



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ II

4ο Τέταρτο παραδοτέο – Υλοποίηση Βάσης Γραφοδεδομένων

Διονύσιος Θεοδόσης	A.M:321/2015066
Σταύρος Σταυρούδης	A.M:321/2015189
Ονούφριος Νταβρής	A.M:321/2015150

28/1/2024



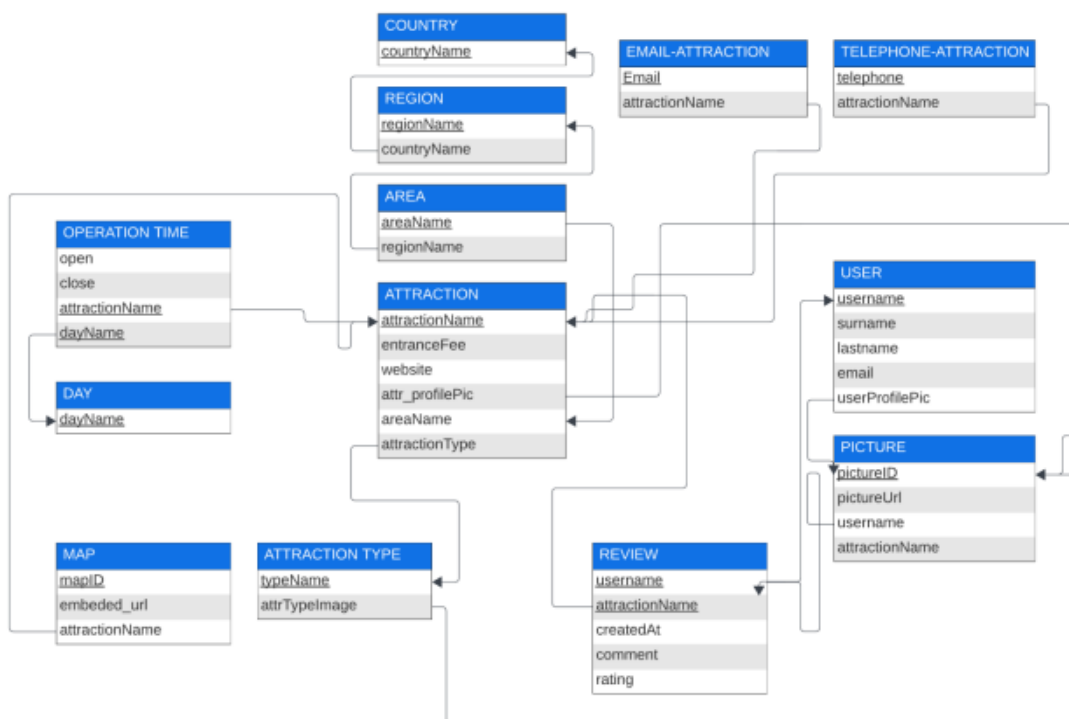
Περιεχόμενα

ΒΗΜΑΤΑ ΜΕΤΑΤΡΟΠΗΣ ΤΗΣ ΣΧΕΣΙΑΚΗΣ ΒΑΣΗΣ ΣΕ ΒΑΣΗ ΓΡΑΦΟΔΕΔΟΜΕΝΩΝ.....	3
Γενικά:.....	3
ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΣΤΟ ΝΕΟ4J ΚΑΙ ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΜΕΣΩ CSV.....	4
Στιγμιότυπα οθόνης για την δημιουργία των κόμβων και εισαγωγή μοναδικού χαρακτηριστικού γνωρίσματος:.....	5
Στιγμιότυπα οθόνης για την δημιουργία συσχετίσεων μεταξύ των κόμβων	6
Διαγραφή των ιδιοτήτων (ξένα κλειδιά) από τους κόμβους.....	7
Ο ΤΕΛΙΚΟΣ ΣΥΓΚΕΝΤΡΩΤΙΚΟΣ ΓΡΑΦΟΣ.....	8
ΠΕΝΤΕ ΕΡΩΤΗΜΑΤΑ ΣΤΟ ΝΕΟ4J ΒΑΣΕΙ ΤΟΥ 2 ^{ΟΥ} ΠΑΡΑΔΟΤΕΟΥ.....	9
Ερώτημα 1 ^ο :	9
Ερώτημα 2 ^ο :	10
Ερώτημα 3 ^ο :	10
Ερώτημα 4 ^ο :	11
Ερώτημα 5 ^ο :	12

