



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

Course : M 149. Database Management Systems

Student : Iordanis Kousis – M1596

Programming Project I



M 149. Συστήματα Βάσεων Δεδομένων

Εργασία #1

Ιορδάνης Κούσης – M1596

Συνημμένα : Παραρτήματα 1,2 και 3

Αθήνα 11/12/2018

Θέμα : Υλοποίηση Σχεσιακής Βάσης Δεδομένων

Σύντομη αναφορά / τεκμηρίωση - Κύριο Σώμα

1. Γενικά

Στην παρούσα εργασία ζητήθηκε η υλοποίηση μιας βάσης δεδομένων(ΒΔ) με αξιοποίηση στοιχείων προερχόμενα από πίνακες αναφορών διάφορων προβλημάτων/γεγονότων ενός δήμου και που αφορούν την ποιότητα ζωής των πολιτών

2. Διαμόρφωση της Βάσης Δεδομένων

Οι πίνακες αρχικά συλλέχθηκαν από τον ιστότοπο που δίδεται στην εκφώνηση σε μορφή αρχείων **excel**. Η πρώτη διαμόρφωση πριν την έναρξη υλοποίησης της βάσης δεδομένων(ΒΔ) έγινε σε αυτούς τους πίνακες. Συγκεκριμένα βρέθηκαν και απομακρύνθηκαν διπλές και λάθος εγγραφές και δημιουργήθηκαν νέοι πίνακες σύμφωνα με τους **κανόνες κανονικοποίησης** για την διατήρηση των δεδομένων και την ελαχιστοποίηση των πλεονασμών. Το σχήμα που προέκυψε τελικά εμφανίζεται στο Παράρτημα "1" της παρούσας αναφοράς.

Για την διαμόρφωση της ΒΔ δημιουργήθηκε ένας κεντρικός πίνακας (σχέση) **"incident"** ο οποίος περιέχει όλα τα κοινά στοιχεία των γεγονότων όλων των τύπων. Στη συνέχεια δημιουργήθηκαν πίνακες ξεχωριστά για κάθε ένα είδος αναφοράς οι οποίοι περιέχουν τα ειδικά χαρακτηριστικά. Οι εν λόγω πίνακες συνδέονται με τον πίνακα "incident" με το πρωτεύον κλειδί του εν λόγω πίνακα, που είναι το **inc_id**. Επίσης δημιουργήθηκαν επιπλέον δύο σχέσεις **"incident type"** και **"status type"** με τα χαρακτηριστικά που έχουν όλες οι σχέσεις όπως έναν τύπο (incident type – 11 τύποι γεγονότος) και μια κατάσταση (incident status – ολοκληρωμένο ή μη γεγονός).

Στη συνέχεια έγινε υλοποίηση της βάσης δεδομένων με χρήση του εργαλείου **PgAdmin 4** της σχεσιακής βάσης δεδομένων **PostgreSQL** σύμφωνα με τις απαιτήσεις της εκφώνησης και εισαγωγή των δεδομένων από τους πίνακες CSV από το **SQL Shell (psql)**.



3. Υλοποίηση Εφαρμογής και Ιστοσελίδας

Η γλώσσα προγραμματισμού που επιλέχθηκε για την υλοποίηση της εφαρμογής (**app**) και του interface (**website**) είναι η **Python** και ως εκ τούτου έγινε χρήση του **API psycopg2** (<https://pypi.org/project/psycopg2/>) για την διασύνδεση της γλώσσας με την βάση **PostgreSQL**.

Το Web Framework που επιλέχθηκε είναι μια παραλλαγή του **Bottle** (<https://bottlepy.org/docs/dev/tutorial.html>) της Python για PostgreSQL, το **bottle-pgsql** (<https://pypi.org/project/bottle-pgsql/>).

Οι φόρμες και το index γράφτηκαν σε γλώσσα HTML5.

