,'IEEE 802.11ac','TP-LINK TD-  
W8961N','Xortiatl','COSMOTE','DIM.PAN','Panasonic KX-TS');
```

UPS inserts:

```
INSERT INTO UPS VALUES(50,'APC BR1300MI',1300,8,'CHRISTIANA XENOU PC');
```

```
INSERT INTO UPS VALUES(51,'APC BR1300MI',1400,6,'DESPINA SISKOU PC');
```

```
INSERT INTO UPS VALUES(52,'NOD-U1500',1500,4,'ELENA PAPA PC');
```

SERVERS inserts:

```
INSERT INTO SERVERS VALUES(47605,'Dell PowerEdge T40','192.180.3.4','1TB','KALI LINUX',8,'Intel®  
Xeon E-2224G',8,'DIM.PAN');
```

```
INSERT INTO SERVERS VALUES(47606,'Dell PowerEdge T40','192.180.0.5','1TB','KALI LINUX',8,'Intel®  
Xeon E-2224G',8,'DIM.PAN');
```

```
INSERT INTO SERVERS VALUES(47607,'Dell Precision 3630','192,180.8.1','512GB','WINDOWS  
10',8,'Intel® Xeon E-2224G',8,'LIB.PAN');
```

IT_SYSTEM inserts:

```
INSERT INTO IT_SYSTEM(NAME_IT , OPERATING_SYSTEM , MAINTENANCE_COMPANY ,  
END_OF_CONTRACT_MAINTENANCE , MANUFACTURER, DATABASE_IT, SERVER_CODE)  
values('Panorama IT','Windows 10','Allianz','4/12/2022','Intel','ABI',47605);
```

```
INSERT INTO IT_SYSTEM(NAME_IT , OPERATING_SYSTEM , MAINTENANCE_COMPANY ,  
END_OF_CONTRACT_MAINTENANCE , MANUFACTURER, DATABASE_IT, SERVER_CODE)  
values('Panorama GYM IT','Windows 10','Allianz','4/12/2022','Intel','ABI',47609);
```

```
INSERT INTO IT_SYSTEM(NAME_IT , OPERATING_SYSTEM , MAINTENANCE_COMPANY ,  
END_OF_CONTRACT_MAINTENANCE , MANUFACTURER, DATABASE_IT, SERVER_CODE)  
values('Panorama DIM IT','Windows 10','Allianz','4/12/2022','Intel','ABI',47610);
```

Εισαγωγή δεδομένων στους χωρικούς πίνακες:

BUILDING inserts:

```
INSERT INTO building (code_b, location_name, name_b, adress, adress_num, tk, location_b,  
structured_wiring) VALUES ('DIM.PAN', 'Location1', 'Building1', 'Address1', 123, 456,  
ST_GeomFromText('POLYGON((8 8, 8 9, 9 9, 9 8, 8 8))'), TRUE);
```

```
INSERT INTO building (code_b, location_name, name_b, adress, adress_num, tk, location_b,  
structured_wiring) VALUES ('KEP.ASV', 'Location2', 'Building2', 'Address2', 234, 567,  
ST_GeomFromText('POLYGON((9 9, 9 10, 10 10, 10 9, 9 9))'), FALSE);
```

LAKE inserts:

```
INSERT INTO LAKE (LAKE_ID, NAME_L, LAKE_BOUNDARY, SUITABLE_FOR_FISHING)  
VALUES  
    (1, 'Lake1', ST_GeomFromText('POLYGON((2 8, 5 8, 5 10, 2 10, 2 8))'), true),  
    (2, 'Lake2', ST_GeomFromText('POLYGON((1 5, 3 5, 5 7, 1 7, 1 5))'), false);
```

EMPLOYEE LAKE inserts:

```
INSERT INTO EMPLOYEE_LAKE (AM, LAKE_ID)  
VALUES (1, 1), (8, 1), (18, 1), (32, 1), (42, 1),  
    (57, 2), (68, 2), (73, 2), (86, 2), (95, 2),  
    (103, 3), (118, 3), (125, 3), (139, 3), (147, 3),  
    (152, 1), (167, 1), (175, 1), (188, 1), (195, 1),  
    (206, 2), (217, 2), (223, 2), (238, 2), (247, 2),  
    (254, 3), (266, 3), (277, 3), (285, 3), (297, 3),  
    (304, 1), (319, 1), (327, 1), (333, 1), (339, 1),  
    (345, 2), (352, 2), (365, 2), (372, 2), (388, 2),
```

(399, 3), (404, 3), (415, 3), (422, 3), (435, 3),
(443, 1), (459, 1), (465, 1), (473, 1), (489, 1);

ATHLETIC FACILITY inserts:

```
INSERT INTO ATHLETIC_FACILITY (AF_ID, NAME_AF, BOUNDARY, NUMBER_OF_COURTS)
VALUES (1, 'Facility1', ST_GeomFromText('POLYGON((1 1, 1 2, 2 2, 2 1, 1 1))', 4326), 3);
```

```
INSERT INTO ATHLETIC_FACILITY (AF_ID, NAME_AF, BOUNDARY, NUMBER_OF_COURTS)
VALUES (2, 'Facility2', ST_GeomFromText('POLYGON((2 2, 2 3, 3 3, 3 2, 2 2))', 4326), 2);
```

EMPLOYEE AF inserts:

```
INSERT INTO EMPLOYEE_AF (AM, AF_ID)
VALUES (1, 1);
```

```
INSERT INTO EMPLOYEE_AF (AM, AF_ID)
VALUES (8, 1);
```

ROAD inserts:

```
INSERT INTO ROAD (ROAD_ID, NAME_R, TYPE_R, ROUTE, AF_ID, LAKE_ID,
TWO_DIRECTIONS, CODE_BUILDING)
VALUES
    (1, 'Road1', 'Highway', ST_GeomFromText('LINESTRING(2 1, 3 2, 11 2)'), 1, 1, TRUE,
'DIM.PAN'),
    (2, 'Road2', 'Street', ST_GeomFromText('LINESTRING(3 4, 3 11, 11 11)'), 2, 1, FALSE,
'KEP.ASV'),
    (3, 'Road3', 'Street', ST_GeomFromText('LINESTRING(7 6, 7 1, 4 3)'), 1, 2, TRUE,
'DIM.PYL'),
    (4, 'Road4', 'Highway', ST_GeomFromText('LINESTRING(10 7, 8 7, 7 6)'), 3, 2, TRUE,
'DIM.XOR'),
```

EMPLOYEE ROAD inserts:

```
INSERT INTO EMPLOYEE_ROAD (AM, ROAD_ID) VALUES (1, 1);
INSERT INTO EMPLOYEE_ROAD (AM, ROAD_ID) VALUES (8, 2);
```

Με το ακόλουθο Python script μπορούμε να κάνουμε design τα Spatial data.

Το script βρίσκεται στο κεφάλαιο **10. Παράρτημα κώδικα**.

Αφού τελειώσαμε με τα inserts, είμαστε σε θέση να ενεργοποιήσουμε ξανά τα constraints στην βάση.

-- Enable all constraints for database
SET session_replication_role = 'origin';

6. Κατασκευάστε μια σειρά από τουλάχιστον 7 χρήσιμα μη-χωρικά ερωτήματα, εκφράστε τα σε SQL και καταγράψτε τα. Στα ερωτήματα αυτά θα πρέπει να υπάρχει τουλάχιστον ένα ερώτημα ομαδοποίησης (GROUP BY), τουλάχιστον ένα ερώτημα σύνδεσης (JOIN), τουλάχιστον ένα ερώτημα συνάθροισης (AVG, SUM, COUNT, MIN, MAX), τουλάχιστον ένα εμφωλευμένο ερώτημα, και τουλάχιστον ένα ερώτημα με δημιουργία και χρήση όψης (VIEW).

1ο Ερώτημα:

```
SELECT code_group, COUNT(*) AS Total_Employees  
FROM Employee  
GROUP BY code_group;
```

Επεξήγηση: Αυτό το ερώτημα θα ομαδοποιήσει τους υπαλλήλους ανά τμήμα (ota_group/code_group) και θα υπολογίσει τον συνολικό αριθμό των υπαλλήλων για κάθε τμήμα. Το πεδίο "code_group" είναι το πεδίο με βάση το οποίο θα γίνει η ομαδοποίηση, ενώ το "Total_Employees" είναι το όνομα του πεδίου που θα εμφανίσει τον συνολικό αριθμό των υπαλλήλων σε κάθε τμήμα.

2ο Ερώτημα:

```
SELECT Employee.first_name, Employee.last_name, Personal_Computer.Model  
FROM Employee  
JOIN Personal_Computer ON Employee.am = Personal_Computer.am_employee;
```

Επεξήγηση: Σε αυτό το ερώτημα, γίνεται η σύζευξη (JOIN) μεταξύ των πινάκων Employee και Personal_Computer, χρησιμοποιώντας το πεδίο "am" του πίνακα Employee και το πεδίο "am_employee" του πίνακα Personal_Computer. Το αποτέλεσμα περιλαμβάνει τα πεδία "first_name" και "last_name" από τον πίνακα Employee, καθώς και το πεδίο "Model" από τον πίνακα Personal_Computer. Έτσι, επιστρέφονται τα ονόματα των υπαλλήλων και τα μοντέλα των προσωπικών υπολογιστών τους.

3ο Ερώτημα:

```
SELECT DEVICE_MODEL, AVG(SPEED_Mbs) AS AVERAGE_SPEED  
FROM TELEPHONY  
GROUP BY DEVICE_MODEL;
```

Επεξήγηση: Αυτό το ερώτημα θα επιστρέψει τον μέσο όρο των ταχυτήτων (SPEED_Mbs) για κάθε μοντέλο τηλεφώνου (DEVICE_MODEL) στον πίνακα TELEPHONY.

4ο Ερώτημα:

```
SELECT *FROM ROAD WHERE LAKE_ID IN (SELECT LAKE_ID FROM LAKE WHERE  
NAME_L = 'Lake1');
```

Επεξήγηση: Αυτό το ερώτημα επιστρέφει όλους τους δρόμους που συνδέονται με τη λίμνη με όνομα 'Lake1'. Χρησιμοποιεί ένα εμφωλευμένο ερώτημα για να επιλέξει το LAKE_ID της συγκεκριμένης λίμνης και στη συνέχεια χρησιμοποιεί αυτό το αποτέλεσμα για να φιλτράρει τους δρόμους.

5ο Ερώτημα:

```
CREATE VIEW Employee_PC AS  
SELECT Employee.am, Employee.first_name, Personal_Computer.computer_id,  
Personal_Computer.model  
FROM Employee  
JOIN Personal_Computer ON Employee.am = Personal_Computer.am_employee;  
select *from Employee_PC where model = 'Dell XPS 15'
```

Επεξήγηση: Έχει δημιουργηθεί η όψη (VIEW) με όνομα "Employee_PC". Εδώ εμφανίζεται η επιλογή όλων των εγγραφών στην όψη Employee_PC όπου το μοντέλο (model) είναι "Dell XPS 15".

6ο Ερώτημα:

```
SELECT EMPLOYEE.AM, EMPLOYEE.FIRST_NAME, EMPLOYEE.SEX, OTA_GROUP.NAME_OG,  
PERSONAL_COMPUTER.MODEL  
FROM EMPLOYEE  
JOIN OTA_GROUP ON EMPLOYEE.CODE_GROUP = OTA_GROUP.CODEG  
JOIN PERSONAL_COMPUTER ON EMPLOYEE.AM = PERSONAL_COMPUTER.AM_EMPLOYEE  
where name_og = 'LOGISTICS' AND SEX = 'F';
```

Επεξήγηση: Το παραπάνω ερώτημα χρησιμοποιεί τον τελεστή WHERE για να φιλτράρει τα αποτελέσματα με βάση τα πεδία NAME_OG από τον πίνακα OTA_GROUP και SEX από τον πίνακα EMPLOYEE. Επιλέγονται οι στήλες AM, FIRST_NAME, SEX από τον πίνακα EMPLOYEE, η στήλη NAME_OG από τον πίνακα OTA_GROUP και η στήλη MODEL από τον πίνακα PERSONAL_COMPUTER. Τα αποτελέσματα περιορίζονται μόνο στις εγγραφές όπου το πεδίο NAME_OG έχει την τιμή 'LOGISTICS' και το πεδίο SEX έχει την τιμή 'F' (γυναίκες).

7ο Ερώτημα:

```
SELECT AVG(CAST(monitor AS numeric)) AS average_monitor_value  
FROM PERSONAL_COMPUTER;
```

Επεξήγηση: Αυτό το ερώτημα επιλέγει τη μέση τιμή του πεδίου "monitor" στον πίνακα "PERSONAL_COMPUTER", μετά τη μετατροπή του πεδίου σε αριθμητικό τύπο χρησιμοποιώντας τη συνάρτηση CAST().

7. Κατασκευάστε μια σειρά από τουλάχιστον 9 χρήσιμα χωρικά ερωτήματα, εκφράστε τα σε SQL και καταγράψτε τα. Στα ερωτήματα αυτά θα πρέπει να υπάρχει τουλάχιστον ένα ερώτημα υπολογισμού εμβαδών (area), τουλάχιστον ένα ερώτημα υπολογισμού ή ελέγχου αποστάσεων (distance), τουλάχιστον δύο ερωτήματα χρήσης τοπολογικού τελεστή (π.χ. touch, intersect, cross, overlap, disjoint, within, contains) μεταξύ ζεύγους χωρικών αντικειμένων (spatial join), τουλάχιστον ένα ερώτημα self spatial join με χρήση τοπολογικού τελεστή, τουλάχιστον ένα εμφωλευμένο ερώτημα, και τουλάχιστον ένα ερώτημα με δημιουργία και χρήση όψης.

1ο Ερώτημα:

```
SELECT code_b, ST_Area(location_b) AS building_area, location_b
FROM building
WHERE code_b = 'KEP.PYL';
```

Επεξήγηση: Υπολογισμός εμβαδού του κτιρίου με κωδικό KEP.PYL (ΚΕΠ Πυλαίας).

2ο Ερώτημα:

```
SELECT l.name_l, l.lake_boundary
FROM lake l
JOIN building b ON ST_DWithin(l.lake_boundary, b.location_b, 2)
WHERE b.code_b = 'KEP.XOR' AND l.suitable_for_fishing = false;
```

Επεξήγηση: Εύρεση των μη κατάλληλων λιμνών για ψάρεμα που βρίσκονται κοντά (απόσταση 2) σε ένα συγκεκριμένο κτίριο (KEP.XOR - building 6).

3ο Ερώτημα:

```
SELECT r.name_r, r.route
FROM road r
JOIN lake l ON st_touches(r.route, l.lake_boundary)
WHERE l.suitable_for_fishing = true;
```

Επεξήγηση: Αυτό το ερώτημα επιστρέφει το όνομα των δρόμων που εφάπτονται με τις λίμνες που είναι κατάλληλες για ψάρεμα.

4ο Ερώτημα:

```
SELECT b1.code_b, b1.name_b, b2.name_b, b1.location_b
```



```
FROM building b1, building b2
WHERE st_touches(b1.location_b, b2.location_b)
AND b1.code_b <> b2.code_b;
```

Επεξήγηση: Σε αυτό το ερώτημα, ο πίνακας building συνδέεται με τον εαυτό του (b1 και b2) βάσει του όρου ότι οι γεωμετρίες τους (location_b) αγγίζονται. Η υπόθεση AND εξασφαλίζει ότι αποκλείουμε τις περιπτώσεις όπου το κτίριο αγγίζει τον εαυτό του (b1.name_b <> b2.name_b).

5ο Ερώτημα:

```
SELECT e.am, e.first_name
FROM employee e
JOIN building b ON ST_Distance(b.location_b, (SELECT lake_boundary FROM lake WHERE
lake_id = 3)) >= 4
WHERE e.building = b.code_b;
```

Επεξήγηση: Σε αυτό το ερώτημα, ο πίνακας employee συνδέεται με τον πίνακα building μέσω του κωδικού κτιρίου (code_b). Ο περιορισμός ST_Distance εξασφαλίζει ότι τα κτίρια βρίσκονται σε απόσταση μεγαλύτερη ή ίση με 4 μέτρα από την συγκεκριμένη λίμνη (lake_id = 3). Τέλος, επιστρέφονται οι αριθμοί μητρώου (am) και τα ονόματα (first_name) των αντίστοιχων υπαλλήλων.

6ο Ερώτημα:

```
SELECT r.name_r, r.route
FROM road r
WHERE type_r = 'Street' and EXISTS (
    SELECT 1
    FROM lake l
    WHERE st_intersects(r.ROUTE, l.lake_boundary)
);
```

Επεξήγηση: Σε αυτό το ερώτημα, επιλέγουμε το όνομα των δρόμων (r.NAME_R) που διαπερνούν μια λίμνη. Το εμφωλευμένο ερώτημα ελέγχει για κάθε δρόμο αν υπάρχει τουλάχιστον μία λίμνη όπου ο δρόμος διασταυρώνεται με τη γεωμετρία της. Το πεδίο r.ROUTE αναπαριστά το γεωμετρικό στοιχείο του δρόμου.

7ο Ερώτημα:

```
CREATE VIEW employees_view AS
SELECT e.am, e.first_name, b.code_b, a.name_af
FROM employee e
JOIN building b ON e.building = b.code_b
```

```
JOIN employee_af ea ON e.am = ea.am  
JOIN athletic_facility a ON ea.af_id = a.af_id;
```

```
select *from employees_view where code_b = 'DIM.PAN'
```

Επεξήγηση: Δημιουργούμε μια αναλυτική προβολή που ονομάζεται employees_view που συνδυάζει πληροφορίες από τους πίνακες employee, building, employee_af και athletic_facility. Η προβολή περιλαμβάνει στήλες για τον κωδικό υπαλλήλου, το όνομα υπαλλήλου, το όνομα του κτιρίου και το όνομα της αθλητικής εγκατάστασης.

Το ερώτημα θα επιστρέψει όλες τις γραμμές από την προβολή employees_view όπου η στήλη code_b είναι 'DIM.PAN'.

8ο Ερώτημα:

```
SELECT l.name_l, l.lake_boundary  
FROM lake l  
JOIN building b ON ST_DWithin(l.lake_boundary, b.location_b, 8)  
WHERE b.code_b = 'DIM.PAN' AND l.suitable_for_fishing = true;
```

Επεξήγηση: Εύρεση των κατάλληλων λιμνών για ψάρεμα που βρίσκονται κοντά (απόσταση 8) σε ένα συγκεκριμένο κτίριο (DIM.PAN - building 1).

9ο Ερώτημα:

```
SELECT e.am, e.first_name, e.last_name, SUM(ST_Area(l.lake_boundary)) AS lake_area  
FROM employee e  
JOIN employee_lake el ON e.am = el.am  
JOIN lake l ON el.lake_id = l.lake_id  
WHERE l.suitable_for_fishing = true  
GROUP BY e.am;
```

Επεξήγηση: Ερώτημα που θα επιστρέψει τα ονόματα των υπαλλήλων και το συνολικό εμβαδόν των λιμνών που είναι κατάλληλες για ψάρεμα και ψαρεύουν οι εργαζόμενοι, βασισμένο στον πίνακα EMPLOYEE_LAKE

8. Εκτελέστε τα ερωτήματα αυτά στο περιβάλλον διαχείρισης της βάσης σας (π.χ. στο σύστημα PostgreSQL + PostGIS), και καταγράψτε τα αποτελέσματα τους.

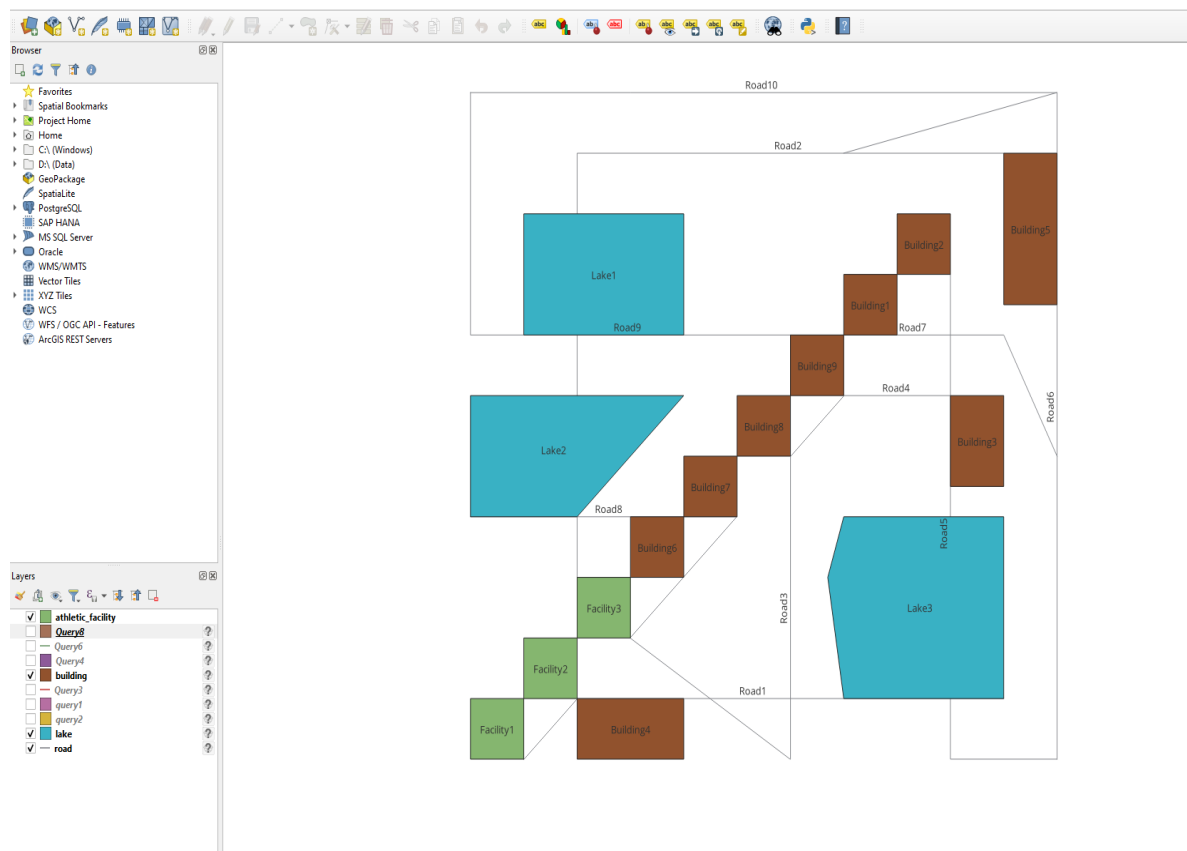
9. Για τα μη χωρικά ερωτήματα εμφανίστε τα αποτελέσματα σε μορφή πινάκων ή screenshots, ενώ για τα χωρικά ερωτήματα οπτικοποιήστε την εκτέλεση τους χρησιμοποιώντας ένα κατάλληλο tool (π.χ. uDig, MapServer, κλπ.).

Η αναπαράσταση των χωρικών δεδομένων έγινε με την βοήθεια του εργαλείου QGIS. Παρακάτω αναγράφονται τα βήματα για την ορθή λειτουργία του εργαλείου που ακολούθησα:

- Ανοίξετε το QGIS και συνδεθείτε στην γεωγραφική σας βάση δεδομένων.
- Επιλέξτε την επιθυμητή βάση δεδομένων από τη λίστα συνδέσεων.
- Προσθέστε τον πίνακα ή την προβολή που περιέχει τα γεωγραφικά δεδομένα που θέλετε να κάνετε visualization και SQL queries.
- Στην γραμμή εργαλείων επιλέγουμε Databases και στην συνέχεια DB Manager.
- Στο παράθυρο που εμφανίζεται επιλέγουμε PostGIS και στην συνέχεια κάνουμε κλικ στην βάση μας.
- Κάντε κλικ στο κουμπί "SQL WINDOW" για να κάνετε SQL queries.
- Αφού συνταχθεί το ερώτημα κάνουμε κλικ στο execute και στην συνέχεια το αποτέλεσμα εφόσον είναι χωρικό μπορούμε να το φορτώσουμε ως ένα νέο layer.

Με αυτά τα βήματα θα μπορείτε να κάνετε SQL ερωτήματα γεωγραφικών δεδομένων στο QGIS και να επεξεργαστείτε τα αποτελέσματα.

Παρακάτω μπορούμε να δούμε την αναπαράσταση των χωρικών μας δεδομένων από τα Spatial tables της βάσης δεδομένων.



Εμφάνιση αποτελεσμάτων μη χωρικών ερωτημάτων:









1ο Ερώτημα:

Data Output		Messages	Notifications
	code_group character varying		total_employees bigint
1	ENV		5
2	SUP		7
3	ATH		1
4	POL		6
5	DIK		7
6	OIK		2
7	LOG		7
8	TPE		8
9	TEK		7

2ο Ερώτημα:

Data Output				Messages		Notifications	
    				 			
	first_name character varying (50)		last_name character varying (50)		model character varying (35)		
1	MARIA		IOANNOU		Lenovo ThinkPad X1 Carbon		
2	NIKOS		GEORGIOU		HP EliteBook 840 G6		
3	ELENA		PAPADOPOULOU		Dell XPS 13		
4	DIMITRIOS		PANAGIOTOU		Lenovo ThinkCentre M920		
5	ELENI		GEORGIOU		Dell OptiPlex 5070		
6	NIKOLAOS		KONSTANTINOU		HP Pavilion Gaming Desktop		
7	MARINA		PAPANIKOLAOU		Acer Aspire TC-895		
8	STAVROULA		GEORGIOU		Asus ROG Strix G15		
9	JOHN		PAPADOPOULOS		MSI GS66 Stealth		
10	SOPHIA		PANAGIOTOU		HP ProDesk 600 G6		
11	ALEXANDROS		KONSTANTINOU		Dell OptiPlex 3080		
12	GEORGIA		PAPANIKOLAOU		Lenovo Legion 5		
13	CHRISTINA		LOUKA		Acer Nitro 5		
14	PETROS		KARAGIANNIS		HP Pavilion 590		
15	ATHINA		PAPANASTASIOU		Dell Inspiron 3880		
16	ELENI		ANDREOU		Asus VivoBook 15		
17	GEORGIOS		PAPADOPOULOS		Lenovo IdeaCentre 3		
18	ELENI		ANTONIOU		Dell G5 Gaming Desktop		
19	DIMITRIOS		PANAGIOTOU		HP Pavilion TP01		
20	ELENI		GEORGIOU		Acer Predator Helios 300		
21	GEORGIOS		KARAGIANNIS		Lenovo ThinkPad X1 Carbon		
22	GIORGOS		PAPADOPOULOS		Dell XPS 15		
23	DIMITRA		GEORGIADOU		HP EliteDesk 800 G6		
24	NIKOLAOS		KONSTANTINOU		Asus ROG Zephyrus G14		
25	MARINA		PAPANIKOLAOU		Dell Alienware Aurora R10		
26	PANAGIOTIS		CHRISTODOULOU		Lenovo Legion T5		
27	STAVROULA		ANTONIOU		HP ENVY x360		
28	DIMITRIOS		PANAGIOTOU		Acer Aspire TC-895		
29	CHRISTINA		LOUKA		Dell XPS 8940		
Total rows: 47 of 47		Query complete 00:00:00.049					









3ο Ερώτημα:

	Data Output	Messages	Notifications
	       		
	device_model character varying (15)		average_speed numeric
1	Alcatel TMAX 10		20.0000000000000000

4ο Ερώτημα:

[illegible]

5ο Ερώτημα:

Data Output					Messages	Notifications	
							
	am integer	first_name character varying (50)	computer_id character varying (35)	model character varying (35)			
1	57	NIKOS	PC504	Dell XPS 15			
2	399	GIORGOS	PC406	Dell XPS 15			

6ο Ερώτημα:

	am integer	first_name character varying (50)	sex character varying (10)	name Og character varying (15)	model character varying (35)
1	42	MARIA	F	LOGISTICS	Lenovo ThinkPad X1 Carbon
2	327	ELENI	F	LOGISTICS	Dell OptiPlex 5070
3	254	ELENI	F	LOGISTICS	Asus VivoBook 15
4	327	ELENI	F	LOGISTICS	Acer Predator Helios 300
5	42	MARIA	F	LOGISTICS	Apple MacBook Air

7ο Ερώτημα:

	average_monitor_value numeric
1	19.55636363636364

Εμφάνιση αποτελεσμάτων χωρικών ερωτημάτων:

1ο Ερώτημα:

Εκτέλεση ερωτήματος:

The screenshot shows the QGIS DB Manager interface. On the left, the 'Providers' tree is expanded to 'PostGIS' > 'connectiondb' > 'public'. The main panel displays a SQL query:

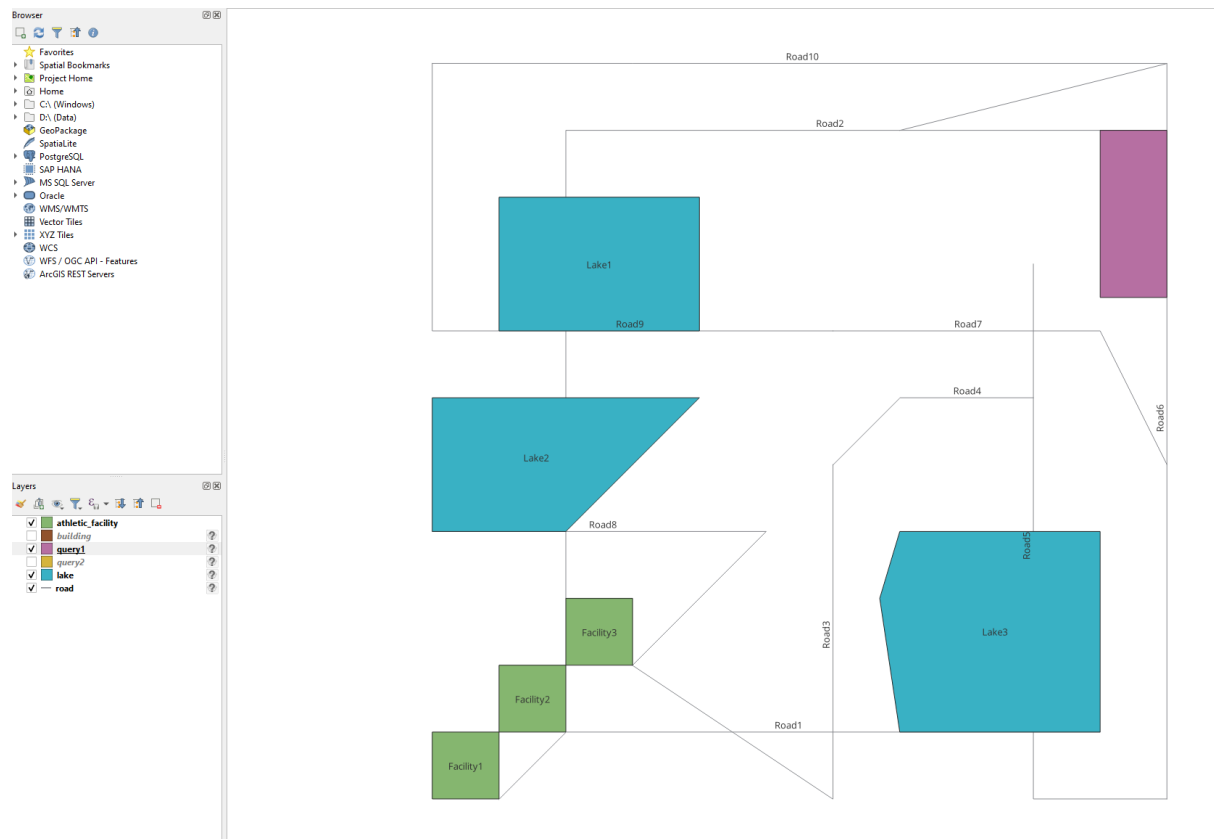
```
1 SELECT code_b, ST_Area(location_b) AS building_area, location_b
2 FROM building
3 WHERE code_b = 'KEP.PYL'
4
5
```

The query has been executed, resulting in 1 row and 0.001 seconds. The result is displayed in a table:

code_b	building_area	location_b
1 KEP.PYL	2.5	01030000000010...

Below the table, the 'Load as new layer' section is checked. The 'Column(s) with unique values' is set to 'code_b' and the 'Geometry column' is set to 'location_b'. The 'Layer name (prefix)' is 'query1'. The 'Load' button is visible.

Γραφική απεικόνιση του αποτελέσματος:



2ο Ερώτημα:

Εκτέλεση ερωτήματος:

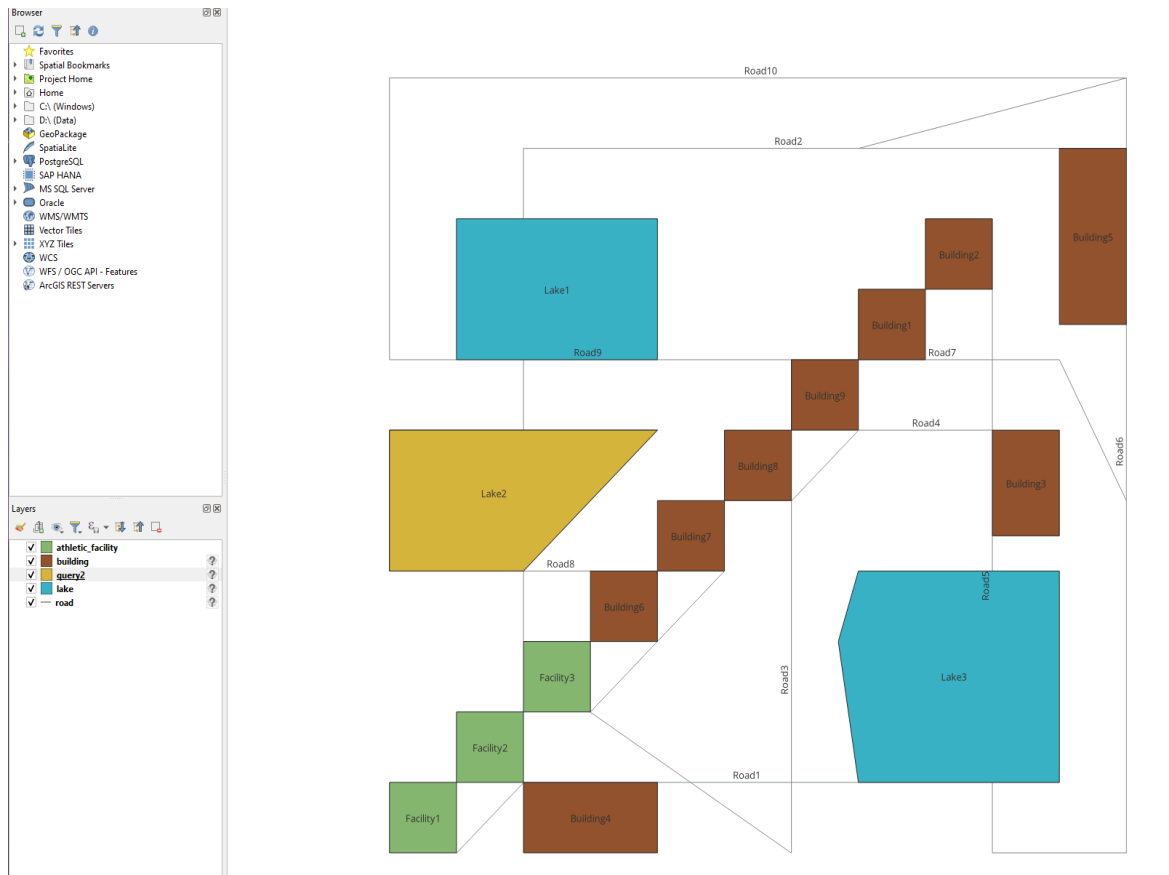
The screenshot shows the DB Manager interface with a SQL query execution window. The query is:

```
1 SELECT 1.name_1, 1.lake_boundary
2 FROM lake 1
3 JOIN building b ON ST_DWithin(1.lake_boundary, b.location_b, 2)
4 WHERE b.code_b = 'KEP.XOR' AND 1.suitable_for_fishing = false;
5
```

The result table shows one row:

name_1	lake_boundary
1 Lake2	0103000000010...

Γραφική απεικόνιση του αποτελέσματος:



3ο Ερώτημα:

Εκτέλεση ερωτήματος:

The screenshot shows the DB Manager interface with a SQL query execution window. The query is:

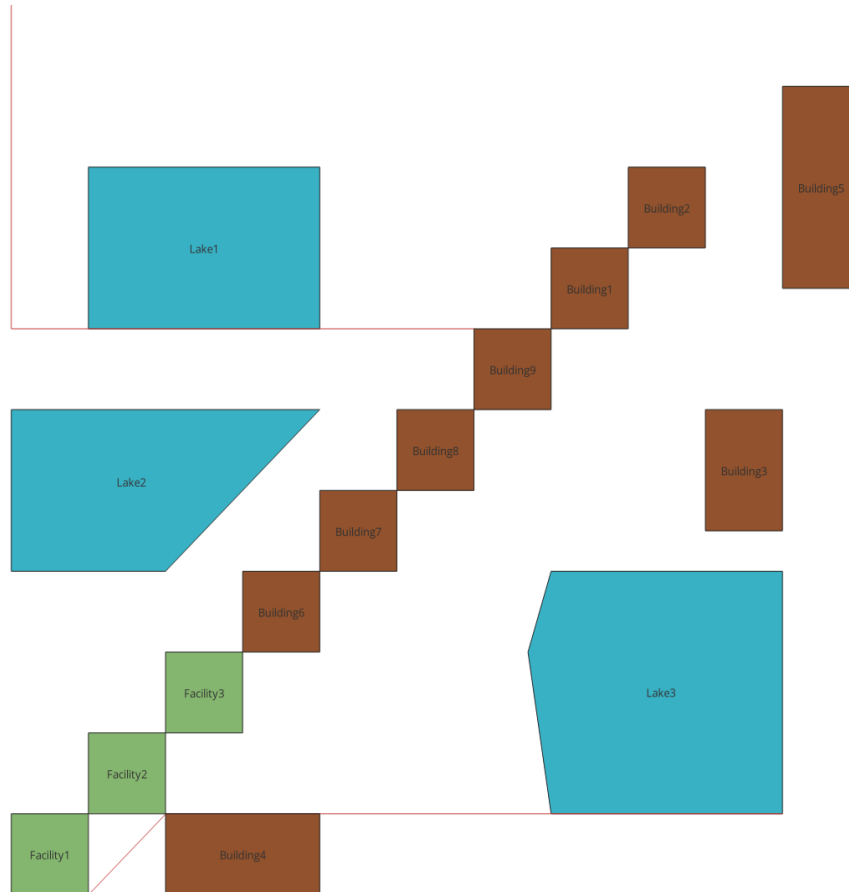
```
1 SELECT r.name_r, r.route
2 FROM road r
3 JOIN lake l ON st_touches(r.route, l.lake_boundary)
4 WHERE l.suitable_for_fishing = true;
5
6
7
```

The results show 2 rows:

	name_r	route
1	Road9	01020000000030...
2	Road1	01020000000030...

The interface also shows a 'Load as new layer' dialog with options to load the query results as a new layer.

Γραφική απεικόνιση του αποτελέσματος:



4ο Ερώτημα:

Εκτέλεση ερωτήματος:

DB Manager

Database Schema Table

Import Layer/File... Export to File...

Providers

- GeoPackage
- Oracle Spatial
- PostGIS
 - connectiondb
 - public
 - athletic_facility
 - building
 - employee
 - employee_af
 - employee_lake
 - employee_pc
 - employee_road
 - geography_columns
 - geometry_columns
 - it_system
 - lake
 - ota_group
 - personal_computer
 - road
 - servers
 - spatial_ref_sys
 - telephony
 - ups
- Spatialite
- Virtual Layers

Info Table Preview Query (connectiondb) X

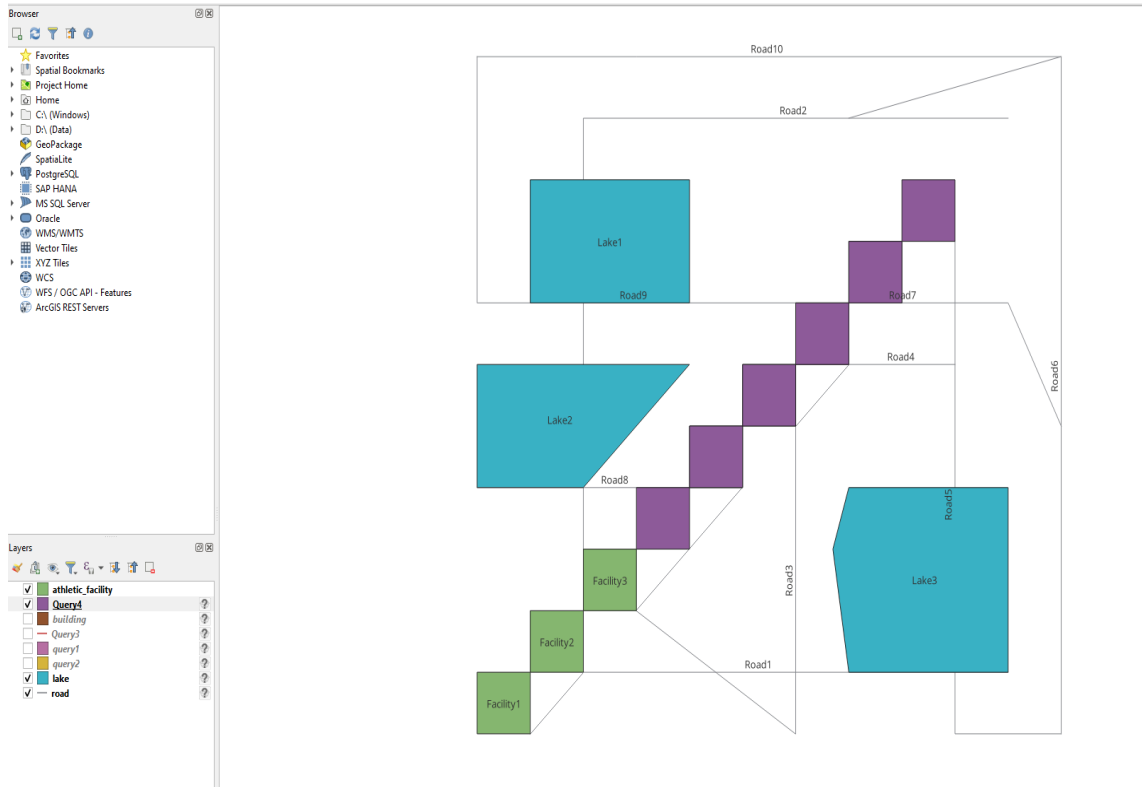
Saved query Name Save Delete Load File Save As File