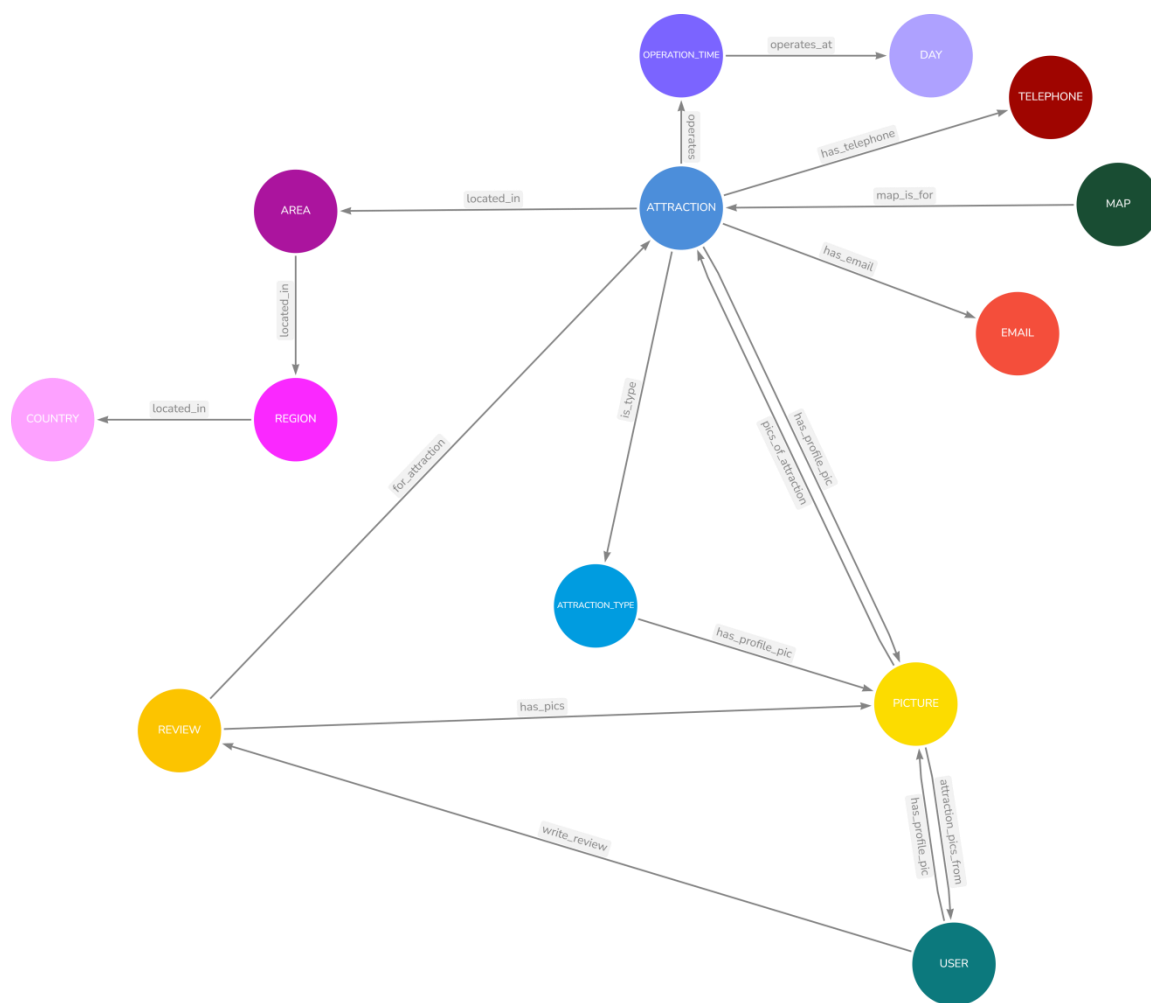
ΒΗΜΑΤΑ ΜΕΤΑΤΡΟΠΗΣ ΤΗΣ ΣΧΕΣΙΑΚΗΣ ΒΑΣΗΣ ΣΕ ΒΑΣΗ ΓΡΑΦΟΔΕΔΟΜΕΝΩΝ

Γενικά:

Για την μετατροπή οποιασδήποτε σχεσιακής βάσης δεδομένων σε βάση γραφοδεδομένων, το πρώτο πράγμα που θα πρέπει να δούμε είναι το διάγραμμα οντοτήτων συσχετίσεων (π.χ. εικόνα 1) και βάση αυτού να κάνουμε τις κατάλληλες μετατροπές. Αφού το έχουμε μπροστά μας μετατρέπουμε όποιες οντότητες υπάρχουν σε κόμβους (nodes) και τις συσχετίσεις μεταξύ των οντοτήτων σε ακμές (edges), καθώς και όποιον πίνακα που έχει προκύψει μέσω συσχέτισης. Έτσι δίνοντας τα ονόματα στους κόμβους των οντοτήτων και τα ονόματα των συσχετίσεων στις ακμές μπορούμε να έχουμε τον γενικό γράφο (π.χ. εικόνα 2). Μετά πάμε και αφαιρούμε τα ξένα κλειδιά και κρατάμε μόνο τα κλειδιά τα οποία είναι απαραίτητα. Όπως επίσης στον συνολικό γράφο της βάσης για κάθε κόμβο που θα είναι τύπου του κόμβου του γενικού γράφου και θα έχει ως label το κύριο κλειδί του π.χ. θα παίρνει ως properties τα χαρακτηριστικά τα οποία δεν είναι επαναλαμβανόμενα και επίσης υπάρχουν, έτσι δεν χρειάζεται να έχουμε χαρακτηριστικά τα οποία δεν χρησιμοποιούμε. Δηλαδή κάθε γραμμή μιας οντότητας γίνεται κόμβος με label το πρωτεύον κλειδί του κόμβου και κάθε στήλη γίνεται property εφόσον αυτή υπάρχει.



