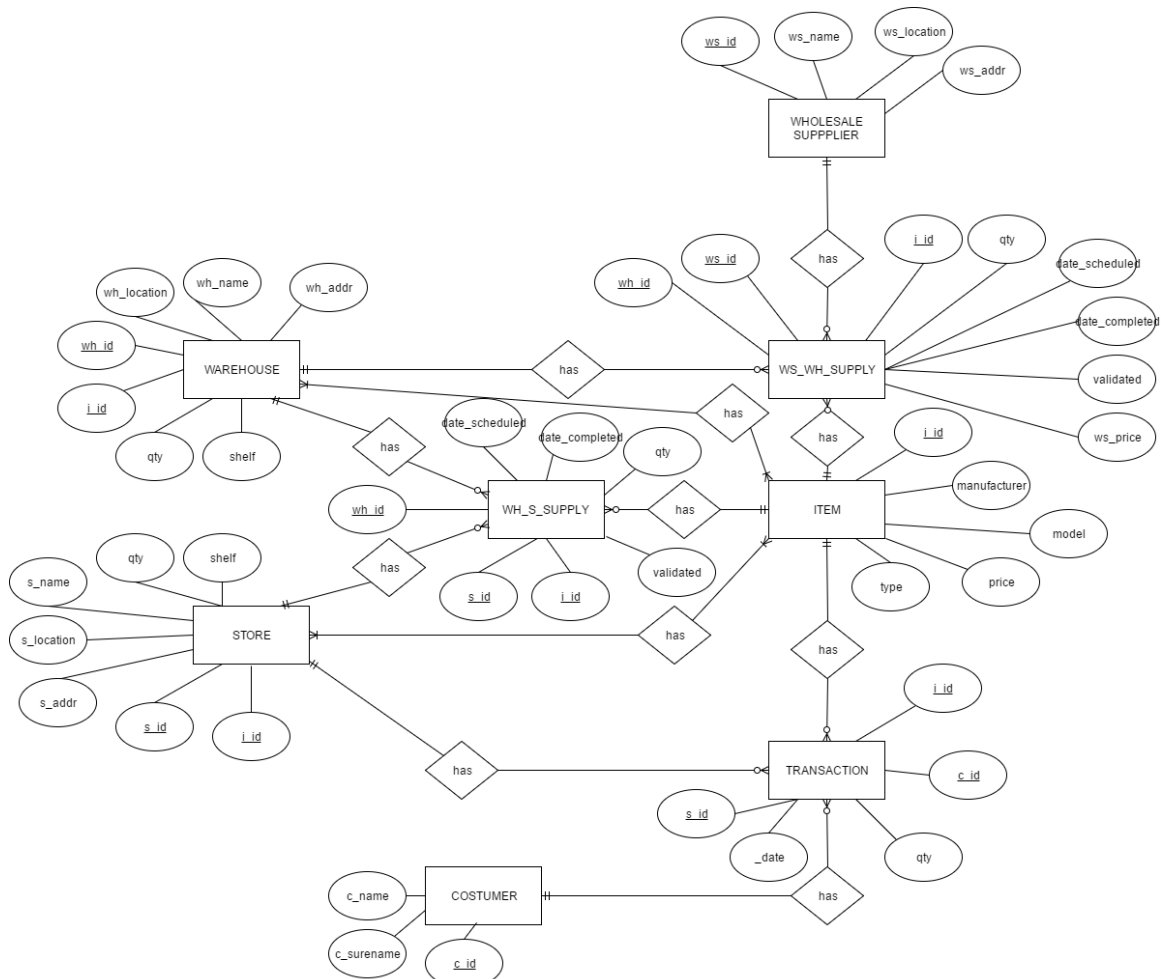


# <<Σχεδιασμός ΒΔ αποθήκης καταστήματος>>

Ορέστης Νικόλας, ΑΜ:228438

## -Παραδοχές & περιγραφή του μικρόκοσμου.

Παρακάτω φαίνεται το Entity - Relation μοντέλο που υλοποίησα για το μικρόκοσμο αυτό.

