Συστήματα Διαχείρισης και Ανάλυσης Δεδομένων

2ο Project

Παντελίδης Ιπποκράτης – 3210150

Ζήτημα 1ο

Στο ζήτημα αυτό θα δημιουργήσουμε το λογικό σχήμα της αποθήκης και θα το φορτώσουμε με τα απαραίτητα δεδομένα. Πιο συγκεκριμένα θα ακολουθήσουμε τα ακόλουθα 4 βήματα.

Βήμα 1)

i) Αρχικά θα φτιάξουμε την βάση δεδομένων στον SQL Server με όνομα INSDW χρησιμοποιώντας την εντολή `CREATE DATABASE INSDW;`

ii) Στην συνέχεια θα κατασκευάσουμε τον πίνακα με όνομα inspections_data, όπως ακριβώς μας δίνεται στην εκφώνηση, χρησιμοποιώντας την ακόλουθη εντολή στην SQL :

```
CREATE TABLE inspections_data (
    rid INT,
    lat FLOAT,
    lon FLOAT,
    insdate DATE,
    insyear INT,
    insmonth INT,
    insday INT,
    insweekday INT,
    inscode INT,
    instype NVARCHAR(100),
    criticalIssue INT,
    nonCriticalIssue INT,
    vcode INT,
    vdescription NVARCHAR(255),
    vcategory NVARCHAR(255)
);
```

iii) Τέλος, με την εντολή που μας δίνεται στην εκφώνηση θα φορτώσουμε τα δεδομένα του αρχείου inspections_data.txt στον πίνακα που φτιάξαμε στο προηγούμενο βήμα :

```
BULK INSERT inspections_data
FROM 'C:\inspections_data.txt'
WITH (DATAFILETYPE = 'widechar', FIRSTROW =2, FIELDTERMINATOR= '|',
ROWTERMINATOR = '\n');
```

Βήμα 2)

Για να υλοποιήσουμε το λογικό σχήμα της αποθήκης δεδομένων με την μορφή αστέρα θα πρέπει πρώτα να αντλήσουμε από την περιγραφή που μας δίνεται τις διαστάσεις και τα μετρήσιμα της αποθήκης. Πιο συγκεκριμένα :

> Η υπηρεσία δημόσιας υγιεινής ενδιαφέρεται να αναπτύξει μια αποθήκη για την άντληση χρήσιμων πληροφοριών σχετικά με τα στοιχεία των επιθεωρήσεων. Οι απαιτήσεις της υπηρεσίας δημόσιας υγιεινής εστιάζουν μεταξύ άλλων στην ανάλυση του αριθμού των επιθεωρήσεων και των παραβάσεων ανα τύπο επιθεώρησης, κατηγορία παράβασης, περιοχή, καθώς και οποιονδήποτε συνδυασμό αυτών. Εξυπακούεται ότι στην ανάλυση των δεδομένων θα πρέπει να ληφθεί υπόψη και ο παράγοντας του χρόνου ετσι ώστε, η υπηρεσία να είναι σε θέση να παράγει στατιστικές αναφορές στοιχεία από τα ευρύματα των επιθεωρήσεων ανα έτος, μήνα, ημέρα κ.λπ.

Το star schema θα περιλαμβάνει έναν πίνακα για κάθε διάσταση και οι πίνακες αυτοί δημιουργούνται με τις εξής εντολές :

```
CREATE TABLE dim_inspection_type (
    inscode INT PRIMARY KEY,
    instype NVARCHAR(100)
);
```

... ανά τύπο επιθεώρησης

```
CREATE TABLE dim_violation_category (
    vcode INT PRIMARY KEY,
    vdescription NVARCHAR(255),
    vcategory NVARCHAR(255)
);
```

... ανά κατηγορία παράβασης

```
CREATE TABLE dim_restaurant (
    rid INT PRIMARY KEY,
    lat FLOAT,
    lon FLOAT
);
```

... ανά περιοχή, δηλαδή ανά εστιατόριο

```
CREATE TABLE dim_time (
    time_key DATE PRIMARY KEY,
    year INT,
    month INT,
    day INT,
    weekday INT
);
```

... υπόψη ο παράγοντας του χρόνου

Στην συνέχεια έχουμε τον fact table, που περιέχει τα γεγονότα, και έχει ως foreign keys τα κλειδιά των dimension tables.

```
CREATE TABLE inspections_fact (
    rid INT,
    time_key DATE,
    inscode INT,
    vcode INT,
    criticalIssue INT,
    nonCriticalIssue INT,
    PRIMARY KEY (rid, time_key, inscode, vcode),
    FOREIGN KEY (rid) REFERENCES dim_restaurant(rid),
    FOREIGN KEY (time_key) REFERENCES dim_time(time_key),
    FOREIGN KEY (inscode) REFERENCES dim_inspection_type(inscode),
    FOREIGN KEY (vcode) REFERENCES dim_violation_category(vcode)
);
```