Εικόνα 1 Σχεσιακό διάγραμμα



Εικόνα 2 (Property Graph Data Model)

ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΣΤΟ NEO4J ΚΑΙ ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ ΜΕΣΩ CSV

Αφότου εξαγάγουμε τα δεδομένα από τους πίνακες της βάσης που είχαμε στην mysql σε csv μορφή . Τότε φτιάχνουμε την βάση στο neo4j και βάζουμε τα αρχεία στον φάκελο import. Στην συνέχεια μπορούμε να ενεργοποιήσουμε τον server και να πάμε στο browser του neo4j για να γράψουμε τις εντολές για την καταχώρηση της βάσης.

Σαν πρώτο στάδιο πρέπει να πούμε ότι η τακτική που ακολουθήσαμε ήταν να περάσουμε όλα τα δεδομένα της βάσης από την mysql και δίνοντας επίσης constraint το κύριο κλειδί που είχαμε στην mysql , στην συνέχεια αφού έγιναν οι εισαγωγές των κόμβων και των ιδιοτήτων τους , τότε πήγαμε και φτιάξαμε όλες τις συσχετίσεις που υπάρχουν μεταξύ των κόμβων και σε τελευταίο στάδιο διαγράψαμε τις ιδιότητες τις οποίες τις είχαμε εισάγει μόνο και μόνο για να μας βοηθήσουν για τις συσχετίσεις και όπως ξέρουμε στην nosql δεν χρειαζόμαστε ξένα κλειδιά. Όλες οι εντολές για την δημιουργία της

βάσης βρίσκονται μαζεμένες στο αρχείο «κώδικαςInitCYPHER.txt», αλλά και επίσης θα παρουσιαστούν στα παρακάτω στιγμιότυπα οθόνης.

Στιγμιότυπα οθόνης για την δημιουργία των κόμβων και εισαγωγή μοναδικού χαρακτηριστικού γνωρίσματος:

```
LOAD CSV WITH HEADERS FROM "file:///area.csv" AS row
MERGE (a:Area{areaName: row.areaName})
SET a.regionName=row.regionName
RETURN(a);
CREATE CONSTRAINT FOR (a:Area) REQUIRE (a.areaName) IS UNIQUE;
```

Εικόνα 3: Εισαγωγή κόμβου για την περιοχή.

```
LOAD CSV WITH HEADERS FROM "file:///day.csv" AS row
MERGE (d:Day{dayName: row.dayName})
RETURN(d);
CREATE CONSTRAINT FOR (d:Day) REQUIRE (d.dayName) IS UNIQUE;
```

Εικόνα 4: Εισαγωγή κόμβου για την Ημέρα.

```
LOAD CSV WITH HEADERS FROM "file:///attraction.csv" AS row
MERGE (a:Attraction{attractionName: row.attractionName})
SET a.entranceFee=toFloat(row.entrance_fee),a.website=row.website,a.attr_profilePic=toInteger(row.attr_profilePic),a.areaName=row.areaName,a.attractionType=row.attractionType
RETURN(a);
CREATE CONSTRAINT FOR (a:Attraction) REQUIRE (a.attractionName) IS UNIQUE;
```

Εικόνα 5: Εισαγωγή κόμβου για το αξιοθέατο.

```
LOAD CSV WITH HEADERS FROM "file:///attraction Type.csv" AS row
MERGE (a:Attraction_Type {typeName: row.typeName})
SET a.attrTypeImage = toInteger(row.attrTypeImage)
RETURN a;
CREATE CONSTRAINT FOR (a:Attraction_Type) REQUIRE (a.typeName) IS UNIQUE;
```

Εικόνα 6: Εισαγωγή κόμβου για το τύπο αξιοθέατων.

```
LOAD CSV WITH HEADERS FROM "file:///region.csv" AS row
MERGE (r:Region{regionName: row.regionName})
SET r.countryName=row.countryName
RETURN(r);
CREATE CONSTRAINT FOR (r:Region) REQUIRE (r.regionName) IS UNIQUE;
```

Εικόνα 7: Εισαγωγή κόμβου για τις Περιφέρειες.

```
LOAD CSV WITH HEADERS FROM "file:///country.csv" AS row
MERGE (c:Country{countryName: row.countryName})
RETURN(c);
CREATE CONSTRAINT FOR (c:Country) REQUIRE (c.countryName) IS UNIQUE;
```

Εικόνα 8: Εισαγωγή κόμβου για την Χώρα.

```
LOAD CSV WITH HEADERS FROM "file:///email attraction.csv" AS row
MERGE (e:Email_Attraction{email: row.email})
SET e.attractionName=row.attractionName
RETURN(e);
CREATE CONSTRAINT FOR (e:Email_Attraction) REQUIRE (e.email) IS UNIQUE;
```

Εικόνα 9: Εισαγωγή κόμβου για τα email του αξιοθέατου/

```
LOAD CSV WITH HEADERS FROM "file:///telephone attraction.csv" AS row
MERGE (t:Telephone_Attraction{telephone: row.telephone})
SET t.attractionName=row.attractionName
RETURN(t);
CREATE CONSTRAINT FOR (t:Telephone_Attraction) REQUIRE (t.telephone) IS UNIQUE;
```