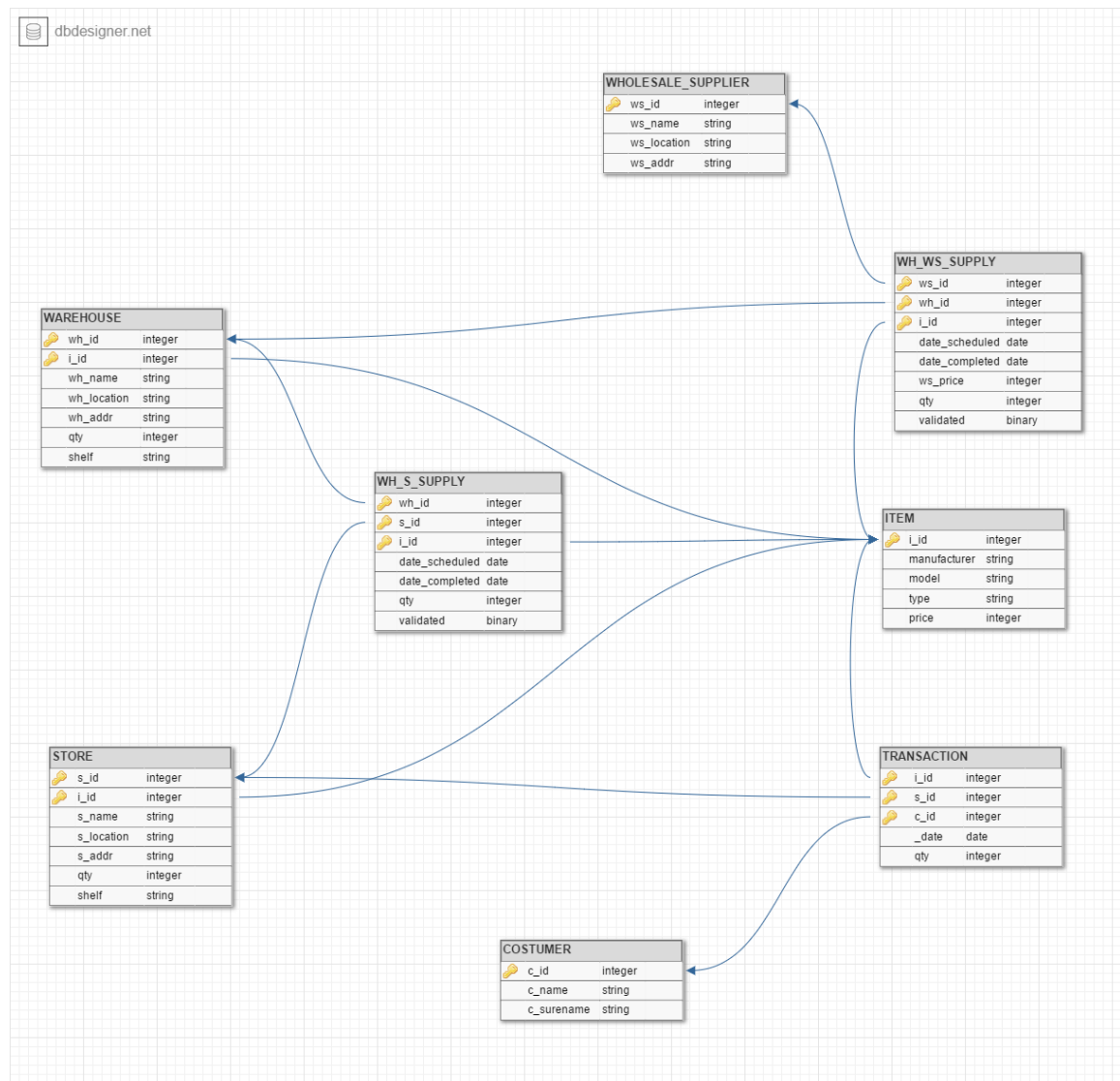


Η ΙΔΕΑ ήταν να κάνω μια βάση δεδομένων για τις αποθήκες μίας σχετικά μεγάλης αλυσίδας καταστημάτων όπως π.χ. το πλαίσιο, την οποία αλυσίδα την ενδιαφέρει να ξέρει τι αποθέματα έχει σε κάθε κατάστημα της, τι αποθέματα έχουν οι ίδιες οι αποθήκες ώστε να ξέρει πότε χρειάζονται να ανανεώσουν τα ελλειπή αποθέματα φέρνοντας καινούρια από το δίκτυο πωλητών Ολικής (Wholesale Supplier). Προφανώς σε μία τέτοια βάση δεδομένων μας ενδιαφέρουν επίσης σε πολύ μεγάλο βαθμό οι συναλλαγές που γίνονται σε όλα τα επίπεδα.

Οι ΠΑΡΑΔΟΧΕΣ που έχω κάνει είναι οι εξής:

1. Για τους wholesale supplier δεν μας ενδιαφέρει το τι αποθέματα έχει και από ποια προϊόντα και φαντάζομαι ίσως να μην το γνωρίζουμε αυτό σε ένα πραγματικό πρόβλημα, το μόνο που μας ενδιαφέρει είναι η συναλλαγή που θα γίνει (ws\_wh\_supply) και τα διάφορα πεδία της όπως date\_scheduled, qty.
2. Το entity ITEM είναι καθαρά η οντότητα του ίδιου του προϊόντος, δηλαδή το id του και διάφορα στοιχεία τα οποία μπορεί να ενδιαφέρουν. Το price που έχω γενικότερα το έβαλα να είναι κάτι σαν την ενδεικτική τιμή που δίνει ο κατασκευαστής και υποθέτω ότι στο transaction, που είναι πρακτικά ο πελάτης να παίρνει κάποιο ITEM από ένα κατάστημα, η τιμή per-item που θα χρησιμοποιηθεί θα είναι η ενδεικτική. Επίσης όσον αφορά το ws\_wh\_supply έχω βάλει και το ws\_price ώστε να ξέρουμε πόσο φθηνότερα τα πουλάει ο wholesale supplier συγκριτικά με την ενδεικτική.
3. Όσον αφορά τις συναλλαγές (ws\_wh\_supply/wh\_s\_supply/transaction), ακολουθούν την κλασική 1 προς 1 προς 1 συναλλαγή, π.χ. στο transaction ο customer με c\_id=5 αγοράζει το item με i\_id=10 από το store με s\_id = 4 με qty = 2. Κάτι παρόμοιο είναι και στο ws\_wh\_supply και στο wh\_s\_supply με τις διαφορές ότι εκεί έχω και date\_scheduled, date\_completed και το verified. Το verified είναι αρκετά σημαντικό γιατί αργότερα στον κώδικα python η εκπλήρωση και η ανανέωση των πινάκων για τη συναλλαγή εξαρτώνται από αυτό. Επίσης υποθέτω ότι στο transaction με το που μπει κάποια καινούρια εγγραφή τότε αυτόματα ανανεώνονται τα αποθέματα. Επιπλέον θεωρώ ότι όλες οι συναλλαγές γίνονται up-front.
4. Για τα Store/Warehouse ακολουθούν τη λογική ότι το wh\_id=1 έχει το i\_id=5 σε qty = 40. Έχω διαχωρίσει το location και το addr θεωρώντας ότι το location είναι η πόλη που βρίσκεται το warehouse/store/wholesale\_supplier ενώ το addr είναι κανονικά η διεύθυνση. Σημαντικό είναι ότι ούτε όλα τα warehouse ούτε τα store έχουν όλα τα items, θα το αναλύσω αργότερα αυτό στον αλγόριθμο **Random\_Data\_Generator** που έφτιαξα για να φτιάξω ένα instance της βάσης δεδομένων ώστε να μπορώ να τη δοκιμάσω.
5. Όσον αφορά την επεκτασιμότητα της βάσης, δηλαδή σε περίπτωση που θέλω να βάλω στην αγορά έχω νέο item τότε το εκχωρώ στο item πρώτα και μετά συνεχίζω κανονικά, πράττω παρόμοια αν υπάρχει καινούριο warehouse/store/wholesale\_supplier/customer.

Τώρα πάμε στο SCHEMA:



Εδώ πρακτικά είναι η υλοποίηση του ER-D σε πίνακες, το μόνο που έχω να πω για εδώ είναι ότι αργότερα στο **StorageMySQLCreate** βάζω default 0 σε όλα τα verified των συναλλαγών γιατί θεωρώ ότι είναι κάτι το οποίο κάποιος υπάλληλος θα πρέπει να είναι υπεύθυνος και να τα επιβλέπει. Επίσης δεν αφήνω πουθενά NULL εκτός από τα date\_scheduled, date\_complete και shelf σε περίπτωση που είναι άγνωστα κατά την εγγραφή. Στο \_date του transaction υποθέτω ότι θα μπαίνει αυτόματα η (τότε) μέρα που γίνεται η εγγραφή.

## -Random Data Generator Algorithm και το παρόν instance της βάσης.

Δεν νομίζω ότι έχει νόημα να εισέλθω σε λεπτομέρειες για το πώς έκανα randomize τα δεδομένα. Όλος ο κώδικας υπάρχει στο αρχείο `Random_Data_Generator.py` απλά θα αναφέρω μερικά σημεία και θα δείξω με εικόνες τα αποτελέσματα.