Βήμα 3)

Σε αυτό το βήμα θα χρησιμοποιήσουμε τις ακόλουθες εντολές SQL για να τροφοδοτήσουμε την αποθήκη με τα απαραίτητα στοιχεία από τον πίνακα inspections_data :

i) Τροφοδότηση της διάστασης **dim_restaurant** :

```
INSERT INTO dim_restaurant (rid, lat, lon)
SELECT DISTINCT rid, lat, lon
FROM inspections_data;
```

ii) Τροφοδότηση της διάστασης **dim_time** :

```
INSERT INTO dim_time (time_key, year, month, day, weekday)
SELECT DISTINCT insdate, insyear, insmonth, insday, insweekday
FROM inspections_data;
```

iii) Τροφοδότηση της διάστασης **dim_inspections_type** :

```
INSERT INTO dim_inspection_type (inscode, instype)
SELECT DISTINCT inscode, instype
FROM inspections_data;
```

iv) Τροφοδότηση της διάστασης **dim_violation_category** :

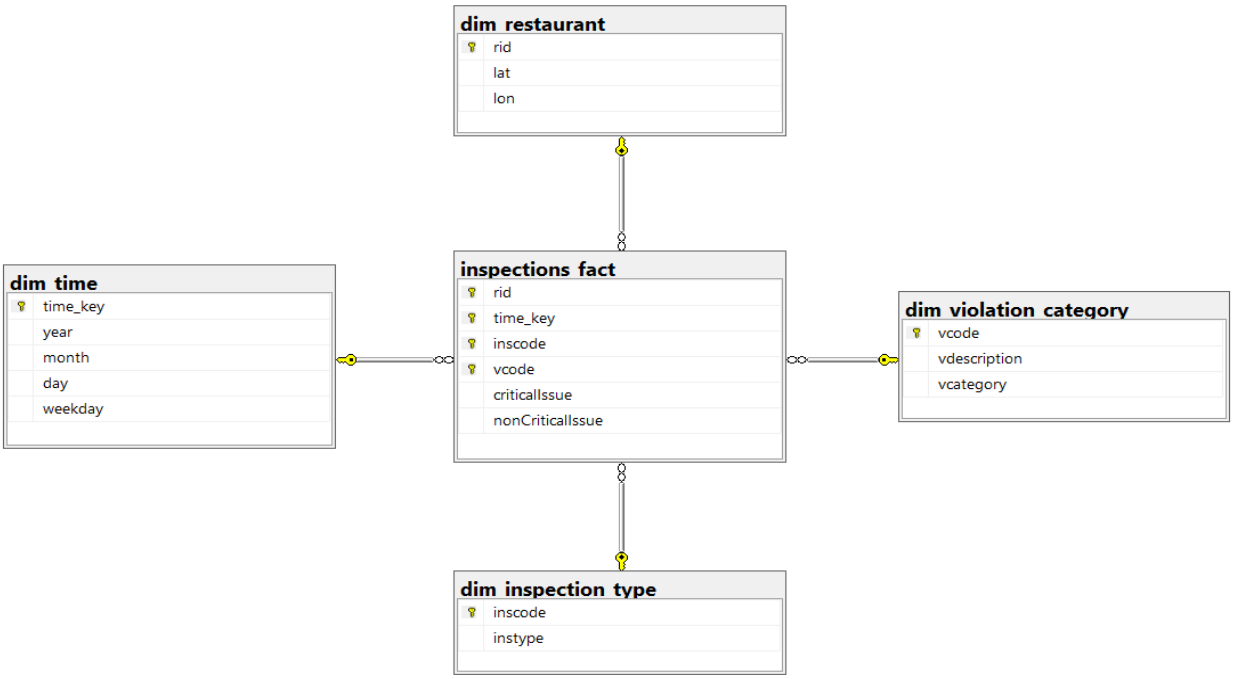
```
INSERT INTO dim_violation_category (vcode, vdescription, vcategory)
SELECT DISTINCT vcode, vdescription, vcategory
FROM inspections_data;
```

v) Τροφοδότηση του πίνακα γεγονότων **inspections_fact** :

```
INSERT INTO inspections_fact (rid, time_key, inscode, vcode, criticalIssue, nonCriticalIssue)
SELECT rid, insdate, inscode, vcode, criticalIssue, nonCriticalIssue
FROM inspections_data;
```

Βήμα 4)

Παρακάτω μπορούμε να δούμε και μία διαγραμματική αναπαράσταση του σχήματος της αποθήκης δεδομένων :



Ζήτημα 2ο

Στο ζήτημα αυτό θα χρησιμοποιήσουμε την αποθήκη δεδομένων που φτιάξαμε στο Ζήτημα 1 για να απαντήσουμε με την μορφή SQL επερωτήσεων σε ερωτήματα σχετικά με την διοίκηση του υπουργείου.

