

# ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΕΣΤΙΑΤΟΡΙΟΥ

Εργασία της ομάδας 12 στα πλαίσια του μαθήματος “Βάσεις Δεδομένων” ακαδημαϊκό έτος 2022-2023

Ζωιτάκης Γεώργιος

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών, Πάτρα, Ελλάδα,  
[up1066998@upnet.gr](mailto:up1066998@upnet.gr)

Φέρτης Ιωάννης

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών, Πάτρα, Ελλάδα,  
[up1072656@upnet.gr](mailto:up1072656@upnet.gr)

## 1 ΕΙΣΑΓΩΓΗ

### 1.1 Εισαγωγικό Σημείωμα

Η εργασία που μας ανατέθηκε έχει ως θέμα την σχεδίαση μιας βάσης δεδομένων για μια εφαρμογή υποστήριξης ενός εστιατορίου. Για την δημιουργία της βάσης μας χρησιμοποιήσαμε το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων της SQLite και η ανάπτυξη των συναρτήσεων και της εφαρμογής μέσω των οποίων επικοινωνούμε με την βάση μας έγινε χρησιμοποιώντας τη γλώσσα προγραμματισμού Python. Το μοντέλο οντοτήτων συσχετίσεων (ERD) της βάσης σχεδιάστηκε με το εργαλείο [erd-maker](#) και το σχεσιακό μοντέλο με το εργαλείο [dbdesigner](#).

### 1.2 Περιγραφή

Η βάση δεδομένων που σχεδιάσαμε αναπτύχθηκε με σκοπό να χρησιμοποιηθεί σε μια εφαρμογή υποστήριξης ενός εστιατορίου. Έχοντας ετούτο τον σκοπό στο μυαλό μας αναπτύξαμε την βάση μας ώστε να παρέχει στον ιδιοκτήτη του εστιατορίου και στους υπαλλήλους του τα απαραίτητα δεδομένα για να γίνεται με ευκολία η διαχείριση και η λειτουργία του εστιατορίου.

Η βάση μας δίνει ιδιαίτερη βάση στις παραγγελίες προμηθειών από το εστιατόριο στους προμηθευτές του, στις παραγγελίες που γίνονται από τους πελάτες του εστιατορίου εντός και εκτός αυτού, στις κρατήσεις των τραπέζιων και στην διαχείριση των υπαλλήλων, των πελατών και των προμηθευτών του εστιατορίου. Δεν δόθηκε ιδιαίτερη βάση σε οικονομικά θέματα καθώς και σε θέματα που αφορούν συμβόλαια του εστιατορίου.

## 2 ΔΙΑΔΙΚΑΣΙΑ ΣΧΕΔΙΑΣΗΣ

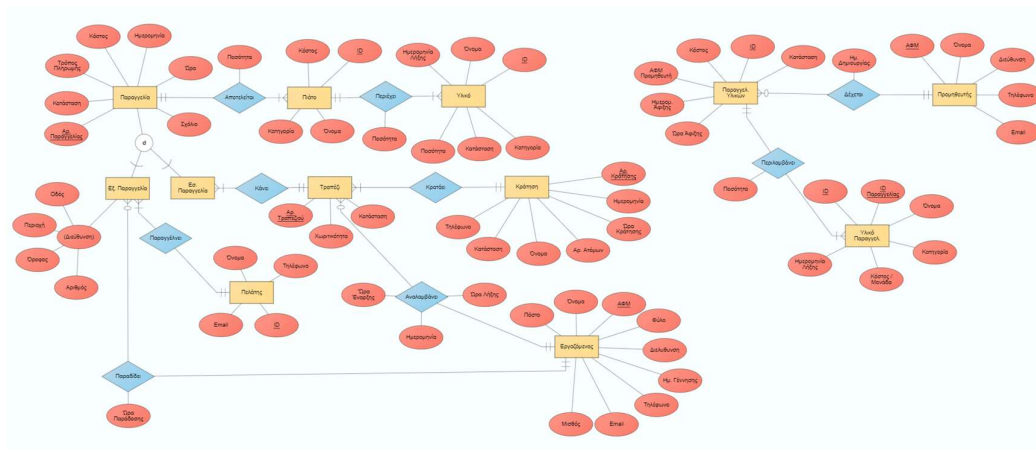
Στην ενότητα αυτή περιγράφουμε τα βήματα που ακολουθήσαμε για την δημιουργία της βάσης δεδομένων μας. Τα βήματα που ακολουθήσαμε είναι με την σειρά, η σχεδίαση του μοντέλου οντοτήτων συσχετίσεων ERD, η μετάφραση του ERD σε σχεσιακό μοντέλο και η σχεδίαση του και τέλος η δημιουργία της βάσης με χρήση SQLite και Python. Τα queries και ο κώδικας για την δημιουργία της βάσης αναλύονται στην ενότητα 3, σε αυτή την ενότητα όμως σας δείχνουμε την αντιστοίχιση των ελληνικών ονομάτων που φαίνονται στα διαγράμματα στα αγγλικά ονόματα της βάσης.

### 2.1 Μοντέλο Οντοτήτων Συσχετίσεων

Το μοντέλο οντοτήτων συσχετίσεων της βάσης ή αλλιώς ERD (Entity Relation Diagram) αποτελείται από δώδεκα οντότητες και από εννέα συσχετίσεις μεταξύ αυτών. Η δημιουργία του ERD μοντέλου ήταν μια σχετικά δύσκολη διαδικασία καθώς δεν υπάρχει εξ'ολοκλήρου μια σωστή προσέγγιση, παρόλα αυτά μετά από πολλή σκέψη και μερικές αλλαγές καθόλη την διάρκεια παραγωγής της εργασίας καταλήξαμε στο σχέδιο που φαίνεται στην εικόνα 1. Οι οντότητες που περιέχει το ERD μας είναι οι εξής : Υλικό, Υλικό Παραγγελίας, Προμηθευτής, Παραγγελία Υλικών, Πιάτο, Παραγγελία, Εξωτερική Παραγγελία, Εσωτερική Παραγγελία, Τραπέζι, Πελάτης, Κράτηση, Εργαζόμενος. Στη συνέχεια ακολουθεί μια σύντομη περιγραφή για την κάθε οντότητα.