4. Υλοποίηση ερωτημάτων (Queries)

Στην εργασία ζητείται η υλοποίηση συνολικά δώδεκα(12) ερωτημάτων (queries), τα οποία αφού υλοποιήθηκαν στη γλώσσα **SQL** ελέγχθηκαν για την επιβεβαίωση της ορθής λειτουργίας στο shell με εισαγωγή ενδεικτικών δεδομένων από τη βάση.

Στη συνέχεια για το σύνολο των ερωτημάτων υλοποιήθηκαν οι ενσωματωμένες συναρτήσεις (Stored Functions) που εμφανίζονται στο συνημμένο Παράρτημα "2".

5. Ιστοσελίδα (Web Page)

Στο Παράρτημα "3" έχουν αποτυπωθεί διάφορα στιγμιότυπα της ιστοσελίδας στην οποία υπάρχουν οι φόρμες, login/signup, εισαγωγής στοιχείων καθώς και επεξηγήσεις επί της σελίδας, για όπου απαιτείται. Η πρόσβαση στη σελίδα γίνεται με τον σύνδεσμο <http://localhost:8080/> αφού πρώτα τρέξει η εφαρμογή (**>py website.py**) από ένα command line (λειτουργικό σύστημα Windows).

6. Πέρα από τα ερωτήματα που δίδονται, ζητούνται να υλοποιηθούν επιπλέον :

α. Γενικότερα ερωτήματα που αφορούν την εμφάνιση δεδομένων των γεγονότων (incidents) με εισαγωγή zip code, street ή issue. Η υλοποίησή τους έγινε εντός της εφαρμογής (ερωτήματα Q13, Q14, Q15).

β. Δυνατότητα εισαγωγής στοιχείων (νέων γεγονότων). Η υλοποίηση έγινε επίσης εντός της εφαρμογής και ενδεικτικά για την εισαγωγή δεδομένων στον πίνακα "incident" και "tree_trims". Εύκολα μπορεί να υλοποιηθεί με τον ίδιο τρόπο για τους υπόλοιπους τύπους γεγονότων.



7. Διαδικασίες Login και Register(Sign up)

Για την εισαγωγή στην ιστοσελίδα (**login**) απαιτείται αρχικά η εισαγωγή **login name και password** που αρχικά πρέπει να έχουν καταχωρηθεί με την **διαδικασία εγγραφής (sign up)**. Για την υλοποίηση της διαδικασίας registration-login έγινε με χρήση ενός πίνακα **"users"** εντός της βάσης ο οποίος ελέγχεται κάθε φορά για την μοναδικότητα των στοιχείων που δίνονται είτε πρόκειται για login είτε για registration διαδικασία.

8. Αναφορές

Παρακάτω παρατίθενται σύνδεσμοι από ιστότοπους όπου αντλήθηκαν τόσο χρήσιμες πληροφορίες για την λειτουργία της PostgreSQL όσο και προγράμματα/κώδικες που χρησιμοποιήθηκαν στην εργασία.

- α. <http://www.postgresqltutorial.com/>
- β. <https://stackoverflow.com/>
- γ. <https://pypi.org/project/bottle-pgsql/>
- δ. <https://pypi.org/project/psycopg2/>
- ε. <https://www.postgresql.org>

Βιβλιογραφία :

- α. DB System Concepts -6th edition(Abraham Silberschatz-Henry F.Korth-S. Sudarshan)
- β. Ύλη (παρουσιάσεις- σημειώσεις- εργαστηριακό υλικό) από μάθημα "Σχεδίαση και Χρήση Βάσεων Δεδομένων" του προπτυχιακού κύκλου 4ου εξαμήνου Τμήματος Πληροφορικής και Τηλεπικοινωνιών
- γ. Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων - 7th edition(R.Elmasri-S.B.Navathe)
- δ. Ύλη (παρουσιάσεις- σημειώσεις) από μάθημα "Συστήματα Βάσεων Δεδομένων" του μεταπτυχιακού κύκλου του Τμήματος Πληροφορικής και Τηλεπικοινωνιών