Εικόνα 10: Εισαγωγή κόμβου για τα τηλέφωνα ενός αξιοθέατου.

```
LOAD CSV WITH HEADERS FROM "file:///map.csv" AS row
MERGE (m:Map{mapID: toInteger(row.mapID)})
SET m.attractionName=row.attractionName,m.embedded_url=row.embedded_url
RETURN(m);
CREATE CONSTRAINT FOR (m:Map) REQUIRE (m.mapID) IS UNIQUE;
```

Εικόνα 11: Εισαγωγή κόμβου για το iframe ενός αξιοθέατου

```
LOAD CSV WITH HEADERS FROM "file:///operation Time.csv" AS row
MERGE (o:OperationTime{attractionName: row.attractionName,dayName: row.dayName})
SET o.open=row.open,o.close=row.close
RETURN(o);
CREATE CONSTRAINT FOR (o:OperationTime) REQUIRE (o.attractionName,o.dayName) IS UNIQUE;
```

Εικόνα 12: Εισαγωγή κόμβου για το ωράριο λειτουργίας.

```
LOAD CSV WITH HEADERS FROM "file:///picture.csv" AS row
MERGE (p:Picture{pictureID: toInteger(row.pictureID)})
SET p.picture_url=row.picture_url,p.attractionName=row.attractionName,p.username=row.username
RETURN(p);
CREATE CONSTRAINT FOR (p:Picture) REQUIRE (p.pictureID) IS UNIQUE;
```

Εικόνα 13: Εισαγωγή κόμβου για τις φωτογραφίες.

```
LOAD CSV WITH HEADERS FROM "file:///review.csv" AS row
MERGE (r:Review{attractionName: row.attractionName,username: row.username})
SET r.createdAt=row.createdAt,r.comment=row.comment,r.rating=toFloat(row.rating)
RETURN(r);
CREATE CONSTRAINT FOR (r:Review) REQUIRE (r.attractionName,r.username) IS UNIQUE;
```

Εικόνα 14: Εισαγωγή κόμβου για τις κριτικές.

```
LOAD CSV WITH HEADERS FROM "file:///user.csv" AS row
MERGE (u:User{username: row.username})
SET u.name=row.surname,u.lastname=row.lastname,u.email=row.email,u.userProfilePic=toInteger(row.userProfilePic)
RETURN(u);
CREATE CONSTRAINT FOR (u:User) REQUIRE (u.username) IS UNIQUE;
```

Εικόνα 15: Εισαγωγή κόμβου για τους χρήστες.

Στιγμιότυπα οθόνης για την δημιουργία συσχετίσεων μεταξύ των κόμβων

```
MATCH (a:Attraction), (ar:Area)
WHERE a.areaName = ar.areaName
CREATE (a)-[:IN_AREA]->(ar);
```

Εικόνα 16: Συσχέτιση Αξιοθέατου με περιοχή.

```
MATCH (a:Attraction), (p:Picture)
WHERE a.attr_profilePic = p.pictureID
CREATE (a)-[:HAS_PROFILE_PIC]->(p);
```

Εικόνα 17: Συσχέτιση Αξιοθέατου με Φωτογραφία.

```
MATCH (a:Attraction), (at:Attraction_Type)
WHERE a.attractionType = at.typeName
CREATE (a)-[:TYPE_OF]->(at);
```

Εικόνα 18: Συσχέτιση Αξιοθέατου με Τύπο Αξιοθέατου.

```
MATCH (at:Attraction_Type), (p:Picture)
WHERE at.attrTypeImage = p.pictureID
CREATE (at)-[:ATTRACTION_TYPE_IMAGE]->(p);
```

Εικόνα 19: Συσχέτιση Τύπου Αξιοθέατου με Φωτογραφία.

```
MATCH (o:OperationTime), (a:Attraction)
WHERE o.attractionName = a.attractionName
CREATE (o)-[:OF]->(a);
```

Εικόνα 20: Συσχέτιση Ωραρίου λειτουργίας με Αξιοθέατο.

```
MATCH (o:OperationTime), (d:Day)
WHERE o.dayName = d.dayName
CREATE (o)-[:OPERATES_ON]->(d);
```

Εικόνα 21: Συσχέτιση Ωραρίου λειτουργίας με Ημέρα.

```
MATCH (ar:Area), (r:Region)
WHERE ar.regionName = r.regionName
CREATE (ar)-[:IN_REGION]->(r);
```

Εικόνα 22: Συσχέτιση Περιοχής με Περιφέρεια.

```
MATCH (r:Region), (c:Country)
WHERE r.countryName = c.countryName
CREATE (r)-[:IN_COUNTRY]->(c);
```

Εικόνα 23: Συσχέτιση Περιφέρειας με Χώρα.

```
MATCH (u:User), (p:Picture)
WHERE u.userProfilePic = p.pictureID
CREATE (u)-[:USER_PROFILE_PIC]->(p);
```

Εικόνα 24: Συσχέτιση Χρήστη με Φωτογραφία.

```
MATCH (r:Review), (a:Attraction)
WHERE r.attractionName = a.attractionName
CREATE (r)-[:REVIEW_FOR]->(a);
```