- **Υλικό** : Αντιπροσωπεύει τα υλικά που διαθέτει το εστιατόριο και με τα οποία μπορεί να δημιουργήσει τα πιάτα που προσφέρει.
- **Υλικό Παραγγελίας** : Αντιπροσωπεύει τα υλικά που βρίσκονται σε παραγγελίες υλικών που έχουν γίνει από το εστιατόριο στους προμηθευτές.

- Προμηθευτής : Αντιπροσωπεύει τους προμηθευτές, που προμηθεύουν στο εστιατόριο τα υλικά του.
- Παραγγελία Υλικών : Αντιπροσωπεύει τις παραγγελίες από το εστιατόριο στους προμηθευτές του.
- Πιάτο : Αντιπροσωπεύει τα πιάτα ή αλλιώς το μενού του εστιατορίου.
- Παραγγελία : Αντιπροσωπεύει όλες τις παραγγελίες από τους πελάτες που γίνονται είτε εντός είτε εκτός του εστιατορίου. Επίσης η οντότητα Παραγγελία με τις οντότητες Εσωτερική Παραγγελία και εξωτερική παραγγελία έχουν σχέση κλάσης – υποκλάσης μεταξύ τους.
- Εξωτερική Παραγγελία : Αντιπροσωπεύει τις παραγγελίες των πελατών στο εστιατόριο που δεν βρίσκονται στο χώρο του εστιατορίου και τις παραγγελίες που είναι take-away.
- Εσωτερική Παραγγελία : Αντιπροσωπεύει τις παραγγελίες που κάνουν οι πελάτες που θα κάτσουν σε τραπέζι.
- Τραπέζι : Αντιπροσωπεύει τα τραπέζια που διαθέτει το εστιατόριο.
- Πελάτης : Αντιπροσωπεύει τους πελάτες του εστιατορίου που βρίσκονται στο πελατολόγιο.
- Κράτηση : Αντιπροσωπεύει τις κρατήσεις τραπέζιών που γίνονται στο εστιατόριο.
- Εργαζόμενος : Αντιπροσωπεύει τους υπάλληλους του εστιατορίου.

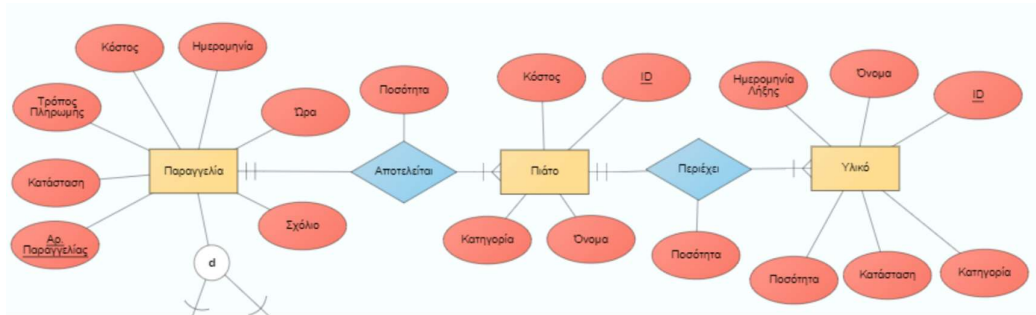


Εικόνα 1 :Μοντέλο Οντοτήτων Συσχετίσεων της βάσης δεδομένων.

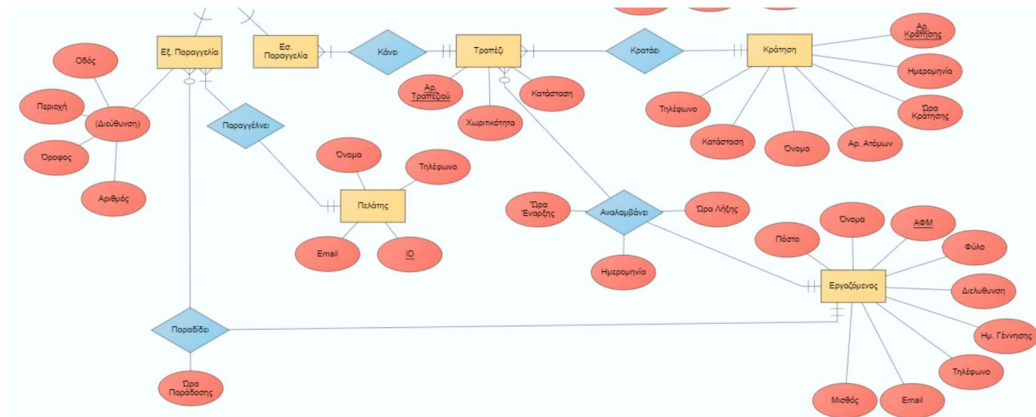
Οι σχέσεις μεταξύ των οντοτήτων της βάσης μας είναι οι ακόλουθες :

- Περιέχει : Ένα πιάτο περιέχει από ένα έως πολλά υλικά.
- Αποτελείτε : Μία παραγγελία αποτελείτε από ένα έως πολλά πιάτα.
- Παραγγέλλνει : Ένας πελάτης παραγγέλλνει από μια έως πολλές παραγγελίες.
- Κάνει : Ένα τραπέζι κάνει από μία έως πολλές εσωτερικές παραγγελίες.
- Κρατάει : Μια κράτηση κρατάει από ένα έως πολλά τραπέζια.
- Αναλαμβάνει : Ένας εργαζόμενος αναλαμβάνει κανένα έως πολλά τραπέζια.
- Παραδίδει : Ένας εργαζόμενος παραδίδει καμία έως πολλές εξωτερικές παραγγελίες.
- Δέχεται : Ένας προμηθευτής δέχεται καμία έως πολλές παραγγελίες υλικών.
- Περιλαμβάνει : Μία παραγγελία υλικών περιλαμβάνει ένα έως πολλά υλικά παραγγελίας.