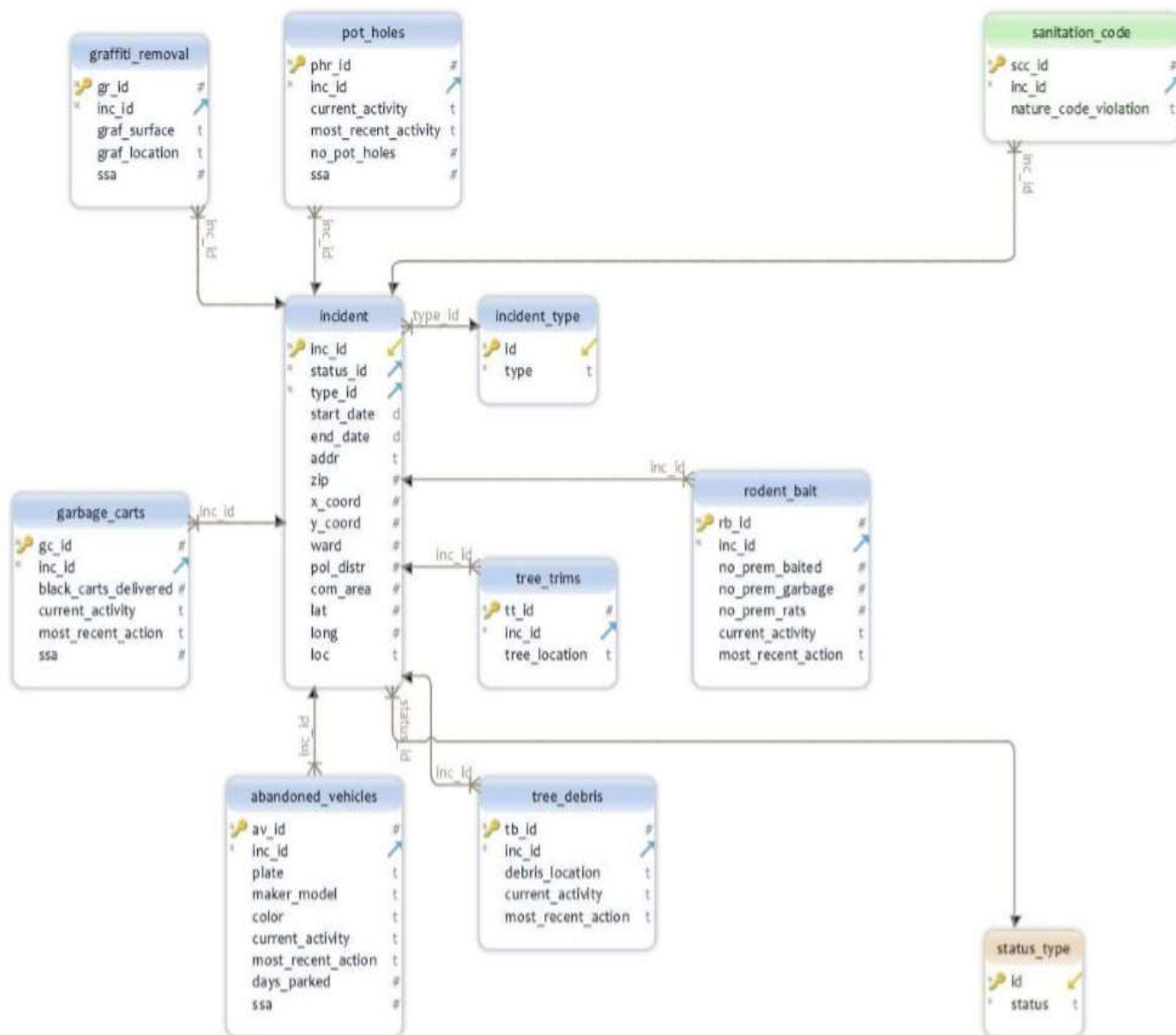




Θέμα : Υλοποίηση Σχεσιακής Βάσης Δεδομένων

Παράρτημα "1"

Το σχήμα (Schema) που προέκυψε μετά την επεξεργασία της βάσης, σε μορφή διαγράμματος έχει ως ακολούθως :



Εικόνα 1 : Διαγραμματική διάταξη σχέσεων ΒΔ 311ci - Για την εξαγωγή του σχήματος έγινε χρήση του εργαλείου DBSchema.