Εικόνα 26: Συσχέτιση Κριτικής με Αξιοθέατο.

```
MATCH (e:Email_Attraction), (a:Attraction)
WHERE e.attractionName=a.attractionName
CREATE (e)-[:EMAIL_FOR]->(a);
```

Εικόνα 28: Συσχέτιση Email Αξιοθέατου με Αξιοθέατο.

```
MATCH (m:Map), (a:Attraction)
WHERE m.attractionName=a.attractionName
CREATE (m)-[:IFRAME_FOR]->(a);
```

Εικόνα 30: Συσχέτιση iframe(Χάρτη) με Αξιοθέατο.

```
MATCH (r:Review), (p:Picture)
WHERE r.attractionName = p.attractionName AND r.username=p.username
CREATE (r)-[:PICTURES_OF_REVIEW]->(p);
```

Εικόνα 25: Συσχέτιση Κριτικής με Φωτογραφία.

```
MATCH (r:Review), (u:User)
WHERE r.username=u.username
CREATE (r)-[:REVIEW_FROM]->(u);
```

Εικόνα 27: Συσχέτιση Κριτικής με Χρήστη.

```
MATCH (t:Telephone_Attraction), (a:Attraction)
WHERE t.attractionName=a.attractionName
CREATE (t)-[:TELEPHONE_FOR]->(a);
```

Εικόνα 29: Συσχέτιση Τηλέφωνου Αξιοθέατου με Αξιοθέατο.

Διαγραφή των ιδιοτήτων (ξένα κλειδιά) από τους κόμβους

```
MATCH (a:Attraction)
SET a.areaName = null, a.attractionType = null, a.attr_profilePic = null;
```

Εικόνα 31: Διαγραφή της περιοχής, του τύπου αξιοθέατου και της φωτογραφίας προφίλ από το αξιοθέατο.

```
MATCH (r:Region)
SET r.countryName = null;
```

Εικόνα 33: Διαγραφή της χώρας από την περιφέρεια.

```
MATCH (e:Email_Attraction)
SET e.attractionName = null;
```

Εικόνα 35: Διαγραφή του αξιοθέατου από το email_Αξιοθέατου.

```
MATCH (ar:Area)
SET ar.regionName = null;
```

Εικόνα 32: Διαγραφή της περιφέρειας από την περιοχή.

```
MATCH (at:Attraction_Type)
SET at.attrTypeImage = null;
```

Εικόνα 34: Διαγραφή της φωτογραφίας από τον τύπο αξιοθέατου.

```
MATCH (t:Telephone_Attraction)
SET t.attractionName = null;
```

Εικόνα 36: Διαγραφή του αξιοθέατου από το τηλέφωνο_Αξιοθέατου.

```
MATCH (u:User)
SET u.userProfilePic = null;
```

Εικόνα 37: Διαγραφή της Φωτογραφίας από τον Χρήστη.

```
MATCH (r:Review)
SET r.attractionName = null,r.username = null;
```

Εικόνα 38: Διαγραφή του συνθηματικού του χρήστη και του ονόματος του αξιοθέατου από την κριτική.

```
MATCH (p:Picture)
SET p.attractionName = null,p.username = null;
```

Εικόνα 39: Διαγραφή του συνθηματικού του χρήστη και του ονόματος του αξιοθέατου από την φωτογραφία.