1. Επειδή είχα την παραδοχή ότι δεν έχουν όλα τα store/warehouse όλα τα items έχω φροντίσει να κάνω πρώτα randomize όλους τους πίνακες πέρα των πινάκων που έχουν να κάνουν με συναλλαγές ώστε μετά εκεί να μπορώ να δώ τι items έχει ο καθένας ώστε τα transaction να είναι σωστά (και υλοποιησιμα). Για παράδειγμα το κομμάτι του κώδικα που χρησιμοποίησα για να κάνω randomize τον πίνακα `wh_s_supplies` φαίνεται παρακάτω:

```
# _____ wh_s_supply_initialisation _____

#Checking which stores have what items

select_stmt = ("SELECT i_id FROM store WHERE s_id=%s")
s_items=[]
for storenum in range(11):
    items=[]
    count=0
    cur.execute(select_stmt, (str(storenum)))
    desc = [x[0] for x in cur.description]
    for row in cur.fetchall():
        for i in row :
            items.append(i)
        count +=1
    s_items.append(items) #s_items[1] = [1,2,3,5,6,...29,30]
#Checking which warehouses have what items

select_stmt = ("SELECT i_id FROM warehouse WHERE wh_id=%s")
wh_items=[]
for whnum in range(6):
    items=[]
    count=0
    cur.execute(select_stmt, (str(whnum)))
    desc = [x[0] for x in cur.description]
    #print(desc,"\t")
    for row in cur.fetchall():
        for i in row :
            items.append(i)
        count +=1
    wh_items.append(items) #wh_items[1] = [1,2,3,5,6,...29,30]

#Inserting random edible transactions

insert_stmt = (
    "INSERT INTO wh_s_supply (wh_id, s_id, i_id, qty, date_scheduled, date_completed) "
    "VALUES (%s, %s, %s, %s, %s, %s)"
)

for i in range(90):

    #making sure that the warehouse AND the store indeed has the randomised item
    wh_id = randint(1,5)
    s_id = randint (1,10)
    i_id = randint(1,30)
    while (i_id not in wh_items[wh_id]) and (i_id not in s_items[s_id]):
        i_id = randint(1,30)

    #making sure that date_completed>=date_scheduled
    date_scheduled = datetime.date(randint(2000,2017),randint(1,12),randint(1,25))
    date_completed = datetime.date(randint(2000,2017),randint(1,12),randint(1,25))
    while date_completed<date_scheduled:
        date_completed = datetime.date(randint(2000,2017),randint(1,12),randint(1,25))

    qty = randint(10,100)

    data = (wh_id, s_id, i_id, qty, date_scheduled, date_completed)
    cur.execute(insert_stmt,data)
    cur.execute('commit')
```

1

2. Τα περισσότερα στοιχεία είναι randomised πάνω σε κάποιο συγκεκριμένο πλαίσιο, ακόμα και τα dates.
3. Μετά την αρχικοποίηση έκανα όλες τις συναλλαγές validated εκτός μερικών και θεώρησα ότι αυτή είναι η κατάσταση της βάσης και δουλεψα από κεί και πέρα.

Τώρα θα αναθέσω μερικά print screens για το πώς είναι περίπου τα στοιχεία στη βάση δεδομένων μετά την αρχικοποίηση:

### COSTUMER

✓ Εμφάνιση εγγραφών 0 - 24 (1000 συνολικά, Το ερώτημα χρειάστηκε 0.0006 δευτερ

`SELECT * FROM `costumer``

1 > >> | Αριθμός εγγραφών: 25 Φιλτράρισμα εγγρ

+ Επιλογές

			c_id	c_name	c_surname
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	1	c_name1 c_surname1
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	2	c_name2 c_surname2
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	3	c_name3 c_surname3

### ITEM

✓ Εμφάνιση εγγραφών 0 - 24 (30 συνολικά, Το ερώτημα χρειάστηκε 0.0009 δευτερόλεπτα.)

`SELECT * FROM `item``

1 > >> | ☐ Εμφάνιση όλων | Αριθμός εγγραφών: 25 Φιλτράρισμα εγγ

+ Επιλογές

				i_id	manufacturer	model	type	price
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	1	SAMSUNG	Smodel1	Mobile	225
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	2	SAMSUNG	Smodel2	Mobile	268
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	3	SAMSUNG	Smodel3	Laptop	137
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	4	SAMSUNG	Smodel4	Mobile	88
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	5	SAMSUNG	Smodel5	Desktop	283
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	6	SAMSUNG	Smodel6	Laptop	91
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	7	SAMSUNG	Smodel7	Mobile	431
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	8	SAMSUNG	Smodel8	Desktop	240
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	9	SAMSUNG	Smodel9	Laptop	100
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	10	SAMSUNG	Smodel10	Mobile	334
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	11	DELL	Dmodel1	Mobile	496
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	12	DELL	Dmodel2	Desktop	377

### WHOLESALE\_SUPPLIER

Εμφάνιση εγγραφών 0 - 9 (10 συνολικά, Το ερώτημα χρειάστηκε 0.0007 δευτερόλεπτα.)

SELECT \* FROM `wholesale\_supplier`

☐ Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτηση

+ Επιλογές

			ws_id	ws_name	ws_location	ws_addr
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
		1	ws_name1	ws_location1	ws_addr1	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
		2	ws_name2	ws_location2	ws_addr2	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
		3	ws_name3	ws_location3	ws_addr3	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή
		4	ws_name4	ws_location4	ws_addr4	

WAREHOUSE

Εμφάνιση εγγραφών 0 - 24 (114 συνολικά, Το ερώτημα χρειάστηκε 0.0008 δευτερόλεπτα.)