Επίσης παρατίθενται παρακάτω οι πίνακες και στήλες τους:

<div>incident</div> <div><div>column_name data_type</div><div>-----+-----</div><div>inc_id integer</div><div>status_id integer</div><div>type_id integer</div><div>start_date date</div><div>end_date date</div><div>addr character varying</div><div>zip integer</div><div>x_coord double precision</div><div>y_coord double precision</div><div>ward integer</div><div>pol_distr integer</div><div>com_area integer</div><div>lat double precision</div><div>long double precision</div><div>loc character varying</div></div>	<div>incident_type</div> <div><div>column_name data_type</div><div>-----+-----</div><div>id integer</div><div>type character varying</div></div>	<div>status_type</div> <div><div>column_name data_type</div><div>-----+-----</div><div>id integer</div><div>status character varying</div></div>
<div>garbage_carts</div> <div><div>column_name data_type</div><div>-----+-----</div><div>-----</div><div>gc_id integer</div><div>inc_id integer</div><div>black_carts_delivered integer</div><div>current_activity character varying</div><div>most_recent_action char varying</div><div>ssa integer</div></div>	<div>graffiti_removal</div> <div><div>column_name data_type</div><div>-----+-----</div><div>-</div><div>gr_id integer</div><div>inc_id integer</div><div>graf_surface character varying</div><div>graf_location character varying</div><div>ssa integer</div></div>	<div>pot_holes</div> <div><div>column_name data_type</div><div>-----+-----</div><div>-----</div><div>phr_id integer</div><div>inc_id integer</div><div>current_activity character varying</div><div>most_recent_activity character varying</div><div>ssa integer</div></div>
<div>rodent_bait</div> <div><div>column_name data_type</div><div>-----+-----</div><div>-----</div><div>rb_id integer</div><div>inc_id integer</div><div>no_prem_baited integer</div><div>no_prem_garbage integer</div><div>no_prem_rats integer</div><div>current_activity char varying</div><div>most_recent_action char varying</div></div>	<div>sanitation_code</div> <div><div>column_name data_type</div><div>-----+-----</div><div>-----</div><div>scc_id integer</div><div>inc_id integer</div><div>nature_code_violation char vary</div></div>	<div>tree_debris</div> <div><div>column_name data_type</div><div>-----+-----</div><div>-----</div><div>tb_id integer</div><div>inc_id integer</div><div>debris_location character varying</div><div>current_activity char varying</div><div>most_recent_action chara varyin</div></div>
<div>tree_trim</div> <div><div>column_name data_type</div><div>-----+-----</div><div>-</div><div>tt_id integer</div><div>inc_id integer</div><div>tree_location character varying</div></div>		



M 149. Συστήματα Βάσεων Δεδομένων

Ιορδάνης Κούσης – M1596

Εργασία #1

Αθήνα 11/12/2018

Θέμα : Υλοποίηση Σχεσιακής Βάσης Δεδομένων

Σύντομη αναφορά / τεκμηρίωση - Παράρτημα "2"

Οι συναρτήσεις που χρησιμοποιήθηκαν για την απάντηση των ερωτημάτων queries της εκφώνησης και ενσωματώθηκαν ως ενσωματωμένες συναρτήσεις (**stored functions**) είναι ως ακολούθως:

#query 1 - function q1

```
CREATE OR REPLACE FUNCTION q1(d1 date,d2 date)
RETURNS TABLE (type integer, count bigint)
AS $$
SELECT type_id, count(*) FROM incident
WHERE start_date BETWEEN d1 AND d2
GROUP BY type_id
ORDER BY 2 desc;$$
LANGUAGE SQL;
```