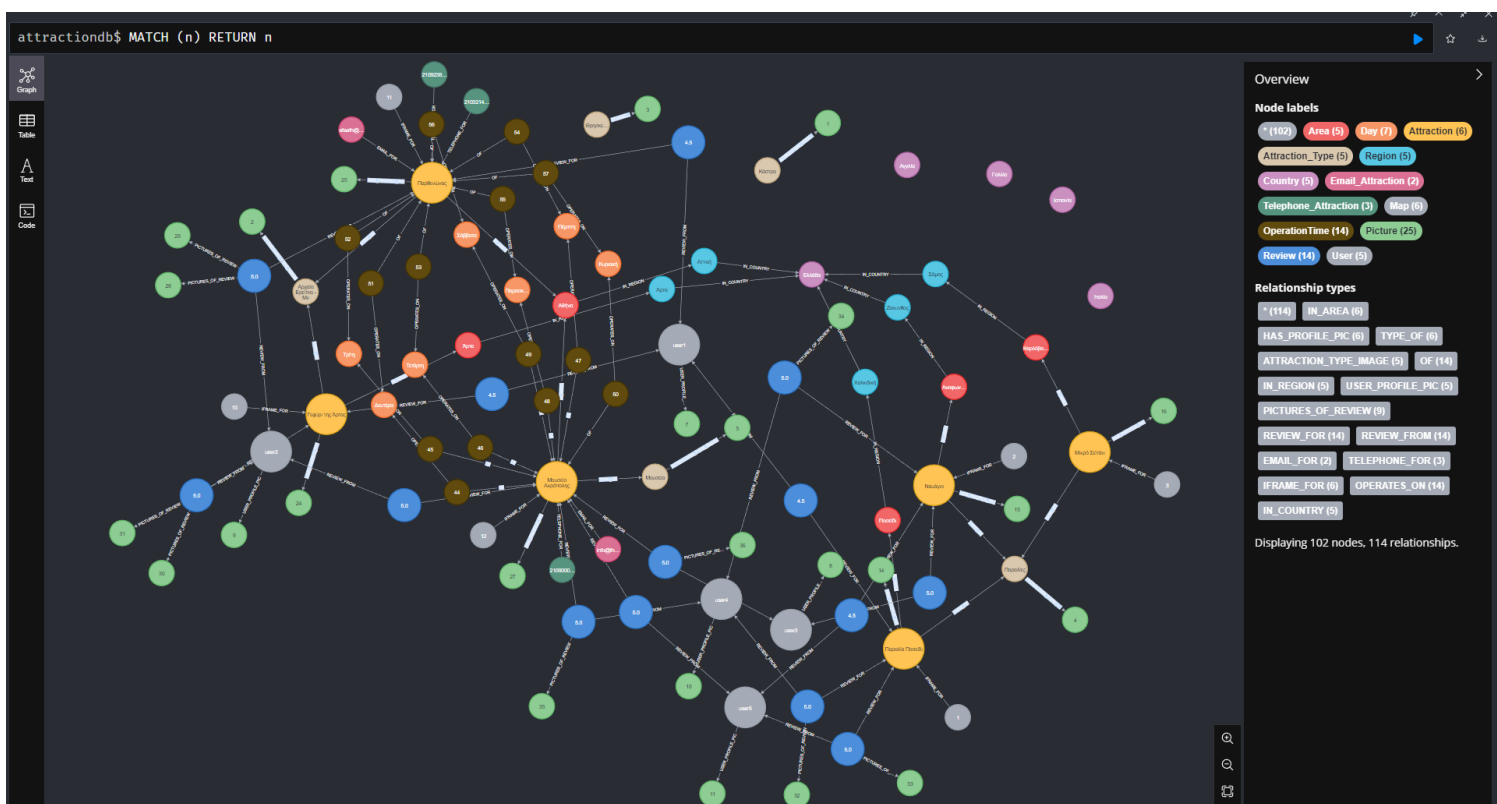
```
MATCH (o:OperationTime)
SET o.attractionName = null,o.dayName = null;
```

Εικόνα 40: Διαγραφή του ονόματος του αξιοθέατου και της ημέρας από το ωράριο λειτουργίας.

```
MATCH (m:Map)
SET m.attractionName = null;
```

Εικόνα 41: Διαγραφή του ονόματος του αξιοθέατου από τον χάρτη.

Ο ΤΕΛΙΚΟΣ ΣΥΓΚΕΝΤΡΩΤΙΚΟΣ ΓΡΑΦΟΣ



Εικόνα 42

ΠΕΝΤΕ ΕΡΩΤΗΜΑΤΑ ΣΤΟ ΝΕΟ4J ΒΑΣΕΙ ΤΟΥ 2^{ΟΥ} ΠΑΡΑΔΟΤΕΟΥ.

Ερώτημα 1^ο :

Σαν 1^ο ερώτημα έχουμε πάρει το **ερώτημα 1.1** από το 2^ο παραδοτέο το οποίο : Επιστρέφει τα 5 κορυφαία αξιοθέατα βάση βαθμολογίας. Υποσημείωση: εδώ θα δούμε μια διαφορά στους επιστρεφόμενους αριθμούς του ερωτήματος στον μέσο όρο και αυτό διότι έχει να κάνει ότι στη MySQL υποστήριζε το DECIMAL και την στρογγυλοποίηση ενώ το neo4j όχι οπότε μας φέρνει τον καθαρό μέσο όρο.

	attractionName	avgRating	numReviews	mainPhoto
▶	Μουσείο Ακρόπολης	5.0	4	Photos\attraction photos\Μουσείο Ακρόπολης.jpg
	Ναυάγιο	4.8	3	Photos\attraction photos\Ναυάγιο Ζάκυνθος.jpg
	Παραλία Ποσειδί	4.8	3	Photos\attraction photos\Ποσειδί.jpg
	Γεφύρι της Άρτας	4.8	2	Photos\attraction photos\ΓεφύριΤηςΆρτας.jpg
	Παρθενώνας	4.8	2	Photos\attraction photos\Παρθενώνας.jpg

Εικόνα 43: Απάντηση Ερωτήματος από την mysql.

```

1 MATCH (r:Review)-[:REVIEW_FOR]-(a:Attraction)
2 MATCH(a)-[:HAS_PROFILE_PICTURE]-(p:Picture)
3 WITH a, AVG(toFloat(r.rating)) AS avgRating, COUNT(r.rating) AS numReviews, p.picture_url AS mainPhoto
4 RETURN a.attractionName, avgRating, numReviews, mainPhoto
5 ORDER BY avgRating DESC, numReviews DESC
6 LIMIT 5;
7