SELECT \* FROM `warehouse`

1 > >>

☐ Εμφάνιση όλων | Αριθμός εγγραφών: 25 | Φιλτράρισμα εγγραφών: Αναζήτηση

+ Επιλογές

			wh_id	i_id	wh_name	wh_location	wh_addr	qty	shelf
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	2	wh_name1	wh_location1	wh_addr1	50	Shelf19	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	3	wh_name1	wh_location1	wh_addr1	65	Shelf17	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	4	wh_name1	wh_location1	wh_addr1	190	Shelf17	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	5	wh_name1	wh_location1	wh_addr1	187	Shelf23	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	6	wh_name1	wh_location1	wh_addr1	182	Shelf14	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	7	wh_name1	wh_location1	wh_addr1	187	Shelf19	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	9	wh_name1	wh_location1	wh_addr1	129	Shelf18	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	10	wh_name1	wh_location1	wh_addr1	124	Shelf30	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	11	wh_name1	wh_location1	wh_addr1	69	Shelf16	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	13	wh_name1	wh_location1	wh_addr1	186	Shelf26	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	16	wh_name1	wh_location1	wh_addr1	83	Shelf17	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	17	wh_name1	wh_location1	wh_addr1	76	Shelf2	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	19	wh_name1	wh_location1	wh_addr1	161	Shelf20	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	21	wh_name1	wh_location1	wh_addr1	191	Shelf24	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	23	wh_name1	wh_location1	wh_addr1	170	Shelf25	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	24	wh_name1	wh_location1	wh_addr1	6	Shelf20	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	25	wh_name1	wh_location1	wh_addr1	14	Shelf26	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	26	wh_name1	wh_location1	wh_addr1	198	Shelf6	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	27	wh_name1	wh_location1	wh_addr1	77	Shelf8	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	28	wh_name1	wh_location1	wh_addr1	111	Shelf5	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	29	wh_name1	wh_location1	wh_addr1	103	Shelf8	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		1	30	wh_name1	wh_location1	wh_addr1	4	Shelf13	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		2	1	wh_name2	wh_location2	wh_addr2	91	Shelf30	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		2	2	wh_name2	wh_location2	wh_addr2	33	Shelf20	
<input type="checkbox"/>		Επεξεργασία		Αντιγραφή		Διαγραφή			
		2	3	wh_name2	wh_location2	wh_addr2	77	Shelf18	

STORE

✓ Εμφάνιση εγγραφών 25 - 49 (246 συνολικά, Το ερώτημα χρειάστηκε 0.0009 δευτερόλεπτα.)

`SELECT * FROM `store``

<< < 2 > >> | ☐ Εμφάνιση όλων | Αριθμός εγγραφών: 25 ▾ Φιλτράρισμα εγγραφών

+ Επιλογές

			s_id	i_id	s_name	s_location	s_addr	qty	shelf
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	1	29	s_name1	s_location1	s_addr1	36 Shelf7
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	1	30	s_name1	s_location1	s_addr1	80 Shelf16
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	2	1	s_name2	s_location2	s_addr2	65 Shelf21
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	2	2	s_name2	s_location2	s_addr2	45 Shelf26
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	2	3	s_name2	s_location2	s_addr2	51 Shelf29
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	2	4	s_name2	s_location2	s_addr2	69 Shelf21
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	2	5	s_name2	s_location2	s_addr2	83 Shelf24
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	2	6	s_name2	s_location2	s_addr2	20 Shelf12

## TRANSACTION

✓ Εμφάνιση εγγραφών 0 - 24 (500 συνολικά, Το ερώτημα χρειάστηκε 0.0014 δευτερόλεπτα.)

`SELECT * FROM `transaction``

1 ▾ > >> | ☐ Εμφάνιση όλων | Αριθμός εγγραφών: 25 ▾ Φιλτ

+ Επιλογές

				c_id	s_id	i_id	qty	date_
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	5	1	13	3	2007-09-16
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	5	1	20	5	2017-07-05
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	7	7	23	3	2008-09-07
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	9	8	21	3	2015-06-18
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	13	3	26	5	2002-06-05
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	14	10	26	5	2016-05-22
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	17	7	8	5	2005-09-08
<input type="checkbox"/>	✎ Επεξεργασία	📄 Αντιγραφή	🚫 Διαγραφή	22	2	12	3	2009-02-07

## WS\_WH\_SUPPLY

✓ Εμφάνιση εγγραφών 0 - 24 (42 συνολικά, Το ερώτημα χρειάστηκε 0.0011 δευτερόλεπτα.)

SELECT \* FROM 'ws\_wh\_supply'

1 > >> | ☐ Εμφάνιση όλων | Αριθμός εγγραφών: 25 Φιλτράρισμα εγγραφών: Αναζήτηση σε αυτόν τον πίνακα Ταξινό

+ Επιλογές

<

## WH\_S\_SUPPLY

✓ Εμφάνιση εγγραφών 75 - 93 (94 συνολικά, Το ερώτημα χρειάστηκε 0.0014 δευτερόλεπτα.)