Ερώτημα 1)

Για να απαντήσουμε στο ερώτημα *"Εμφανίστε έναν κατάλογο με τον αριθμό των επιθεωρήσεων ανά έτος και τύπο επιθεώρησης (instype). Ο κατάλογος πρέπει να είναι ταξινομημένος με βάση το έτος σε φθίνουσα διάταξη."* θα χρησιμοποιήσουμε το ακόλουθο επερώτημα στην SQL :

```sql
SELECT t.year, it.instype, COUNT(*) AS inspection_count
FROM inspections_fact f
JOIN dim_time t ON f.time_key = t.time_key
JOIN dim_inspection_type it ON f.inscode = it.inscode
GROUP BY t.year, it.instype
ORDER BY t.year DESC;
```

Το output αυτού του επερωτήματος έχει ακριβώς την μορφή που θέλουμε :

	year	instype	inspection_count
1	2015	Fire on premises	1
2	2015	Restoration	6
3	2015	Preoperational	6
4	2015	License Renewal	3
5	2015	Routine	324
6	2015	Official Call	1
7	2015	Follow-up	124
8	2015	NAME CHANGE	1
9	2015	Remodel	1
10	2015	NEW OWNERSHIP	3
11	2015	Ownership change	4
12	2015	Complaint	66
13	2015	Change of Ownership	2
14	2014	Renovation	1
15	2014	Ownership change	5
16	2014	Sweep	2

Ερώτημα 2)

Για να εμφανίσουμε έναν κατάλογο με τα στοιχεία κάθε εστιατορίου που φαίνονται στην εκφώνηση σε φθίνουσα ταξινομημένη σειρά με βάση το αριθμό των ζητημάτων κρίσιμων και μη, θα χρησιμοποιήσουμε το ακόλουθο επερώτημα στην SQL :

```sql
SELECT r.rid, r.lat, r.lon,
       SUM(f.nonCriticalIssue) AS total_non_critical_issues,
       SUM(f.criticalIssue) AS total_critical_issues,
       SUM(f.nonCriticalIssue + f.criticalIssue) AS total_issues
FROM inspections_fact f
JOIN dim_restaurant r ON f.rid = r.rid
GROUP BY r.rid, r.lat, r.lon
ORDER BY total_issues DESC;
```

Το output αυτού του επερωτήματος έχει ακριβώς την μορφή που θέλουμε :

	rid	lat	lon	total_non_critical_issues	total_critical_issues	total_issues
1	862	38.91425487	-77.01560383	136	95	231
2	4599	38.8859248	-77.0211087	74	68	142
3	4527	38.8323618	-77.0083252	82	58	140
4	4288	38.8767977	-77.00130953	77	59	136
5	3470	38.8968342	-77.0262391	72	61	133
6	976	38.9375716	-76.9907583	75	51	126
7	4603	38.8875921	-76.9984296	73	52	125
8	5580	38.91943313	-77.04187881	79	42	121
9	5167	38.912096	-77.065823	75	45	120
10	3510	38.89651357	-77.03185778	70	48	118
11	1024	38.91994712	-77.00028386	67	46	113
12	47238	38.890969	-77.005925	64	49	113
13	3706	38.8997292	-77.02024024	60	49	109
14	3507	38.896792	-77.050678	63	44	107
15	916	38.8933254	-76.9477203	73	33	106

Ερώτημα 3)

Η επερώτηση SQL που δημιουργεί ένα κύβο, κάθε κελί του οποίου περιέχει τον συνολικό αριθμό των κρίσιμων ζητημάτων που εντοπίστηκαν σε όλες τις επιθεωρήσεις ανά τύπο επιθεώρησης, (instype), κατηγορία παράβασης (vcategory) και έτος επιθεώρησης (year) φαίνεται παρακάτω :

```
SELECT it.instype, v.vcategory, t.year,
    SUM(f.criticalIssue) AS total_critical_issues
FROM inspections_fact f
JOIN dim_inspection_type it ON f.inscode = it.inscode
JOIN dim_violation_category v ON f.vcode = v.vcode
JOIN dim_time t ON f.time_key = t.time_key
GROUP BY
    CUBE(it.instype, v.vcategory, t.year);
```