#query 2 - function q2

```
CREATE OR REPLACE FUNCTION q2(t integer, d1 date,d2 date)
RETURNS TABLE (date date, count bigint)
AS $$
SELECT start_date, count(*) FROM incident
WHERE type_id = t
AND start_date BETWEEN d1 AND d2
GROUP BY start_date;$$
LANGUAGE SQL;
```

#query 3 - function q3

```
CREATE OR REPLACE FUNCTION q3(d date)
RETURNS TABLE (zip integer,max_type bigint,zip1 integer,type integer,inc_id bigint)
AS $$
WITH commonRequestPerZipType(zip, type,inc_id) AS(
SELECT zip, type_id, count(*) inc_id
FROM incident
WHERE start_date = d
GROUP by zip, type_id)
SELECT * FROM (select zip, max(inc_id) max
FROM commonRequestPerZipType group by zip)
AS a INNER JOIN commonRequestPerZipType
AS b ON b.zip = a.zip AND b.inc_id = a.max;$$
LANGUAGE SQL;
```



#query 4 - function q4

```
CREATE OR REPLACE FUNCTION q4(d1 date,d2 date)
RETURNS TABLE (type integer, average integer)
AS $$
SELECT type_id, avg(end_date::date - start_date::date)::int AS days
FROM incident
WHERE start_date between d1 AND d2 AND status_id IN (1,2)
GROUP BY type_id;$$
LANGUAGE SQL;
```

#query 5 - function q5

```
CREATE OR REPLACE FUNCTION q5(d date,x1 real,x2 real,y1 real,y2 real)
RETURNS TABLE (type varchar, numOfRequests bigint)
AS $$
SELECT b.type, count(*) AS numOfRequests
FROM incident a, incident_type b
WHERE a.type_id=b.id
AND lat BETWEEN x1 and x2
AND long BETWEEN y1 and y2
AND start_date = d
GROUP by b.type
ORDER by numOfRequests
FETCH FIRST 1 ROW ONLY;$$
LANGUAGE SQL;
```

#query 6 - function q6

```
CREATE OR REPLACE FUNCTION q6(d1 date,d2 date)
RETURNS TABLE (ssa integer, num bigint, date date)
AS $$
SELECT ssa, count(*) num, sdate FROM (
(SELECT start_date AS sdate, av.ssa AS ssa
FROM incident LEFT JOIN abandoned_vehicles AS av ON av.inc_id = incident.inc_id
WHERE type_id = 1
AND start_date BETWEEN d1 AND d2
AND av.ssa IS NOT NULL)
UNION ALL
(SELECT start_date AS sdate, gb.ssa AS ssa
FROM incident LEFT JOIN garbage_carts AS gb ON gb.inc_id = incident.inc_id
WHERE type_id = 3
AND start_date BETWEEN d1 and d2
AND gb.ssa IS NOT NULL)
UNION ALL
(SELECT start_date AS sdate, gr.ssa AS ssa
FROM incident LEFT JOIN graffiti_removal AS gr ON gr.inc_id = incident.inc_id
WHERE type_id = 4
AND start_date BETWEEN d1 AND d2
AND gr.ssa IS NOT null)
UNION ALL
(SELECT start_date AS sdate, ph.ssa AS ssa
FROM incident LEFT JOIN pot_holes AS ph ON ph.inc_id = incident.inc_id
WHERE type_id = 5
AND start_date BETWEEN d1 AND d2
AND ph.ssa IS NOT NULL)) AS ss
GROUP BY sdate, ssa
ORDER BY num DESC
FETCH FIRST 5 ROWS ONLY;$$
LANGUAGE SQL;
```




#query 7 - function q7

```
CREATE OR REPLACE FUNCTION q7()  
RETURNS TABLE (plate VARCHAR, count BIGINT)  
AS $$  
SELECT DISTINCT plate, COUNT(*)  
FROM abandoned_vehicles  
GROUP BY plate  
HAVING COUNT(*) > 1 ORDER BY plate;$$  
LANGUAGE SQL;
```