`SELECT * FROM 'wh_s_supply'`

<< 4 > >> | ☐ Εμφάνιση όλων | Αριθμός εγγραφών: 25 Φιλτράρισμα εγγραφών: Αναζήτηση σε αυτόν τον πίνακα

+ Επιλογές

				wh_id	s_id	i_id	date_scheduled	date_completed	qty	validated
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	4	10	17	2010-01-23	2013-05-02	11	1
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	4	10	19	2002-09-15	2010-05-12	34	1
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	4	10	20	2007-07-12	2009-11-07	50	1
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	5	1	13	2001-04-03	2002-03-09	68	1
<input type="checkbox"/>	Επεξεργασία	Αντιγραφή	Διαγραφή	5	1	20	2000-01-17	2009-11-18	74	1

## -SQL queries and testing.

Εδώ έχω μερικά queries απευθείας σε SQL και τα αποτελέσματα:

1. Checking which stores the client with id=24 had had transactions and some details:

```
SELECT DISTINCT store.s_name, transaction.i_id, transaction.qty, transaction.date_
```



```
FROM transaction JOIN store ON transaction.s_id=store.s_id
WHERE transaction.c_id=24
```

s_name	i_id	qty	date_
s_name4	18	1	2002-05-04
s_name7	25	4	2016-12-21
s_name8	29	2	2017-05-20

2. Checking shortages of items on stores

```
SELECT s_id,i_id,qty,s_name FROM store WHERE qty<15
```

	s_id	i_id	qty	s_name
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	1	21	10	s_name1
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	3	7	11	s_name3
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	3	10	10	s_name3
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	3	23	11	s_name3
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	4	3	10	s_name4
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	4	10	11	s_name4
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	5	25	12	s_name5
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	5	28	13	s_name5
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	7	24	11	s_name7
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	8	3	14	s_name8
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	8	14	12	s_name8
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	8	15	10	s_name8
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	10	29	14	s_name10

3. Checking which wh\_s\_supplies transactions had been scheduled for 2016 and have been validated:

```
SELECT *
FROM wh_s_supply
WHERE wh_s_supply.date_scheduled LIKE '2016%' AND validated = 1
```

	wh_id	s_id	i_id	date_scheduled	date_completed	qty	validated
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	2	7	26	2016-10-08	2017-09-14	21	1
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	3	2	4	2016-03-23	2016-10-20	43	1
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	3	3	12	2016-04-24	2017-01-11	48	1
<input type="checkbox"/> Επεξεργασία Αντιγραφή Διαγραφή	5	8	24	2016-12-02	2016-12-07	89	1

4. Checking which warehouses have been supplied by wholesale\_supplier with id=1

```
SELECT DISTINCT wh_id
FROM ws_wh_supply
WHERE ws_id = 1
```

+ Επιλογές

wh\_id

1

2

3

5

5. Checking total quantities of items that were given to our warehouses by all wholesale suppliers

```
SELECT ws_id, SUM(qty)
```

```
FROM ws_wh_supply
```

```
GROUP BY ws_id
```

+ Επιλογές

ws_id	SUM(qty)
-------	----------

1	942
---	-----

2	1168
---	------

3	1286
---	------

4	1649
---	------

5	1197
---	------

## -Application.

Το application που έχω υλοποιήσει είναι σε python και έχω βάλει κάποιες λειτουργίες που πιστεύω θα ήταν χρήσιμες σε μια πραγματική εφαρμογή. Όλος ο κώδικας βρίσκεται στο αρχείο **Appliaction.py**

Το menu έχει ως εξής:

```
connected to:storagedb database
Choose action
1:Insert SQL commands
2:Check Tables
3:Transaction Validation Mode
4:Check for store item shortages
Press anything else to exit
>>>|
```

1. Η πρώτη επιλογή είναι προφανές το τι κάνει. Μερικά test που έκανα ήταν να βάλω τις ίδιες εντολές που έκανα στο SQL testing για να δώ πώς θα εμφανίζει τα αποτελέσματα.

```
SELECT *
```

```
FROM wh_s_supply
```

```
WHERE wh_s_supply.date_scheduled LIKE '2016%' AND validated = 1
```

```

connected to:storedb database
Choose action
1:Insert SQL commands
2:Check Tables
3:Transaction Validation Mode
4:Check for store item shortages
Press anything else to exit
>>>1
Bd: storedb Give SQL commands:
>>>SELECT * FROM wh_s_supply WHERE wh_s_supply.date_scheduled LIKE '2016%' AND validated = 1

SELECT * FROM wh_s_supply WHERE wh_s_supply.date_scheduled LIKE '2016%' AND validated = 1
wh_id      s_id      i_id      date_scheduled      date_completed      qty      validated
2          7         26        2016-10-08          2017-09-14          21       b'1'
3          2         4         2016-03-23          2016-10-20          43       b'1'
3          3         12        2016-04-24          2017-01-11          48       b'1'
5          8         24        2016-12-02          2016-12-07          89       b'1'
συνολα : 4

```

```

SELECT DISTINCT store.s_name,transaction.i_id, transaction.qty,
transaction.date_

FROM transaction JOIN store ON transaction.s_id=store.s_id

WHERE transaction.c_id=24