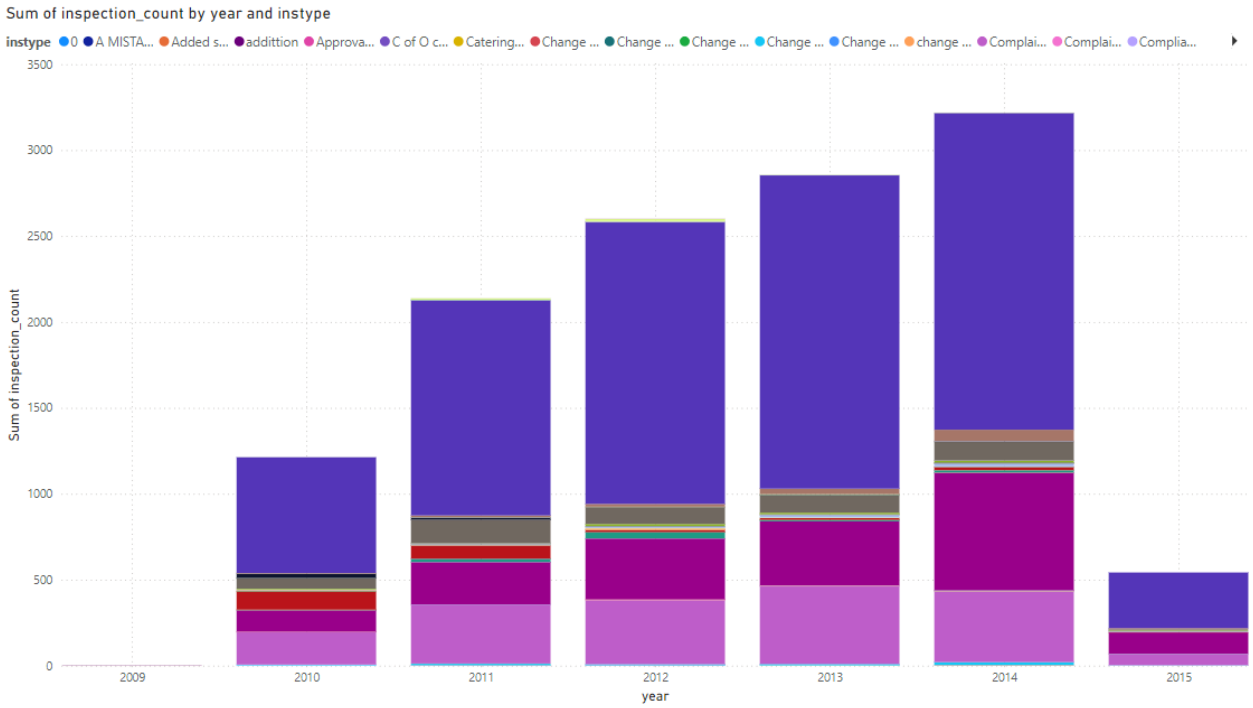
Το output αυτού του επερωτήματος έχει ακριβώς την μορφή που θέλουμε :

	instype	vcategory	year	total_critical_issues
1	License Renewal	Demonstration of knowledge	2009	1
2	NULL	Demonstration of knowledge	2009	1
3	Routine	Utensils. Equipment. and Vending	2009	2
4	NULL	Utensils. Equipment. and Vending	2009	2
5	NULL	NULL	2009	3
6	Complaint	Approved Source	2010	24
7	Follow-up	Approved Source	2010	2
8	Routine	Approved Source	2010	30
9	NULL	Approved Source	2010	56
10	Change of Ownership	Chemical	2010	1
11	Routine	Chemical	2010	2
12	NULL	Chemical	2010	3
13	Follow-up	Conformance with Approved Procedures	2010	2
14	NULL	Conformance with Approved Procedures	2010	2
15	Follow-up	Consumer Advisory	2010	3
16	license application	Consumer Advisory	2010	2

Ζήτημα 3ο

Πρώτη Αναφορά Power BI

Σε αυτό το ζητούμενο θα φτιάξουμε δύο αναφορές Power BI, εκ των οποίων η μία απεικονίζει τα αποτελέσματα του πρώτου επερωτήματος, δηλαδή τον αριθμό επιθεωρήσεων ανά έτος και τύπο επιθεώρησης και φαίνεται παρακάτω :