```
1 SELECT b1.code_b, b1.name_b, b2.name_b, b1.location_b
2 FROM building b1, building b2
3 WHERE st_touches(b1.location_b, b2.location_b)
4 AND b1.code_b <> b2.code_b;
5
6
7
```

Execute 10 rows, 0.004 seconds Create a view Clear Query History

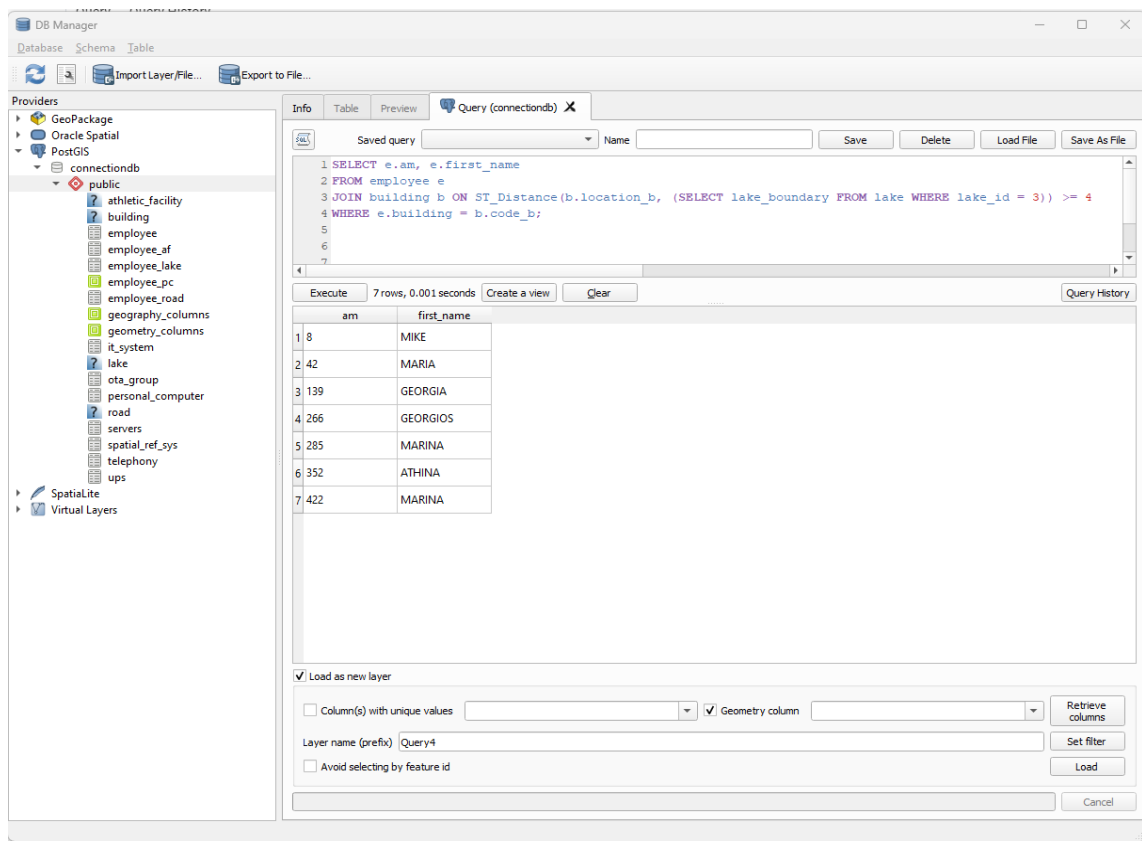
	code_b	name_b	name_b	location_b
1	DIM.PAN	Building1	Building2	0103000000010...
2	DIM.PAN	Building1	Building9	0103000000010...
3	KEP.ASV	Building2	Building1	0103000000010...
4	KEP.XOR	Building6	Building7	0103000000010...
5	AMAKS.PYL	Building7	Building6	0103000000010...
6	AMAKS.PYL	Building7	Building8	0103000000010...
7	AMAKS.PAN	Building8	Building7	0103000000010...
8	AMAKS.PAN	Building8	Building9	0103000000010...
9	AMAKS.XOR	Building9	Building1	0103000000010...
10	AMAKS.XOR	Building9	Building8	0103000000010...

Γραφική απεικόνιση του αποτελέσματος:



5ο Ερώτημα:

Εκτέλεση ερωτήματος:



6ο Ερώτημα: Εκτέλεση ερωτήματος:

DB Manager

Database Schema Table

Providers

- GeoPackage
- Oracle Spatial
- PostGIS
 - connectiondb
 - public
 - athletic_facility
 - building
 - employee
 - employee_af
 - employee_lake
 - employee_pc
 - employee_road
 - geography_columns
 - geometry_columns
 - it_system
 - lake
 - ota_group
 - personal_computer
 - road
 - servers
 - spatial_ref_sys
 - telephony
 - ups
- SpatialLite
- Virtual Layers

Info Table Preview Query (connectiondb) X

Saved query Name Save Delete Load File Save As File