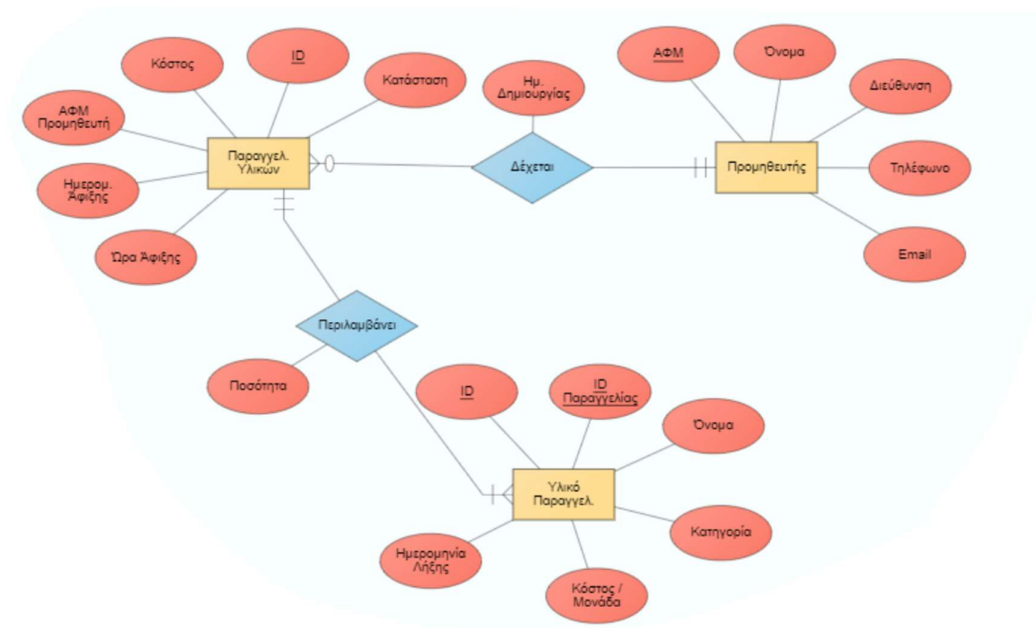
Οι εικόνες 2, 3 και 4 είναι κομμάτια του ERD σε περίπτωση που είναι δύσκολη η ανάγνωση της πρώτης εικόνας.



Εικόνα 2 : Κομμάτι 1/3 του ERD



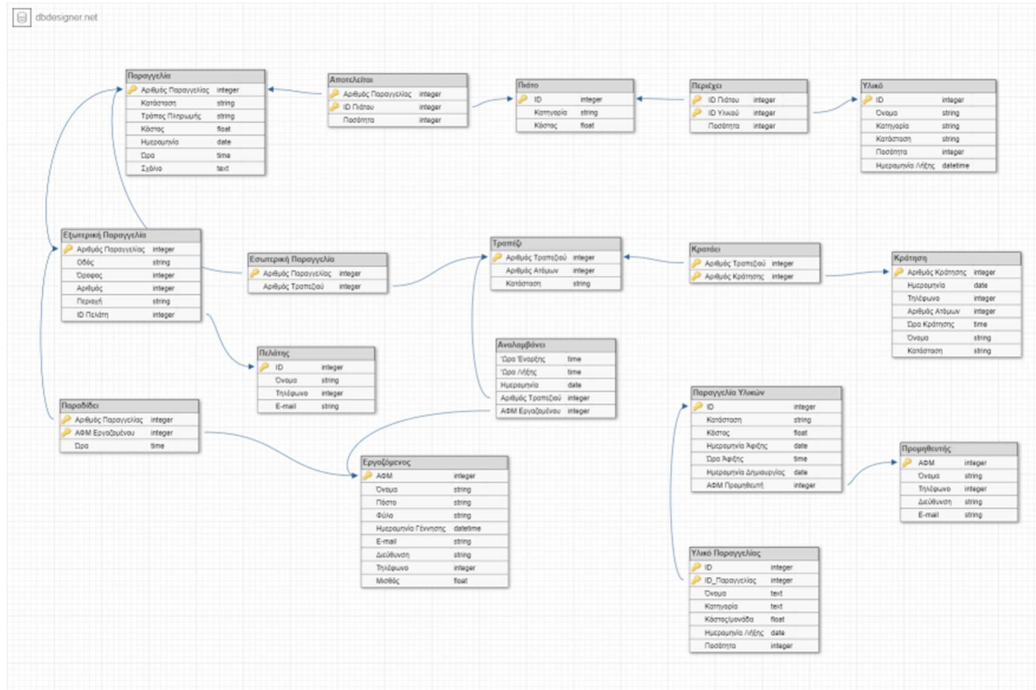
Εικόνα 3 : Κομμάτι 2/3 του ERD



Εικόνα 4 : Κομμάτι 3/3 του ERD

## 2.2 Σχεσιακό Μοντέλο

Μετά από την δημιουργία του διαγράμματος ERD προχωρήσαμε στην σχεδίαση του σχεσιακού μοντέλου (Relational Schema). Η μετατροπή του ERD στο σχεσιακό μοντέλο ήταν μια σχετικά πιο εύκολη διαδικασία από αυτής της δημιουργίας του ERD αλλά χρειάστηκε προσοχή στους κανόνες μετάφρασης και στην επιλογή του κατάλληλου από αυτούς όπου υπήρχε η δυνατότητα επιλογής. Το σχεσιακό μοντέλο στο οποίο καταλήξαμε είναι αυτό που φαίνεται στην εικόνα 5 χωρίς να σημαίνει ότι άλλες εκδοχές δεν ήταν εξίσου κατάλληλες.