```

	a.attractionName	avgRating	numReviews	mainPhoto
1	"Μουσείο Ακρόπολης"	5.0	4	"Photos/attraction photos/Μουσείο Ακρόπολης.jpg"
2	"Παραλία Ποσειδί"	4.833333333333333	3	"Photos/attraction photos/Ποσειδί.jpg"
3	"Ναυάγιο"	4.833333333333333	3	"Photos/attraction photos/Ναυάγιο Ζάκυνθος.jpg"
4	"Γεφύρι της Άρτας"	4.75	2	"Photos/attraction photos/ΓεφύριΤηςΆρτας.jpg"
5	"Παρθενώνας"	4.75	2	"Photos/attraction photos/Παρθενώνας.jpg"

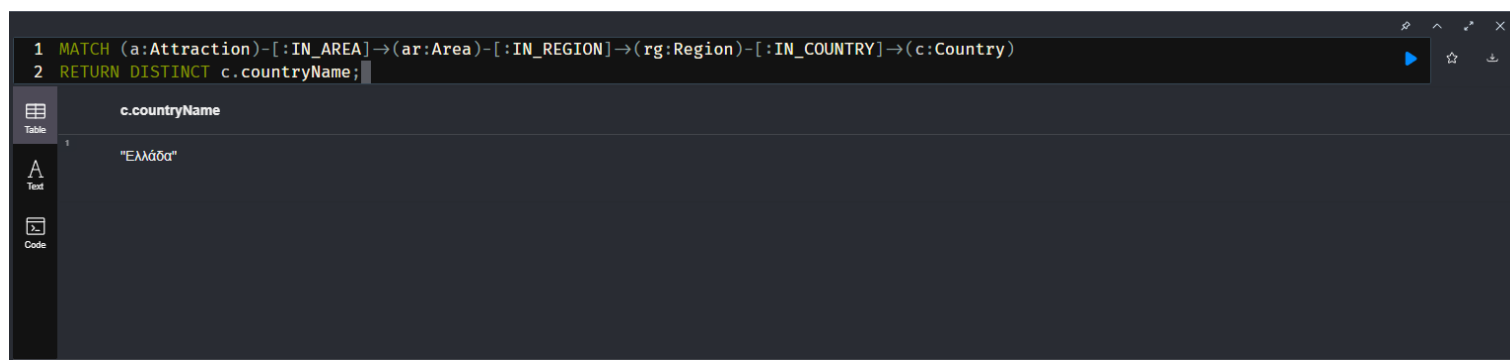
Εικόνα 44: Απάντηση Ερωτήματος από το neo4j.

Ερώτημα 2ο :

Ως 2^ο ερώτημα έχουμε το **ερώτημα 1.5** από το 2^ο παραδοτέο που ζητάει να εμφανίσει όλες τις χώρες που έχουν αξιοθέατα.

	countryName
▶	Ελλάδα

Εικόνα 45: Απάντηση ερωτήματος από την mysql.



Εικόνα 46: Απάντηση ερωτήματος από το neo4j.

Ερώτημα 3ο :

Ως 3^ο ερώτημα έχουμε το **ερώτημα 1.8** από το 2^ο παραδοτέο το οποίο ζητάει από την βάση να επιστρέψει όλα τα ονόματα των τύπων των αξιοθέατων καθώς και το μονοπάτι που είναι αποθηκευμένες οι φωτογραφίες προφίλ του κάθε τύπου ,για τους οποίους τύπους έχουν καταχωρηθεί αξιοθέατα.

	typeName	picture_url
▶	Αρχαία Ερείνια - Μνημεία	Photos\attractionType\ΑρχαίαΕρείνια.jpg
	Παραλίες	Photos\attractionType\Παραλίες.jpg
	Μουσεία	Photos\attractionType\Μουσεία.jpg

Εικόνα 47: Απάντηση ερωτήματος από την mysql.

```

1 MATCH (a:Attraction)-[:TYPE_OF]-(at:Attraction_Type)-[:ATTRACTION_TYPE_IMAGE]-(p:Picture)
2 RETURN DISTINCT at.typeName,p.picture_url;
    
```

	at.typeName	p.picture_url
1	"Μουσεία"	"Photos/attractionType/Μουσεία.jpg"
2	"Αρχαία Ερείπια - Μνημεία"	"Photos/attractionType/ΑρχαίαΕρείπια.jpg"
3	"Παραλίες"	"Photos/attractionType/Παραλίες.jpg"

Εικόνα 48: Απάντηση ερωτήματος από το neo4j.

Ερώτημα 4ο :

Ως 4^ο ερώτημα έχουμε το **ερώτημα 1.19** από το 2^ο παραδοτέο το οποίο ζητάει από την βάση να επιστρέψει απ' όλες τις κριτικές για ένα αξιοθέατο , όλα τα χαρακτηριστικά του όνομα αξιοθέατου , όνομα του λογαριασμού του χρήστη που την καταχώρησε , την βαθμολογία , την χρονική στιγμή που δημιουργήθηκε, το σχόλιο καθώς και την φωτογραφία του προφίλ του χρήστη.

createdAt	comment	rating	attractionName	username	picture_url
2023-12-14 10:26:11	ιστορία	4.5	Παρθενώνας	user1	Photos\user\user1.jpg
2023-12-14 10:26:11	Ιστορία! Φανταστικό μέρος! Πρέπει να δείτε! Χαίρομαι που βλέπουμε την πρόοδο της ανακαίνισης αυτών των όμορφων κτιρίων! Μια ανάμνηση...	5.0	Παρθενώνας	user2	Photos\user\user2.jpg