#query 8 - function q8

```
CREATE OR REPLACE FUNCTION q8()  
RETURNS TABLE (color VARCHAR)  
AS $$  
SELECT * from (SELECT color FROM abandoned_vehicles  
GROUP BY color ORDER BY COUNT(*) DESC LIMIT 2) AS most OFFSET 1 ROWS;$$  
LANGUAGE SQL;
```

#query 9 - function q9

```
CREATE OR REPLACE FUNCTION q9(a integer)  
RETURNS TABLE (no_prem_baited integer)  
AS $$  
select inc_id from rodent_bait where no_prem_baited<a order by inc_id;$$  
LANGUAGE SQL;
```

#query 10 - function q10

```
CREATE OR REPLACE FUNCTION q10(b integer)  
RETURNS TABLE (no_prem_garbage integer)  
AS $$  
SELECT inc_id FROM rodent_bait WHERE no_prem_garbage<b ORDER BY inc_id;$$  
LANGUAGE SQL;
```

#query 11 - function q11

```
CREATE OR REPLACE FUNCTION q11(c integer)  
RETURNS TABLE (no_prem_rats integer)  
AS $$  
SELECT inc_id FROM rodent_bait WHERE no_prem_rats<c ORDER BY inc_id;$$  
LANGUAGE SQL;
```



#query 12 - function q11

```
CREATE OR REPLACE FUNCTION q12(d date)
RETURNS TABLE (police_district integer)
AS $$
(SELECT b.pol_distr
FROM pot_holes a, incident b
WHERE a.inc_id = b.inc_id
AND b.pol_distr IS NOT NULL
AND a.no_pot_holes>0
AND b.end_date = d
GROUP BY b.pol_distr
ORDER BY b.pol_distr)
INTERSECT
(SELECT b.pol_distr
FROM rodent_bait a, incident b
WHERE a.inc_id = b.inc_id
AND b.pol_distr IS NOT NULL
AND a.no_prem_baited > 0
AND b.end_date = d
GROUP BY b.pol_distr
ORDER BY b.pol_distr);$$
LANGUAGE SQL;
```