```

```

Choose action
1:Insert SQL commands
2:Check Tables
3:Transaction Validation Mode
4:Check for store item shortages
Press anything else to exit
>>>1
Bd: storedb Give SQL commands:
>>>SELECT DISTINCT store.s_name,transaction.i_id, transaction.qty, transaction.date_ FROM transaction JOIN store ON transaction.s_id=store.s_id WHERE transaction.c_id=24

SELECT DISTINCT store.s_name,transaction.i_id, transaction.qty, transaction.date_ FROM transaction JOIN store ON transaction.s_id=store.s_id WHERE transaction.c_id=24
s_name      i_id      qty      date_
s_name4     18        1        2002-05-04
s_name7     25        4        2016-12-21
s_name8     29        2        2017-05-20
συνολα : 3

```

## 2. Η δεύτερη επιλογή είναι επίσης προφανής και απλά κάνει SELECT \* στον επιλεγμένο πίνακα

```

Choose action
1:Insert SQL commands
2:Check Tables
3:Transaction Validation Mode
4:Check for store item shortages
Press anything else to exit
>>>2
Available table names are:
0 costumer
1 item
2 store
3 transaction
4 warehouse
5 wholesale_supplier
6 wh_s_supply
7 ws_wh_supply
Select which of the above tables you'd like to inspect
>>>2

s_id      i_id      s_name      s_location      s_addr      qty      shelf
1         1         s_name1     s_location1     s_addr1     71       Shelf18
1         2         s_name1     s_location1     s_addr1     139      Shelf24
1         3         s_name1     s_location1     s_addr1     74       Shelf21
1         5         s_name1     s_location1     s_addr1     19       Shelf26
1         6         s_name1     s_location1     s_addr1     85       Shelf27
1         7         s_name1     s_location1     s_addr1     58       Shelf1
1         8         s_name1     s_location1     s_addr1     23       Shelf22
1         10        s_name1     s_location1     s_addr1     87       Shelf9
1         11        s_name1     s_location1     s_addr1     55       Shelf3
1         12        s_name1     s_location1     s_addr1     64       Shelf8
1         13        s_name1     s_location1     s_addr1     90       Shelf22
1         14        s_name1     s_location1     s_addr1     83       Shelf19
1         15        s_name1     s_location1     s_addr1     35       Shelf30
1         16        s_name1     s_location1     s_addr1     44       Shelf4
1         17        s_name1     s_location1     s_addr1     67       Shelf24
1         18        s_name1     s_location1     s_addr1     22       Shelf2
1         19        s_name1     s_location1     s_addr1     36       Shelf13
1         20        s_name1     s_location1     s_addr1     73       Shelf14
1         21        s_name1     s_location1     s_addr1     10       Shelf28
1         22        s_name1     s_location1     s_addr1     61       Shelf19
1         23        s_name1     s_location1     s_addr1     71       Shelf24
1         24        s_name1     s_location1     s_addr1     17       Shelf6
1         25        s_name1     s_location1     s_addr1     76       Shelf28
1         26        s_name1     s_location1     s_addr1     31       Shelf10
1         27        s_name1     s_location1     s_addr1     19       Shelf3
1         29        s_name1     s_location1     s_addr1     36       Shelf7
1         30        s_name1     s_location1     s_addr1     80       Shelf16
2         1         s_name2     s_location2     s_addr2     65       Shelf21
2         2         s_name2     s_location2     s_addr2     45       Shelf26
2         3         s_name2     s_location2     s_addr2     51       Shelf29
....more ? (y/n)v

```

## 3. Τώρα η Τρίτη επιλογή πιστεύω είναι η πιο ενδιαφέρουσα. Αυτό που κάνει είναι να κοιτάζει τους 2 πίνακες ws\_wh\_surply και wh\_s\_supply και να βλέπει ποια transactions έχουν γίνει(Δηλαδή το date\_completed δεν είναι null) και δεν είναι validated και να ζητάει από τον χρήστη αν θέλει να τα κάνει validate. Κάνοντας τα

validate το πρόγραμμα ανανεώνει τους σχετικούς πίνακες ώστε να έχουν τα κατάλληλα quantities όπως θα γινόταν μετά από μία συναλλαγή. Επίσης το πρόγραμμα κοιτάζει αν είναι εφικτή η συναλλαγή, π.χ. σε περίπτωση όπου ζητείται συναλλαγή με το warehouse για κάποιο item που έχει αλλά η ποσότητα που ζητείται δεν υπάρχει τότε δεν γίνεται η συναλλαγή.