```
1 SELECT r.name_r, r.route
2 FROM road r
3 WHERE type_r = 'Street' and EXISTS (
4   SELECT 1
5   FROM lake l
6   WHERE st_intersects(r.ROUTE, l.lake_boundary)
7 );
```

Execute 4 rows, 0.002 seconds Create a view Clear Query History

	name_r	route
1	Road2	01020000000030...
2	Road5	01020000000030...
3	Road8	01020000000030...
4	Road9	01020000000030...

☒ Load as new layer

Γραφική απεικόνιση του αποτελέσματος:

Browser

- Favorites
- Spatial Bookmarks
- Project Home
- Home
- C:\ (Windows)
- D:\ (Data)
- GeoPackage
- SpatialLite
- PostgreSQL
- SAP HANA
- MS SQL Server
- Oracle
- WMS/WMTS
- Vector Tiles
- XYZ Tiles
- WCS
- WFS / OGC API - Features
- ArcGIS REST Servers

Layers

- ☒ athletic_facility
- ☒ Query4
- ☒ building
- ☐ Query3
- ☐ query1
- ☐ query2
- ☒ lake
- ☐ road

Log Messages

Plugins X General X PostGIS X Messages X Python error X

7ο Ερώτημα:

Εκτέλεση ερωτήματος:

The screenshot shows the DB Manager interface with a query executed against the 'connectiondb' database. The query is: `select * from employees_view where code_b = 'DIM.PAN'`. The result set contains 3 rows.

am	first_name	code_b	name_af
1	JOHN	DIM.PAN	Facility1
18	IOANNA	DIM.PAN	Facility1
32	KOSTAS	DIM.PAN	Facility1

8ο Ερώτημα:

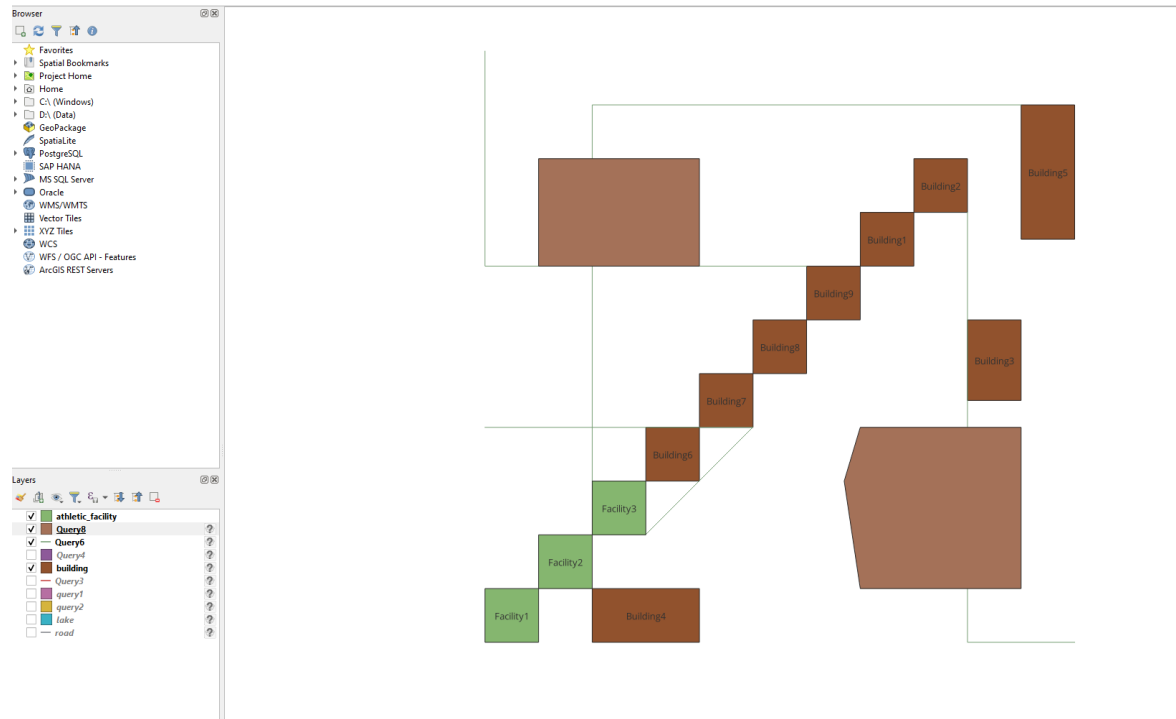
Εκτέλεση ερωτήματος:

The screenshot shows the DB Manager interface with a query executed against the 'connectiondb' database. The query is: `SELECT l.name_l, l.lake_boundary FROM lake l JOIN building b ON ST_DWithin(l.lake_boundary, b.location_b, 8) WHERE b.code_b = 'DIM.PAN' AND l.suitable_for_fishing = true;`. The result set contains 2 rows.

name_l	lake_boundary
Lake1	0103000000010...
Lake3	0103000000010...

☒ Load as new layer

Γραφική απεικόνιση του αποτελέσματος:



9ο Ερώτημα:

Εκτέλεση ερωτήματος:

The screenshot shows a SQL query editor interface. On the left, there is a 'Browsers' panel with a tree view of data sources including 'GeoPackage', 'Oracle Spatial', 'PostGIS', and 'connectiondb'. The 'connectiondb' panel is expanded, showing a list of tables: 'athletic_facility', 'building', 'employee', 'employee_af', 'employee_lake', 'employee_pc', 'employee_road', 'geography_columns', 'geometry_columns', 'it_system', and 'lake'. The main panel shows the SQL query editor with a query to select employee information and lake area.

```
1 SELECT e.am, e.first_name, e.last_name, SUM(ST_Area(l.lake_boundary)) AS lake_area
2 FROM employee e
3 JOIN employee_lake el ON e.am = el.am
4 JOIN lake l ON el.lake_id = l.lake_id
5 WHERE l.suitable_for_fishing = true
6 GROUP BY e.am;
```

The screenshot shows the DB Manager interface. On the left, the 'Providers' tree is expanded to 'PostGIS' > 'connectiondb' > 'public'. The main pane shows a 'Query (connectiondb)' result with 35 rows. The table has columns: 'am', 'first_name', 'last_name', and 'lake_area'. The data represents a list of employees and their associated lake areas.

	am	first_name	last_name	lake_area
1	297	NIKOLAOS	KONSTANTINIDIS	9.45
2	304	ELENI	ANTONIOU	6.0
3	333	GEORGIOS	KARAGIANNIS	6.0
4	443	STAVROULA	ANTONIOU	6.0
5	473	NIKOLAOS	PAPADAKIS	6.0
6	175	GIORGOS	KONSTANTINIDIS	6.0
7	285	MARINA	PAPANIKOLAOU	9.45
8	327	ELENI	GEORGIOU	6.0
9	415	NIKOLAOS	KONSTANTINOU	9.45
10	167	KATERINA	GEORGIOU	6.0
11	399	GIORGOS	PAPADOPOULOS	9.45
12	277	ATHANASIOS	GEORGIAKIS	9.45
13	118	PANAGIOTIS	ANDREOU	9.45
14	195	PETROS	KARAGIANNIS	6.0
15	459	DIMITRIOS	PANAGIOTOU	6.0
16	339	MARIA	ANDREOU	6.0
17	103	ELENI	KARALI	9.45
18	42	MARIA	IOANNOU	6.0
19	188	MARINA	PAPADOPOULOU	6.0
20	152	DIMITRIS	ANTONIOU	6.0
21	125	MARIANNA	STAVROU	9.45
22	147	CHRISTINA	LOUKA	9.45
23	32	KOSTAS	CHRISTOU	6.0
24	139	GEORGIA	PAPANIKOLAOU	9.45
25	254	ELENI	ANDREOU	9.45
26	319	DIMITRIOS	PANAGIOTOU	6.0
27	1	JOHN	PAPADOPOULOS	6.0
28	404	DIMITRA	GEORGIAKIS	9.45
29	422	MARINA	PAPANIKOLAOU	9.45
30	465	CHRISTINA	LOUKA	6.0
31	266	GEORGIOS	PAPADOPOULOS	9.45
32	18	IOANNA	MICHOU	6.0
33	489	STAVROULA	GEORGIOU	6.0
34	435	PANAGIOTIS	CHRISTODOULOU	9.45
35	8	MIKE	ANDREADIS	6.0

At the bottom of the table, there is a checkbox labeled 'Load as new layer' which is checked.

10. Παράρτημα κώδικα

Παρακάτω παρουσιάζεται ο κώδικας σε python που χρησιμοποιήθηκε για την σχεδίαση των χωρικών οντοτήτων:

```
import geopandas as gpd
from matplotlib import pyplot as plt
from matplotlib.patches import Patch
from matplotlib.lines import Line2D
from shapely.geometry import Polygon, LineString

# Assuming you have a dataframe with the building coordinates
# and structure_dwiring column
building_data = {
    'LOCATION_B': [
        Polygon([(10, 7), (11, 7), (11, 5.5), (10, 5.5)]),
        Polygon([(3, 2), (5, 2), (5, 1), (3, 1)]),
        Polygon([(11, 11), (12, 11), (12, 8.5), (11, 8.5)]),
        Polygon([(4, 4), (4, 5), (5, 5), (5, 4)]),
    ]
}
```

```

        Polygon([(5, 5), (5, 6), (6, 6), (6, 5)]),
        Polygon([(6, 6), (6, 7), (7, 7), (7, 6)]),
        Polygon([(7, 7), (7, 8), (8, 8), (8, 7)]),
        Polygon([(8, 8), (8, 9), (9, 9), (9, 8)]),
        Polygon([(9, 9), (9, 10), (10, 10), (10, 9)])
    ]
}

lake_data = {
    'LAKE_ID': [1, 2, 3],
    'NAME_L': ['Lake1', 'Lake2', 'Lake3'],
    'SUITABLE_FOR_FISHING': [True, False, True],
    'LAKE_BOUNDARY': [
        Polygon([(2, 8), (5, 8), (5, 10), (2, 10), (2, 8)]),
        Polygon([(1, 5), (3, 5), (5, 7), (1, 7)]),
        Polygon([(8, 2), (11, 2), (11, 5), (8, 5), (7.7, 4), (8, 2)])
    ]
}

road_data = {
    'ROAD_ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'NAME_R': ['Road1', 'Road2', 'Road3', 'Road4', 'Road5', 'Road6', 'Road7',
    'Road8', 'Road9', 'Road10'],
    'TYPE_R': ['Highway', 'Street', 'Street', 'Highway', 'Street', 'Street',
    'Highway', 'Street', 'Street', 'Highway'],
    'ROUTE': [
        LineString([(2, 1), (3, 2), (11, 2)]),
        LineString([(3, 4), (3, 11), (11, 11)]),
        LineString([(7, 6), (7, 1), (4, 3)]),
        LineString([(10, 7), (8, 7), (7, 6)]),
        LineString([(10, 9), (10, 1), (12, 1)]),
        LineString([(12, 1), (12, 12), (8, 11)]),
        LineString([(7, 8), (11, 8), (12, 6)]),
        LineString([(4, 3), (6, 5), (1, 5)]),
        LineString([(1, 12), (1, 8), (7, 8)]),
        LineString([(12, 12), (4, 12), (1, 12)])
    ],
    'AF_ID': [1, 2, 1, 3, 2, 1, 3, 2, 3, 1],
    'LAKE_ID': [1, 1, 2, 2, 3, 3, 1, 1, 2, 2],
    'TWO_DIRECTIONS': [True, False, True, True, False, True, False, False, True,
    True],
    'CODE_BUILDING': ['B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'B10']
}

af_data = {
    'AF_ID': [1, 2, 3],
    'NAME_AF': ['Facility1', 'Facility2', 'Facility3'],
    'BOUNDARY': [
        Polygon([(1, 1), (1, 2), (2, 2), (2, 1)]),
        Polygon([(2, 2), (2, 3), (3, 3), (3, 2)]),
        Polygon([(3, 3), (3, 4), (4, 4), (4, 3)])
    ],
    'NUMBER_OF_COURTS': [2, 1, 3]
}

employee_road_data = {
    'AM': [1, 8, 18, 32, 42, 57, 68, 73, 86, 95, 103, 118, 125, 139, 147, 152, 167,
    175, 188, 195, 206, 217, 223, 238,
    247, 254, 266, 277, 285, 297, 304, 319, 327, 333, 339, 345, 352, 365,
    372, 388, 399, 404, 415, 422, 435,
    443, 459, 465, 473, 489],
    'ROAD_ID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2,
    3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4,

```

```

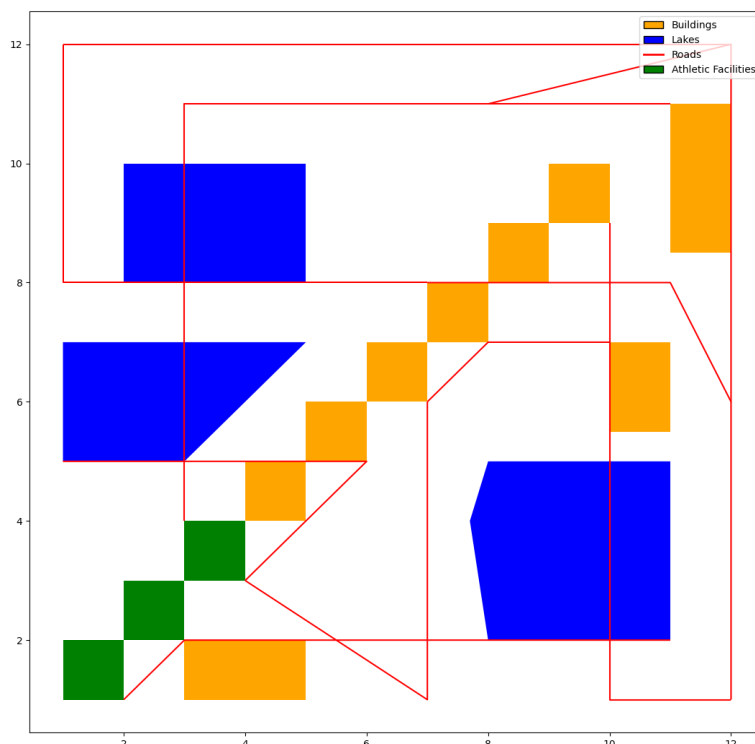
5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
}

employee_af_data = {
    'AM': [1, 8, 18, 32, 42, 57, 68, 73, 86, 95, 103, 118, 125, 139, 147, 152, 167,
175, 188, 195, 206, 217, 223, 238,
247, 254, 266, 277, 285, 297, 304, 319, 327, 333, 339, 345, 352, 365,
372, 388, 399, 404, 415, 422, 435,
443, 459, 465, 473, 489],
    'AF_ID': [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2,
3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1,
2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
}

# Create GeoDataFrames
building_gdf = gpd.GeoDataFrame(building_data, geometry='LOCATION_B')
lake_gdf = gpd.GeoDataFrame(lake_data, geometry='LAKE_BOUNDARY')
road_gdf = gpd.GeoDataFrame(road_data, geometry='ROUTE')
af_gdf = gpd.GeoDataFrame(af_data, geometry='BOUNDARY')
# Set up the plot
fig, ax = plt.subplots()
# Plotting the polygons
building_gdf.plot(ax=ax, color='orange', label='Buildings')
lake_gdf.plot(ax=ax, color='blue', label='Lakes')
road_gdf.plot(ax=ax, color='red', label='Roads')
af_gdf.plot(ax=ax, color='green', label='Athletic Facilities')
# Create custom Legend handles
building_legend = Patch(facecolor='orange', edgecolor='black', label='Buildings')
lake_legend = Patch(facecolor='blue', edgecolor='black', label='Lakes')
road_legend = Line2D([], [], color='red', linewidth=2, label='Roads')
af_legend = Patch(facecolor='green', edgecolor='black', label='Athletic Facilities')
# Add the Legend handles to the plot
ax.legend(handles=[building_legend, lake_legend, road_legend, af_legend])
# Show the plot
plt.show()

```

Παρακάτω βλέπουμε τα αποτελέσματα του visualization:



11. Βιβλιογραφική αναφορά.

- Διαφάνειες μαθήματος:
<https://elearning.auth.gr/course/view.php?id=11632>
- Πληροφορίες για τις Spatial DBs:
<https://learn.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-ver16>
- QGIS:
<https://qgis.org/en/site/>
https://docs.qgis.org/3.28/en/docs/user_manual/