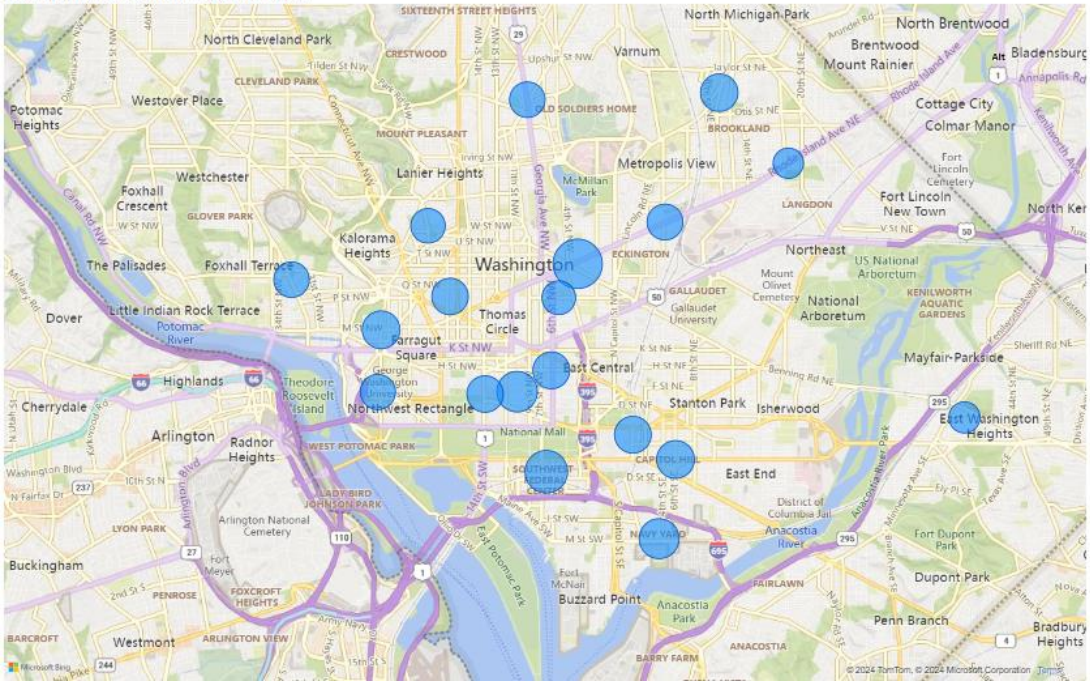


Όπως φαίνεται από το διάγραμμα, στον οριζόντιο άξονα έχουν το έτος της επιθεώρησης και στον κάθετο άξονα το πλήθος των επιθεωρήσεων. Σε κάθε έτος μπορούμε να δούμε το πλήθος επιθεωρήσεων για κάθε τύπο επιθεώρησης, που απεικονίζεται με διαφορετικό χρώμα.

Δεύτερη Αναφορά Power BI

Η δεύτερη αναφορά απεικονίζει τα 20 πρώτα αποτελέσματα του δεύτερου επερωτήματος, δηλαδή έναν κατάλογο 20 εστιατορίων στα οποία προέκυψαν τα περισσότερα ζητήματα από όλες τις επιθεωρήσεις. Πιο συγκεκριμένα, τα καταστήματα αυτά απεικονίζονται σε ένα χάρτη με κουκκίδες, το μέγεθος των οποίων εξαρτάται από τον αριθμό των ζητημάτων που παρατηρήθηκαν. Στα δεξιά του χάρτη, φαίνεται ένας πίνακας ο οποίος για κάθε εστιατόριο δείχνει την τις συντεταγμένες του, δηλαδή το latitude και το longitude. Μπορούμε να αλληλεπιδράσουμε με τον χάρτη πατώντας πάνω σε κάποια κουκκίδα, ή ακόμα και πατώντας πάνω σε κάποιο εστιατόριο του πίνακα ώστε να δούμε σε ποια κουκκίδα αντιστοιχεί.

Sum of total_critical_issues by lat and lon



rid	lon	lat
147691	-77.04	38.91
47238	-77.01	38.89
5580	-77.04	38.92
5167	-77.07	38.91
4692	-77.02	38.94
4603	-77.00	38.89
4599	-77.02	38.89
4527	-77.01	38.83
4288	-77.00	38.88
3706	-77.02	38.90
3510	-77.03	38.90
3507	-77.05	38.90
3470	-77.03	38.90
2973	-77.05	38.91
1826	-77.02	38.91
1024	-77.00	38.92
976	-76.99	38.94
916	-76.95	38.89
862	-77.02	38.91
757	-76.98	38.93