Εικόνα 5 : Το Σχεσιακό Μοντέλο της βάσης

## 2.3 Αντιστοίχιση ονομάτων σχεσιακού μοντέλου και βάσης

Η επιλογή των ονομάτων για τους πίνακες και τα τις ιδιότητες της βάσης δεδομένων, έγιναν στα αγγλικά για ευνόητους λόγους. Σε αυτό το κομμάτι της αναφοράς γίνεται η αντιστοίχιση των πινάκων της βάσης με το σχεσιακό μοντέλο. Η αντιστοιχήσεις φαίνονται στον παρακάτω πίνακα.

Πίνακας 1 : Αντιστοίχιση Σχεσιακού Μοντέλου με Βάση Δεδομένων

Σχεσιακό Μοντέλο	Βάση Δεδομένων
Προμηθευτής	SUPPLIER
Παραγγελία	INGREDIENT_ORDER
Υλικό Παραγγελίας	INGREDIENT_INCLUDED_IN_ORDER
Υλικό	INGREDIENT
Περιέχει	MADE_OF
Πιάτο	DISH
Αποτελείται	CONSISTS
Παραγγελία	RESTAURANT_ORDER
Εξωτερική Παραγγελία	OUTSIDE_ORDER
Εσωτερική Παραγγελία	INSIDE_ORDER
Πελάτης	CUSTOMER
Τραπέζι	TABLE
Κρατάει	RESERVE
Κράτηση	RESERVATION
Αναλαμβάνει	HANDLES
Εργαζόμενος	EMPLOYEE
Παραδίδει	DELIVER

### 3 ΚΩΔΙΚΑΣ

#### 3.1 Κώδικας Δημιουργίας Βάσης

Ο κώδικας για την δημιουργία της βάσης μας βρίσκεται στο αρχείο `database_creation.py`. Όπως αναφέραμε και στην εισαγωγή η χρησιμοποιήσαμε για τη συγγραφή του κώδικα την γλώσσα Python και μάλιστα την έκδοση 3.11 αυτής. Για την δημιουργία μιας βάσης SQLite χρησιμοποιήσαμε τις συναρτήσεις που μας παρέχει η βιβλιοθήκη της python με το αναμενόμενο όνομα `sqlite3`. Οι πρώτες γραμμές του αρχείου αυτού είναι οι παρακάτω,

```
import sqlite3 as sql

database = "restaurant.db"

connection = sql.connect(database)

connection.execute("PRAGMA foreign_keys = 1")

cursor = connection.cursor()
```

Αυτές οι γραμμές είναι καθοριστικής σημασίας και για τα υπόλοιπα αρχεία μας αι για αυτό τον λόγο δίνουμε έμφαση στην περιγραφή τους. Η πρώτη γραμμή του κώδικα μας επιτρέπει να “εισάγουμε” στον κώδικα μας την βιβλιοθήκη της `sqlite3`, δηλαδή μας επιτρέπει να χρησιμοποιήσουμε τις συναρτήσεις που είναι ορισμένες στο αρχείο `sqlite3.py`, για συντομία δηλώνουμε ότι όπου θέλουμε να χρησιμοποιήσουμε συνάρτηση της βιβλιοθήκης αυτής αντί για `sqlite3.όνομα_συνάρτησης` θα γράφουμε `sql.όνομα_συνάρτησης`. Στη δεύτερη γραμμή του κώδικα ορίζουμε ότι η μεταβλητή `database` θα αποθηκεύει το string “restaurant.db” το οποίο είναι και το όνομα του αρχείου στο οποίο αποθηκεύουμε την βάση δεδομένων που έχουμε φτιάξει. Στη Τρίτη γραμμή καλούμε τη συνάρτηση `connect()` της `sqlite3` και της περνάμε σαν όρισμα την μεταβλητή `database` που ορίσαμε προηγουμένως, με αυτό τον τρόπο δημιουργούμε μια σύνδεση μεταξύ του κώδικά μας και της βάσης μας και την αποθηκεύουμε στη μεταβλητή `connection`. Για να μπορούμε να επεξεργαζόμαστε τα δεδομένα της βάσης μας χρειαζόμαστε έναν `cursor` για αυτή, με την πέμπτη γραμμή του κώδικά μας ουσιαστικά φτιάχνουμε έναν τέτοιο `cursor` και τον αναθέτουμε στην μεταβλητή `cursor`. Τέλος η τέταρτη γραμμή ενεργοποιεί τον περιορισμό του ξένου κλειδιού ( `foreign key constraint` ) για την βάση μας.

Στο αρχείο `database_creation.py` οι υπόλοιπες γραμμές που ακολουθούν ουσιαστικά είναι η εκτέλεση της συνάρτησης `sqlite3.Cursor.execute` δεκαεπτά φορές, μια για κάθε πίνακα που θέλουμε να δημιουργήσουμε. Τα ονόματα των πινάκων φαίνονται στον πίνακα 1. Θα παρουσιάσουμε μόνο τρία παραδείγματα δημιουργίας πινάκων γιατί δεν έχει νόημα να τα δείξουμε όλα. Η συνάρτηση `sqlite3.Cursor.execute` δέχεται σαν όρισμα ένα string που περιέχει ένα query στη γλώσσα `sqlite3`. Τα queries που έχουμε στο αρχείο αυτό είναι μόνο queries δημιουργίας και για αυτό δείχνουμε μόνο τρία από αυτά.