Εικόνα 49: Απάντηση ερωτήματος από την mvSQL.

```

1 MATCH (r:Review)-[:REVIEW_FOR]-(a:Attraction { attractionName: 'Παρθενώνας' })
2 MATCH (r)-[:REVIEW_FROM]-(u:User)-[:USER_PROFILE_PICTURE]-(p:Picture)
3 RETURN r.createdAt AS createdAt,
4         r.comment AS comment,
5         r.rating AS rating,
6         u.username AS username,
7         a.attractionName AS attractionName,
8         p.picture_url AS picture_url
9 ORDER BY r.createdAt DESC
10 LIMIT 3;
    
```

	createdAt	comment	rating	username	attractionName	picture_url
1	"2023-12-16 23:45:17"	"Ιστορία! Φανταστικό μέρος! Πρέπει να δείτε! Χαίρομαι που βλέπουμε την πρόοδο της ανακαίνισης αυτών των όμορφων κτιρίων! Μια ανάμνηση ζωής για κάθε επισκέπτη! Συνιστάται ιδιαίτερα να έρθετε και να δείτε αν βρίσκεστε στην Αθήνα. Αγοράστε το εισιτήριό σας ηλεκτρονικά και ελάτε νωρίς καθώς η ζέση μπορεί να ανέβει ψηλά και το πλήθος του κόσμου"	5.0	"user2"	"Παρθενώνας"	"Photos/user/user2.jpg"
2	"2023-12-16 23:44:38"	null	4.5	"user1"	"Παρθενώνας"	"Photos/user/user1.jpg"

Εικόνα 50: Απάντηση ερωτήματος από το neo4j.

Ερώτημα 5^ο :

Ως 5^ο ερώτημα έχουμε το **ερώτημα 1.20** από το 2^ο παραδοτέο το οποίο ζητάει από την βάση να επιστρέψει για ένα δοσμένο όνομα λογαριασμού χρήστη , τα παρακάτω στοιχεία για όλες τις κριτικές που έχει κάνει. Τα οποία στοιχεία είναι: το όνομα του αξιοθέατου της κριτικής , τη βαθμολογία που έχει δώσει, το σχόλιο ,και την χρονική στιγμή της δημιουργίας της κριτικής.

attractionName	rating	comment	createdAt
Γεφύρι της Άρτας	5.0	Το γεφύρι της Άρτας δεσπόζει στον Άραχθο και είναι ορατό από τον περιφερειακό της πόλης. Είναι προσπελάσιμο και περπατήσιμο από το πλάι, μέ...	2023-12-14 10:26:11
Μουσείο Ακρόπολης	5.0		2023-12-14 10:26:11
Παρθενώνας	5.0	Ιστορία! Φανταστικό μέρος! Πρέπει να δείτε! Χαίρομαι που βλέπουμε την πρόοδο της ανοικοδόμησης αυτών των όμορφων κτιρίων! Μια ανάμνηση...	2023-12-14 10:26:11

0

Εικόνα 51: Απάντηση ερωτήματος από τη *mysql*.

```
1 MATCH (r:Review)-[:REVIEW_FROM]-(u:User { username: 'user2' })
2 MATCH (r)-[:REVIEW_FOR]-(a:Attraction)
3 RETURN a.attractionName AS attractionName,
4         r.rating AS rating,
5         r.comment AS comment,
6         r.createdAt AS createdAt
7 ORDER BY r.createdAt;
```

	attractionName	rating	comment	createdAt
1	"Μουσείο Ακρόπολης"	5.0	null	"2023-12-16 23:45:17"
2	"Παρθενώνας"	5.0	"Ιστορία! Φανταστικό μέρος! Πρέπει να δείτε! Χαίρομαι που βλέπουμε την πρόοδο της ανοικοδόμησης αυτών των όμορφων κτιρίων! Μια ανάμνηση ζωής για κάθε επισκέπτη! Συνιστάται ιδιαίτερα να έρθετε και να δείτε αν βρίσκεστε στην Αθήνα. Αγοράστε το εισιτήριό σας ηλεκτρονικά και ελάτε νωρίς καθώς η ζέστη μπορεί να ανέβει ψηλά και το πλήθος του κόσμου"	"2023-12-16 23:45:17"
3	"Γεφύρι της Άρτας"	5.0	"Το γεφύρι της Άρτας δεσπόζει στον Άραχθο και είναι ορατό από τον περιφερειακό της πόλης. Είναι προσπελάσιμο και περπατήσιμο από το πλάι, μέσα από ένα πεζόδρομο. Μπορείτε να βγάλετε φωτογραφία με μέτωπο 3/4 από το παρακείμενο cafe. Είναι μακρύ, εντυπωσιακό, και μπορείτε να το περπατήσετε και να νιώσετε όπως οι ταξιδιώτες του 17ου αιώνα, πάνω από τις καμάρες, στην λιθόστρωτη επιφάνεια."	"2023-12-16 23:45:17"

Εικόνα 52: Απάντηση ερωτήματος από το *neo4j*