Για παράδειγμα, πριν τη μεταφορά έχουμε:

Wh\_s\_supply tuple

	wh_id	s_id	i_id	date_scheduled	date_completed	qty	validated
<input type="checkbox"/> Επεξεργασία  Αντιγραφή  Διαγραφή	1	1	2	2011-01-01	2011-02-03	60	0

Warehouse tuple

	wh_id	i_id	wh_name	wh_location	wh_addr	qty	shelf
<input type="checkbox"/> Επεξεργασία  Αντιγραφή  Διαγραφή	1	2	wh_name1	wh_location1	wh_addr1	50	Shelf9

Store tuple

+ Επιλογές

	s_id	i_id	s_name	s_location	s_addr	qty	shelf
<input type="checkbox"/> Επεξεργασία  Αντιγραφή  Διαγραφή	1	1	s_name1	s_location1	s_addr1	71	Shelf18
<input type="checkbox"/> Επεξεργασία  Αντιγραφή  Διαγραφή	1	2	s_name1	s_location1	s_addr1	139	Shelf24

Choose action

```
1:Insert SQL commands
2:Check Tables
3:Transaction Validation Mode
4:Check for store item shortages
Press anything else to exit
>>>3
```

```
--Pending validation on wholesale-warehouse transaction:
```

```
(ws_id,wh_id,i_id,date_completed): 1 1 3 2011-02-03
```

```
You want to validate the above transaction? y/n:n
```

```
--Pending validation on warehouse-store transaction:
```

```
(wh_id,s_id,i_id,date_completed): 1 1 2 2011-02-03
```

```
You want to validate the above transaction? y/n:y
```

```
--This transaction cant be done because warehouse 1 has 50 of item 2 and the transaction requires 60
```

Εδώ βλέπουμε ότι το warehouse με wh\_id =1 δεν έχει το i\_id=2 στο quantity που ζητάει το transaction, οπότε δεν κάνει τη συναλλαγή.

Τώρα ένα άλλο παράδειγμα:

Choose action

```
1:Insert SQL commands
2:Check Tables
3:Transaction Validation Mode
4:Check for store item shortages
Press anything else to exit
>>>3
```

```
--Pending validation on wholesale-warehouse transaction:
```

```
(ws_id,wh_id,i_id,date_completed): 1 1 3 2011-02-03
```


```
You want to validate the above transaction? y/n:|
```

Πριν το validation:

WAREHOUSE

	wh_id	i_id	wh_name	wh_location	wh_addr	qty	shelf
<input type="checkbox"/> Επεξεργασία  Αντιγραφή  Διαγραφή	1	2	wh_name1	wh_location1	wh_addr1	50	Shelf9
<input type="checkbox"/> Επεξεργασία  Αντιγραφή  Διαγραφή	1	3	wh_name1	wh_location1	wh_addr1	65	Shelf17

Μετά το validation:

+ Επιλογές		←T→		wh_id	i_id	wh_name	wh_location	wh_addr	qty	shelf
<input type="checkbox"/>	 Επεξεργασία	 Αντιγραφή	 Διαγραφή	1	2	wh_name1	wh_location1	wh_addr1	50	Shelf9
<input type="checkbox"/>	 Επεξεργασία	 Αντιγραφή	 Διαγραφή	1	3	wh_name1	wh_location1	wh_addr1	75	Shelf17

Βλέπουμε ότι το warehouse 1 στο item 3 πήρε + 10 στο qty, σύμφωνα με τη συγκεκριμένη συναλλαγή.

4. Η τελευταία επιλογή κοιτάζει τα store και δείχνει που υπάρχουν ελλείψεις(Δηλαδή < 20 qty) και τα βγάζει με τον εξής τρόπο:

```
Choose action
1:Insert SQL commands
2:Check Tables
3:Transaction Validation Mode
4:Check for store item shortages
Press anything else to exit
>>>4
store:1 has shortage of item:5(19 left)
store:1 has shortage of item:21(10 left)
store:1 has shortage of item:24(17 left)
store:1 has shortage of item:27(19 left)
store:2 has shortage of item:14(19 left)
store:2 has shortage of item:18(16 left)
store:3 has shortage of item:7(11 left)
store:3 has shortage of item:10(10 left)
store:3 has shortage of item:11(17 left)
store:3 has shortage of item:23(11 left)
store:3 has shortage of item:30(17 left)
store:4 has shortage of item:3(10 left)
store:4 has shortage of item:10(11 left)
store:5 has shortage of item:12(16 left)
store:5 has shortage of item:25(12 left)
store:5 has shortage of item:28(13 left)
store:6 has shortage of item:15(17 left)
store:6 has shortage of item:20(15 left)
store:7 has shortage of item:2(18 left)
store:7 has shortage of item:24(11 left)
store:8 has shortage of item:3(14 left)
store:8 has shortage of item:5(19 left)
store:8 has shortage of item:11(19 left)
store:8 has shortage of item:14(12 left)
store:8 has shortage of item:15(10 left)
store:8 has shortage of item:28(15 left)
store:9 has shortage of item:19(17 left)
store:9 has shortage of item:25(15 left)
store:10 has shortage of item:1(18 left)
store:10 has shortage of item:29(14 left)
```

ΤΕΛΟΣ

...Ευχαριστώ