Πρώτο παράδειγμα,

```
cursor.execute("""

    CREATE TABLE IF NOT EXISTS RESTAURANT_ORDER(

        order_number INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

        state TEXT DEFAULT "open" NOT NULL,

        payment_method TEXT NOT NULL,

        cost REAL NOT NULL,

        date TEXT DEFAULT (date('now','localtime')) NOT NULL,

        time TEXT DEFAULT (time('now','localtime')) NOT NULL,

        comment TEXT

    )

""")
```

Στο παράδειγμα αυτό δημιουργούμε έναν πίνακα σε περίπτωση που δεν υπάρχει ήδη στη βάση μας με το όνομα “RESTAURANT\_ORDER”, οι πίνακας αυτός αποτελείται από επτά στήλες ή αλλιώς μπορούμε να πούμε ότι έχει επτά ιδιότητες. Ας δούμε τις στήλες το πίνακα μια μια. Πρώτη στήλη είναι η στήλη με όνομα “order\_number” η

στήλη αυτή είναι τύπου INTEGER δηλαδή αποθηκεύει ακέραιους αριθμούς και την έχουμε ορίσει ως PRIMARY KEY του πίνακα, δηλαδή έχουμε ορίσει ότι κάθε γραμμή (κάθε record) του πίνακα αυτού θα ξεχωρίζει σίγουρα από τα άλλα με την τιμή που έχει στη στήλη αυτή. Ακόμα έχουμε ορίσει ότι η στήλη αυτή θα είναι AUTOINCREMENT, αυτό σημαίνει ότι αν η προηγούμενη γραμμή του πίνακα έχει μια τιμή τύπου INTEGER στη στήλη αυτή μπορούμε να εισάγουμε μια νέα στήλη χωρίς να δώσουμε τιμή για το δεδομένο που θα έχει σε αυτή τη στήλη, το δεδομένο αυτό δεν θα μείνει άδειο αλλά θα αποκτήσει την ακριβώς αμέσως ακέραια τιμή από το δεδομένο της στήλης αυτής της προηγούμενης γραμμής. Τέλος για την πρώτη στήλη έχουμε ορίσει ότι δεν μπορεί να πάρει την τιμή NULL, δηλαδή ότι πρέπει πάντα να μην την αφήνουμε κενή. Η δεύτερη στήλη του πίνακα αυτού ονομάζεται “state”, είναι τύπου TEXT και παίρνει σαν DEFAULT τιμή δηλαδή αν δεν ορίσουμε εμείς κάποια τιμή θα πάρει αυτόματα την τιμή “open”. Στη τρίτη γραμμή ορίζεται η στήλη “payment\_method” που είναι και αυτή τύπου TEXT και έχει τον περιορισμό NOT NULL. Στη τέταρτη γραμμή ορίζουμε τη στήλη “cost” η οποία είναι τύπου FLOAT αποθηκεύει δηλαδή τους αριθμούς σε μορφή 8 byte αριθμού κινητής υποδιαστολής σύμφωνα με το πρότυπο της IEEE και βάζουμε τον περιορισμό NOT NULL. Στη πέμπτη γραμμή ορίζουμε την στήλη “date” η οποία είναι τύπου TEXT και παίρνει σα DEFAULT τιμή την τιμή που παράγει η συνάρτηση date μέσα στις παρενθέσεις. Η συνάρτηση date όταν καλείται με τις παραμέτρους (“now”, “localtime”) επιστρέφει ένα string με το ακόλουθο format ‘YYYY-MM-DD’ την τωρινή ημερομηνία στην ώρα του συστήματος στο οποίο τρέχει. Η στήλη “date” παίρνει και αυτή τον περιορισμό NOT NULL. Η έκτη γραμμή ορίζει την “time” στήλη του πίνακα και στο μόνο που διαφέρει από την στήλη “date” είναι ότι σα DEFAULT τιμή δέχεται το string που επιστρέφει η συνάρτηση time με τις παραμέτρους (“now”, “localtime”) του οποίου η τιμή δεν είναι άλλη από την τιμή της τωρινής ώρας στο σύστημα που τρέχει η συνάρτηση, με τη μορφή ‘HH:MM:SS’. Τέλος η τελευταία γραμμή ορίζει την τελευταία στήλη του πίνακα η οποία ονομάζεται “comment” και είναι τύπου TEXT.

Δεύτερο παράδειγμα,

```
cursor.execute("""

CREATE TABLE IF NOT EXISTS CONSISTS(

order_number INTEGER NOT NULL,

dish_id INTEGER NOT NULL,

quantity INTEGER NOT NULL,

PRIMARY KEY (order_number,dish_id),

FOREIGN KEY (order_number) REFERENCES RESTAURANT_ORDER(order_number) ON DELETE

CASCADE,

FOREIGN KEY (dish_id) REFERENCES DISH(id)

)

""")
```

Στο παράδειγμα αυτό δημιουργούμε έναν πίνακα ονόματι “CONSISTS”, το μόνο που έχει σημασία να αναφέρουμε είναι οι τελευταίες γραμμές του query. Ένας άλλος τρόπος ορισμού του PRIMARY KEY του πίνακα φαίνεται σε αυτό το παράδειγμα, αντί να θέσουμε τον περιορισμό πρωτεύοντος κλειδιού σε μια στήλη αμέσως μετά τον ορισμό της, μπορούμε να το κάνουμε με τον εξής τρόπο που φαίνεται στο παράδειγμα αυτό. Στο παράδειγμα αυτό επίσης φαίνεται η χρήση του περιορισμού του ξένου κλειδιού FOREIGN KEY. Ο περιορισμός αυτός συντάσσεται ως εξής,

FOREIGN KEY (στήλη1) REFERENCES πίνακα2(στήλη2)

Ο περιορισμός αυτός μας λέει ότι η στήλη1 η οποία είναι μια στήλη του πίνακα που ορίζουμε, έχει άμεση συσχέτιση (δείχνει) με την στήλη2 του πίνακα2. Οι τιμές που παίρνει η στήλη1 δεν μπορούν να διαφέρουν από αυτές που υπάρχουν στην στήλη2. Ακόμα, ο περιορισμός ON DELETE CASCADE που συναντάμε στην ίδια γραμμή σημαίνει ότι αν σβηστεί η γραμμή του πίνακα2 που σχετίζεται με τη σχέση ξένου κλειδιού μιας γραμμής του πίνακα που ορίζουμε τώρα, τότε θα σβηστεί και η γραμμή αυτού του πίνακα.

Τρίτο Παράδειγμα,

```
cursor.execute("""

CREATE TABLE IF NOT EXISTS DISH(

id INTEGER NOT NULL,

name TEXT NOT NULL,
```

```

        category TEXT NOT NULL,

        cost REAL NOT NULL,

        PRIMARY KEY (id)

    )

"""
)

```

Αυτό το παράδειγμα θα μπορούσε να παραληφθεί αλλά συμπληρώνει τα δύο από πάνω οπότε και για αυτό το βάλαμε στην αναφορά.