Μ 149. Συστήματα Βάσεων Δεδομένων

Εργασία #1

Ιορδάνης Κούσης – Μ1596

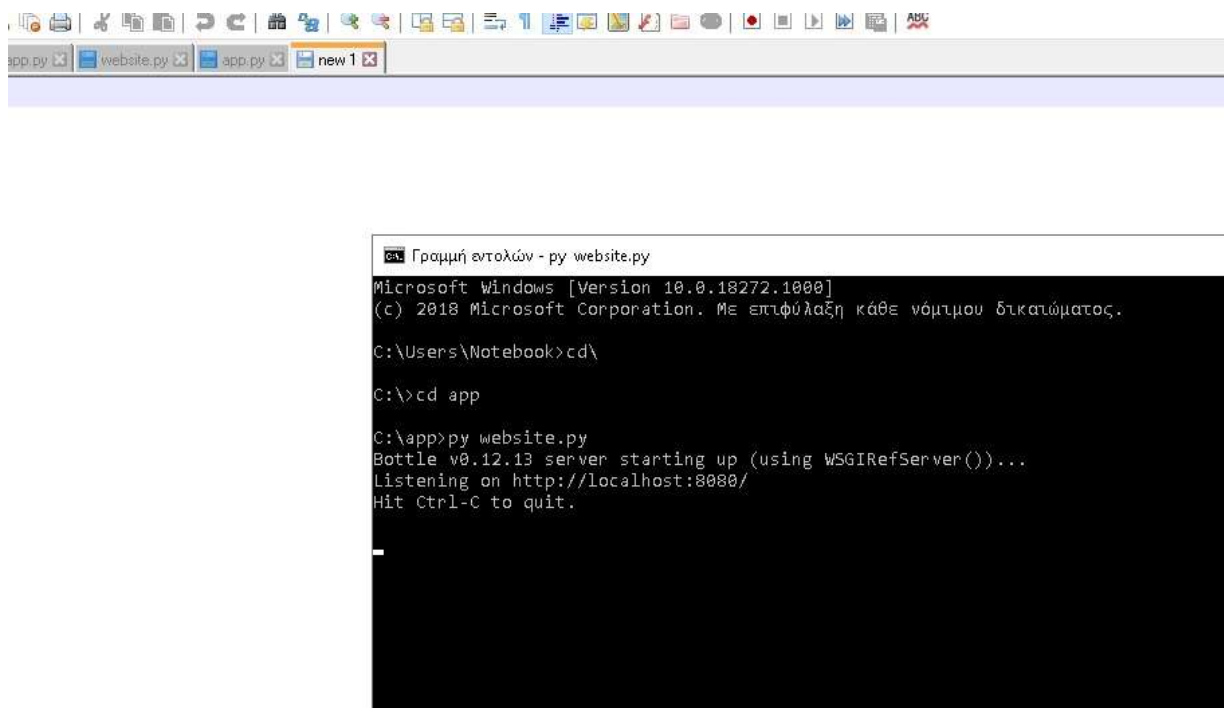
Αθήνα 11/12/2018

Θέμα : Υλοποίηση Σχεσιακής Βάσης Δεδομένων

Σύντομη αναφορά / τεκμηρίωση - Παράρτημα "3"

Παρατίθενται στιγμιότυπα από την εφαρμογή και ιστοσελίδα :

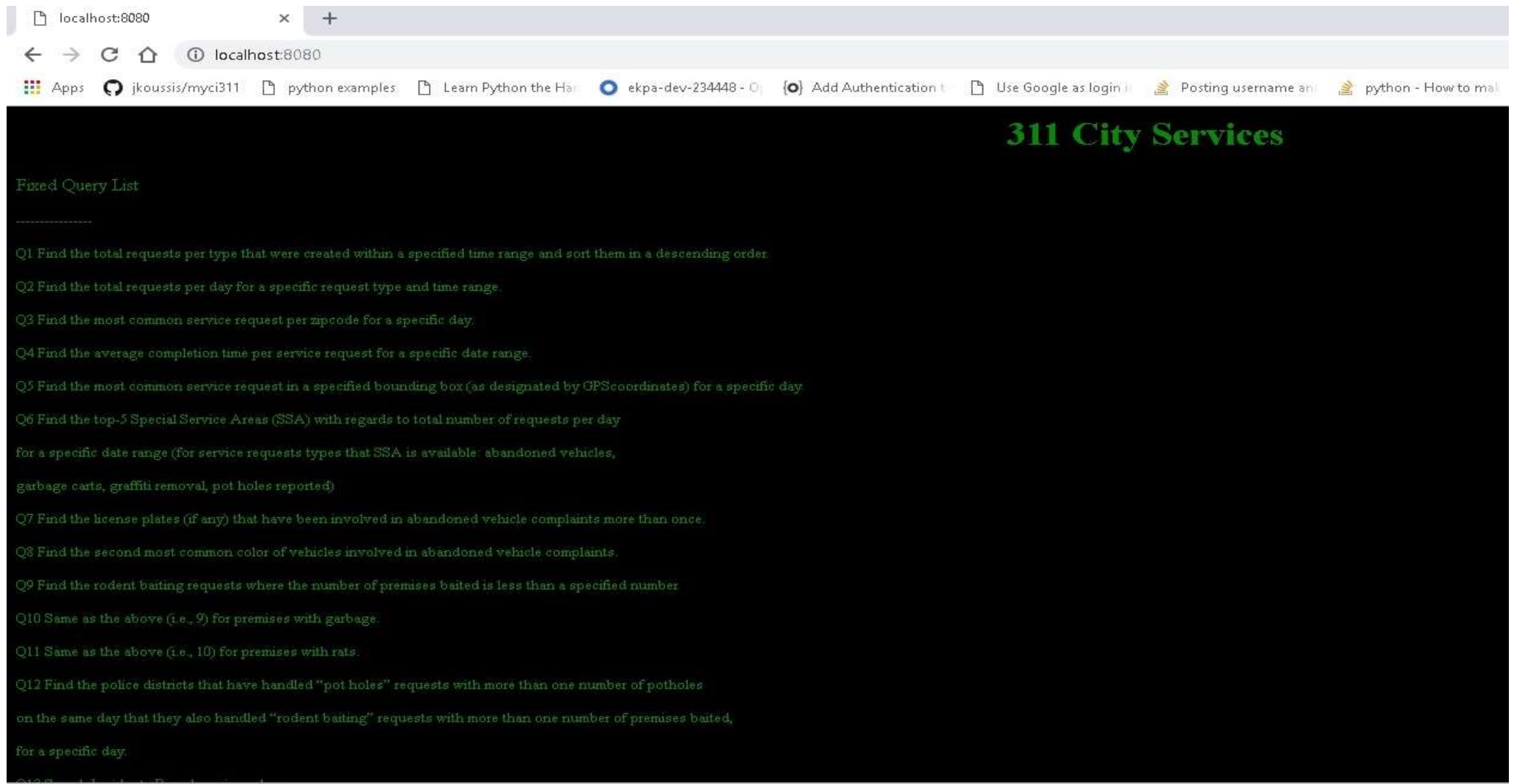
Εικόνα 1: Φόρτωση του handler αρχείου website.py