### 3.2 Ευρετήριο Συναρτήσεων

Σε αυτό το κομμάτι εξηγούμε τι κάνει κάθε συνάρτηση που περιέχεται στο κώδικά μας και επεξεργάζεται, εισάγει και ψάχνει δεδομένα στη βάση μας. Όλες οι συναρτήσεις έχουν γραφεί στην έκδοση της Python 3.11 και βρίσκονται στο αρχείο functions.py. Δεν γίνεται αναλυτική επεξήγηση των συναρτήσεων, απλά γίνεται μια σύντομη περιγραφή της λειτουργίας τους.

create\_order( dishes ) :

Η συνάρτηση αυτή δημιουργεί παραγγελίες και τις τοποθετεί στον πίνακα RESTAURANT\_ORDER. Παίρνει ως παράμετρο dishes ένα λεξικό το οποίο για κλειδιά του έχει τα id των πιάτων που βρίσκονται στον πίνακα DISHES της βάσης και ως τιμές την ποσότητα στην οποία θέλουμε να παραγγείλουμε το κάθε πιάτο. Επίσης εισάγει στον πίνακα CONSISTS τις συσχετίσεις παραγγελίας και πιάτων. Σαν έξοδο επιστρέφει τον αριθμό της παραγγελίας.

create\_employee( tin, name, role, sex, birthday, email, address, phone\_number, salary ) :

Η συνάρτηση αυτή δημιουργεί νέους υπαλλήλους και τους εισάγει στον πίνακα EMPLOYEE. Ως παραμέτρους δέχεται το ΑΦΜ (tin), το όνομα (name), το φύλο (sex), την ημερομηνία γέννησης (birthday), την διεύθυνση (address), το τηλέφωνο (phone\_number) και το μισθό (salary) του εργαζομένου. Κάνει έλεγχο αν υπάρχει υπάλληλος με τον ίδιο αριθμό ΑΦΜ και αν ναι εμποδίζει την εισαγωγή του στον πίνακα.

create\_customer( name, phone\_number, email ) :

Η συνάρτηση αυτή εισάγει νέους πελάτες στον πίνακα CUSTOMER και δέχεται ως παραμέτρους το όνομα (name), το τηλέφωνο (phone\_number) και το email του πελάτη.

create\_table( capacity ) :

Η συνάρτηση αυτή δημιουργεί νέα τραπέζια και τα εισάγει στον πίνακα TABLE, δέχεται ως όρισμα τη χωρητικότητα (capacity) του τραπεζιού.

add\_reservation( name, date, time, phone\_number, people ) :

Η συνάρτηση δημιουργεί μια νέα κράτηση και την εισάγει στον πίνακα RESERVATION. Σαν παραμέτρους δέχεται το όνομα (name) στο οποίο γίνεται η κράτηση, την ώρα ( time ) της κράτησης, ένα τηλέφωνο ( phone\_number ) και τον αριθμό ατόμων για τα οποία γίνεται η κράτηση (people). Επιπλέον ελέγχει αν υπάρχει ένα ελεύθερο τραπέζι από τον πίνακα TABLE και αν υπάρχει αλλάζει την κατάσταση του σε reserved, αλλιώς εμποδίζεται η δημιουργία της κράτησης.

add\_supplier( tin, name, phone\_number, address, email = None ) :

Η συνάρτηση αυτή προσθέτει νέους προμηθευτές για το εστιατόριο στον πίνακα SUPPLIER. Οι παράμετροι της συνάρτησης είναι το ΑΦΜ (tin) του προμηθευτή, το όνομά του (name), ένα τηλέφωνο (phone\_number) για να μπορεί να γίνεται η επικοινωνία μαζί του, η διεύθυνση (address) του προμηθευτή και ένα email.

create\_dish( ingredients, dish\_id, name, category, cost ) :

Η συνάρτηση δημιουργεί πιάτα και τα προσθέτει στον πίνακα DISH που αποτελεί το menu του εστιατορίου. Σα παραμέτρους δέχεται ένα λεξικό (ingredients) το οποίο έχει ως κλειδιά τα id των υλικών που χρησιμοποιούνται στο πιάτο και σαν τιμές την ποσότητα στην οποία χρησιμοποιούνται, το id (dish\_id) που θέλουμε να έχει το πιάτο, το όνομα (name) του πιάτου, την κατηγορία (category) στην οποία ανήκει και το κόστος (cost) με το οποίο το προσφέρει το εστιατόριο. Επίσης προσθέτει τα κατάλληλα δεδομένα στον πίνακα MADE\_OF ο οποίος κάνει την αντιστοιχία ποια υλικά χρησιμοποιούνται σε ποιο πιάτο.

order\_ingredients( ingredients, supplier\_tin ) :

Η συνάρτηση αυτή εισάγει στον πίνακα INGREDIENT\_ORDER τις παραγγελίες υλικών. Δέχεται ως παραμέτρους ένα λεξικό (ingredients) το οποίο έχει ως κλειδιά τα id των υλικών που παραγγέλλει το εστιατόριο και σε τιμές μια λίστα για κάθε υλικό η οποία περιλαμβάνει το όνομα, την κατηγορία, ποσότητα, κόστος ανά μονάδα και ημερομηνία λήξης για κάθε υλικό που παραγγέλλουμε. Ακόμα δέχεται σε παράμετρο και το ΑΦΜ (supplier\_tin) του προμηθευτή στον οποίο γίνεται η παραγγελία. Σε περίπτωση που δεν υπάρχει ο προμηθευτής με το ΑΦΜ που δίνουμε τότε εμποδίζεται η δημιουργία της παραγγελίας. Επίσης κάθε υλικό που παραγγέλλεται εισάγεται στον πίνακα INGREDIENT\_INCLUDED\_IN\_ORDER.

employee\_handle\_table() :

Η συνάρτηση αυτή δίνει την δυνατότητα να γίνει ανάθεση τραπεζιών στους εργαζόμενους. Με άλλα λόγια ορίζει για ποια τραπεζία θα είναι υπεύθυνος ένας συγκεκριμένος εργαζόμενος (προφανώς σερβιτόρος) για κάποια συγκεκριμένη χρονική διάρκεια και ημερομηνία. Τις αναθέσεις αυτές τις εισάγει στον πίνακα HANDLES. Σε περίπτωση που δώσουμε λάθος εσφαλμένα στοιχεία ή ίδια στοιχεία σε μια ανάθεση τότε εμποδίζεται προφανώς η εισαγωγή της στον πίνακα.

create\_outside\_order( dishes, name, phone\_number, email = None, street = None, floor = None, number = None, area = None, customer\_id = None) :

Η συνάρτηση αυτή δημιουργεί εξωτερικές παραγγελίες πελατών προς το κατάστημα και τις εισάγει στον πίνακα OUTSIDE\_ORDER. Δέχεται πολλές παραμέτρους μόνο τρεις από αυτές όμως είναι υποχρεωτικές. Οι παράμετροι είναι οι εξής, ένα λεξικό (dishes) με κλειδιά τα id των πιάτων και τιμές την ποσότητα στην οποία θέλει ο πελάτης να τα παραγγείλει, το όνομα (name) στο οποίο γίνεται η παραγγελία, ένα τηλέφωνο (phone\_number) για επικοινωνία με τον πελάτη, δρόμο (street), όροφο (floor), αριθμό (number), περιοχή (area) της διεύθυνσης στην οποία κάνει ο πελάτης την παραγγελία και αν υπάρχει ήδη ο πελάτης στο πελατολόγιο το id του (customer\_id). Μόνο οι πρώτες τρεις παράμετροι είναι υποχρεωτικές επειδή στις εξωτερικές παραγγελίες συμπεριλαμβάνονται και αυτές που είναι take away. Οι συνάρτηση αυτή καλεί επίσης και την συνάρτηση create\_order που είδαμε προηγουμένως ώστε να εισάγει τα κατάλληλα δεδομένα στον πίνακα RESTAURANT\_ORDER. Ακόμα σε περίπτωση που δεν υπάρχει ήδη ο πελάτης που έκανε την παραγγελία στο πελατολόγιο δηλαδή στον πίνακα CUSTOMER τον εισάγει με τα κατάλληλα δεδομένα που δώσαμε στις παραμέτρους.

confirm\_ingredient\_order\_arriva( order\_id ) :

Η συνάρτηση αυτή δέχεται σε παράμετρο το id (order\_id) μιας παραγγελίας υλικών και αν δεν έχει ολοκληρωθεί ήδη την σημειώνει ως ολοκληρωμένη και αντιγράφει όλα τα υλικά του πίνακα INGREDIENT\_INCLUDED\_IN\_ORDER στον πίνακα INGREDIENT. Τα υλικά που είναι τελείως καινούργια αποθηκεύονται σε νέο entry ενώ για εκείνα που υπάρχουν ήδη ουσιαστικά γίνεται ενημέρωση της ποσότητας και της ημερομηνίας λήξης τους.

create\_inside\_order( table\_number, dishes):

Η συνάρτηση αυτή δημιουργεί εσωτερικές παραγγελίες και τις εισάγει στον πίνακα INSIDE\_ORDER. Σε παραμέτρους δέχεται τον αριθμό του τραπεζιού (table\_number) από το οποίο έγινε η παραγγελία και ένα λεξικό (dishes) με κλειδιά τα id των πιάτων και τιμές τη ποσότητα στην οποία παραγγέλλονται. Όπως και η create\_outside\_order καλεί την συνάρτηση create\_order.

clear\_reserved\_tables():

Μέσω της συνάρτησης αυτής μπορούμε να αλλάξουμε την κατάσταση των τραπεζιών που είναι reserved και να την κάνουμε free. Αν δεν υπάρχουν κρατημένα τραπέζια δεν μας αφήνει να προχωρήσουμε.

ingredient\_orders\_summary( start\_date, end\_date = None) :

Η συνάρτηση αυτή υπολογίζει πόσες παραγγελίες υλικών έχουν δημιουργηθεί στο χρονικό διάστημα από την ημερομηνία start\_date μέχρι την ημερομηνία end\_date (αν υπάρχει), επίσης υπολογίζει το συνολικό κόστος αυτών των παραγγελιών καθώς και το πόσες από αυτές δεν έχουν παραληφθεί ακόμα.

assign\_delivery( order\_number, employee\_tin ) :

Η συνάρτηση αυτή αναθέτει την παράδοση παραγγελιών σε εργαζόμενους και εισάγει τα δεδομένα αυτά στον πίνακα DELIVER. Παίρνει σε παραμέτρους τον αριθμό παραγγελίας (order\_number) και το ΑΦΜ του εργαζομένου (employee\_tin) στον οποίο αναθέτεται. Η συνάρτηση αυτή έχει νόημα μόνο για εξωτερικές παραγγελίες.

restaurant\_orders\_summary( start\_date, end\_date ) :



Η συνάρτηση αυτή υπολογίζει τον αριθμό παραγγελιών που έγιναν στο εστιατόριο, πόσες από αυτές ήταν εσωτερικές, πόσες από αυτές ήταν εξωτερικές και το κέρδος από τις εσωτερικές και εξωτερικές παραγγελίες.

`mark_restaurant_order_as_finished( order_number ) :`

Η συνάρτηση αυτή σηματοδοτεί την ολοκλήρωση μιας παραγγελίας.

`not_finished_restaurant_orders() :`

Η συνάρτηση αυτή επιστρέφει τις παραγγελίες που δεν έχουν ολοκληρωθεί.

`most_ordered_dish() :`

Η συνάρτηση αυτή επιστρέφει το πιάτο που έχει παραγγελθεί περισσότερες φορές.

`best_customer() :`

Η συνάρτηση αυτή επιστρέφει τον πελάτη που βρίσκεται στο πελατολόγιο και έχει κάνει τις περισσότερες παραγγελίες.