Εικόνα 1



Εικόνα 2 : Φόρτωση σελίδας με localhost:8080 από οποιοδήποτε φυλλομετρητή. Εδώ φαίνεται το αρχικό άνω τμήμα της σελίδας όπου δίνονται απαραίτητες επεξηγήσεις στο χρήστη



Εικόνα 2



Εικόνα 3 : Εισαγωγή στοιχείων και εξαγωγή αποτελεσμάτων στο τμήμα αποτελεσμάτων στο κάτω μέρος της σελίδας για το ερώτημα Q1 (πρώτο ερώτημα της εργασίας)

The screenshot shows a web application interface with a grid of 15 question forms (Q1-Q15) and a table of incident data. The forms are arranged in a 3x5 grid. Q1-Q15 are for data entry, while Q16-Q18 are for data retrieval. The table at the bottom shows incident data.

Type_id	total
4	282
5	123
8	120
1	81
3	48
6	38
9	37
7	33
2	21
10	18

Εικόνα 3



Εικόνα 4 : Εισαγωγή στοιχείων και εξαγωγή αποτελεσμάτων στο τμήμα αποτελεσμάτων στο κάτω μέρος της σελίδας για το ερώτημα Q13 (επιπλέον ερώτημα για εξαγωγή στοιχείων βασισμένα στο zip code)

The screenshot shows a web application interface with a grid of 15 question forms (Q1-Q15) and a table of incident data. The interface is in Greek and includes input fields for dates, numbers, and text, along with submit and search buttons. The table at the bottom displays incident details including incident ID, status, type, and date reported.

inc_id	status_id	type_id	date_reported
1100002615	1	1	2011-01-01
1300001328	1	1	2013-01-01
1200003939	1	1	2012-01-02
1300004280	2	1	2013-01-02
1300007907	1	1	2013-01-02
1100010565	1	1	2011-01-03
1100011746	1	1	2011-01-03
1200006738	1	1	2012-01-03
1300012567	2	1	2013-01-03

Εικόνα 4



Εικόνα 5 : Φόρμα εισαγωγής στοιχείων login και registration

The image shows a web browser window displaying a form titled "Welcome - login or sign-up form". The form is divided into two main sections: "login" and "registration".

login section:

- Enter login name:
- Enter password:
- Submit:

registration section:

- Enter a login name:
- Enter a password:
- Submit:

The browser's address bar shows the file path: file:///C:/app/web/index.html. The Windows taskbar at the bottom indicates the time is 1:57 PM on 10/12/2018.

Εικόνα 5