`today's_reservations() :`

Η συνάρτηση αυτή επιστρέφει τις κρατήσεις που έχουν γίνει τη μέρα που καλείται.

### 3.4 Εφαρμογή Διεπαφής Χρήστη

Ακολουθώντας την συμβουλή των διδασκόντων μας να μην ασχοληθούμε με την γραφική διεπαφή στην υλοποίησή μας, δημιουργήσαμε ένα περιβάλλον στο οποίο ο χρήστης θα πρέπει να πατάει κάποιο πλήκτρο το οποίο θα αντιστοιχεί με την ενέργεια που θέλει να πραγματοποιήσει. Παρόλο που φαίνεται απλή αυτή η υλοποίηση είναι αρκετή ώστε να καλύψει τις ανάγκες (σε ένα αρχικό στάδιο) του χρήστη της. Πάντα βέβαια ελπίζουμε στην σωστή χρήση αφού ενώ έχουμε βάλει μερικές βασικές δικλίδες ασφαλείας δεν είναι δυνατό να προβλέψουμε οποιαδήποτε λανθασμένη είσοδο του χρήστη

```
Welcome to Restautant Database
If you want to ADD data PRESS 1
If you want to SEE data PRESS 2
If you want to UPDATE data PRESS 3
For EXTRA'S PRESS 4
If you want to EXIT from the database PRESS ENTER
2

To see CUSTOMER TABLE PRESS 1
To see EMPLOYEE TABLE PRESS 2
To see RESERVATION TABLE PRESS 3
To see SUPPLIER TABLE PRESS 4
To see RESTAURANT_TABLE TABLE PRESS 5
To see DISH TABLE PRESS 6
To see HANDLES TABLE PRESS 7
To see ORDER_INGREDIENTS TABLE PRESS 8
To see INGREDIENT TABLE PRESS 9
To see ORDER TABLE PRESS 10
To see MADE_OF TABLE PRESS 11
To see INGREDIENT_INCLUDED_IN_ORDER TABLE PRESS 12
To see INSIDE_ORDER TABLE PRESS 13
To see OUTSIDE_ORDER TABLE PRESS 14
To see CONSISTS TABLE PRESS 15
To see DELIVERY TABLE PRESS 16
```

Εικόνα 6 : Παράδειγμα της εφαρμογής μας κατά την διάρκεια εκτέλεσης

## 4 ΧΡΟΝΟΔΙΑΓΡΑΜΑ ΚΑΙ ΑΝΑΘΕΣΗ ΔΟΥΛΕΙΑΣ

### 4.1 Χρονοδιάγραμμα

Η εργασία που μας ανατέθηκε μπορούμε να πούμε ότι χωρίστηκε σε δύο μέρη. Το πρώτο μέρος αποτελείται από την δημιουργία του ERD καθώς και από την μετατροπή του στο σχεσιακό μοντέλο. Η ολοκλήρωση αυτών των ενεργειών έγινε στις αρχές Νοεμβρίου και σήμανε και το τέλος της πρώτης φάσης της εργασίας μας. Το δεύτερο σκέλος αποτελείται από τη δημιουργία της βάσης δεδομένων με όλες τις ιδιότητες που αναφέρονται και αναλύονται στην αναφορά μας. Το δεύτερο σκέλος ολοκληρώθηκε στις διακοπές των Χριστουγέννων. Οι χρονικοί περιορισμοί που τέθηκαν από εμάς καθόλη την διάρκεια του εξαμήνου τηρήθηκαν πλήρως και από τους δύο μας.

### 4.2 Ανάθεση δουλειάς

Από την αρχή της συνεργασίας μας υπήρχε διάθεση να ασχοληθούμε και οι δύο με όλα τα κομμάτια της εργασίας ώστε να αποκτήσουμε μια καλή εικόνα για τον σχεδιασμό μιας βάσης δεδομένων. Από αυτό συμπεραίνει κανείς ότι δεν μπορούμε να ορίσουμε ακριβώς τι έκανε ο καθένας. Για την δημιουργία του ERD και του σχεσιακού μοντέλου δουλέψαμε μαζί, το ίδιο και για τη δημιουργία των διάφορων συναρτήσεων και των query για τη δημιουργία της βάσης. Ο διαχωρισμός έγινε στην δημιουργία του προγράμματος διεπαφής με το χρήστη και στη σύνταξη της αναφοράς. Ο φοιτητής Γιώργος Ζωιτάκης ασχολήθηκε με την δημιουργία του προγράμματος διεπαφής ώστε να μπορεί ένας χρήστης να δει εύκολα τη λειτουργικότητα της βάσης μας και ο φοιτητής Ιωάννης Φέρτης σύνταξε την αναφορά, χωρίς αυτό να σημαίνει ότι δεν υπήρχε συνεργασία ακόμα και σε αυτά τα κομμάτια.

## 6. Εγκατάσταση

Ο κώδικας μας αποτελείται από τρία αρχεία, το πρώτο αρχείο ονομάζεται `database_creation.py` και τρέχοντας το δημιουργεί μια βάση δεδομένων ονόματι `“restaurant.db”`, το δεύτερο αρχείο ονομάζεται `functions.py` και περιέχει όλες τις ενδεικτικές συναρτήσεις που έχουμε φτιάξει, το τρίτο και τελευταίο αρχείο ονομάζεται `“user_interface.py”` και αποτελεί ένα πρόγραμμα με το οποίο μπορεί να δει ο χρήστης την λειτουργικότητα της βάσης μας. Για τους σκοπούς της παρουσίασης έχουμε προσθέσει ένα ακόμα αρχείο το `“restaurant_image.db”` το οποίο είναι ένα στιγμιότυπο της βάσης μας με ορισμένες εισαγωγές που έχουμε κάνει εμείς. Αν θέλει κάποιος να χρησιμοποιήσει το στιγμιότυπο αυτό αρκεί πριν τρέξει το `user_interface.py` να αλλάξει την γραμμή `database = “restaurant.db”` σε `database = “restaurant_image.db”`.