

**Name:** Papadopoulos Nikolaos

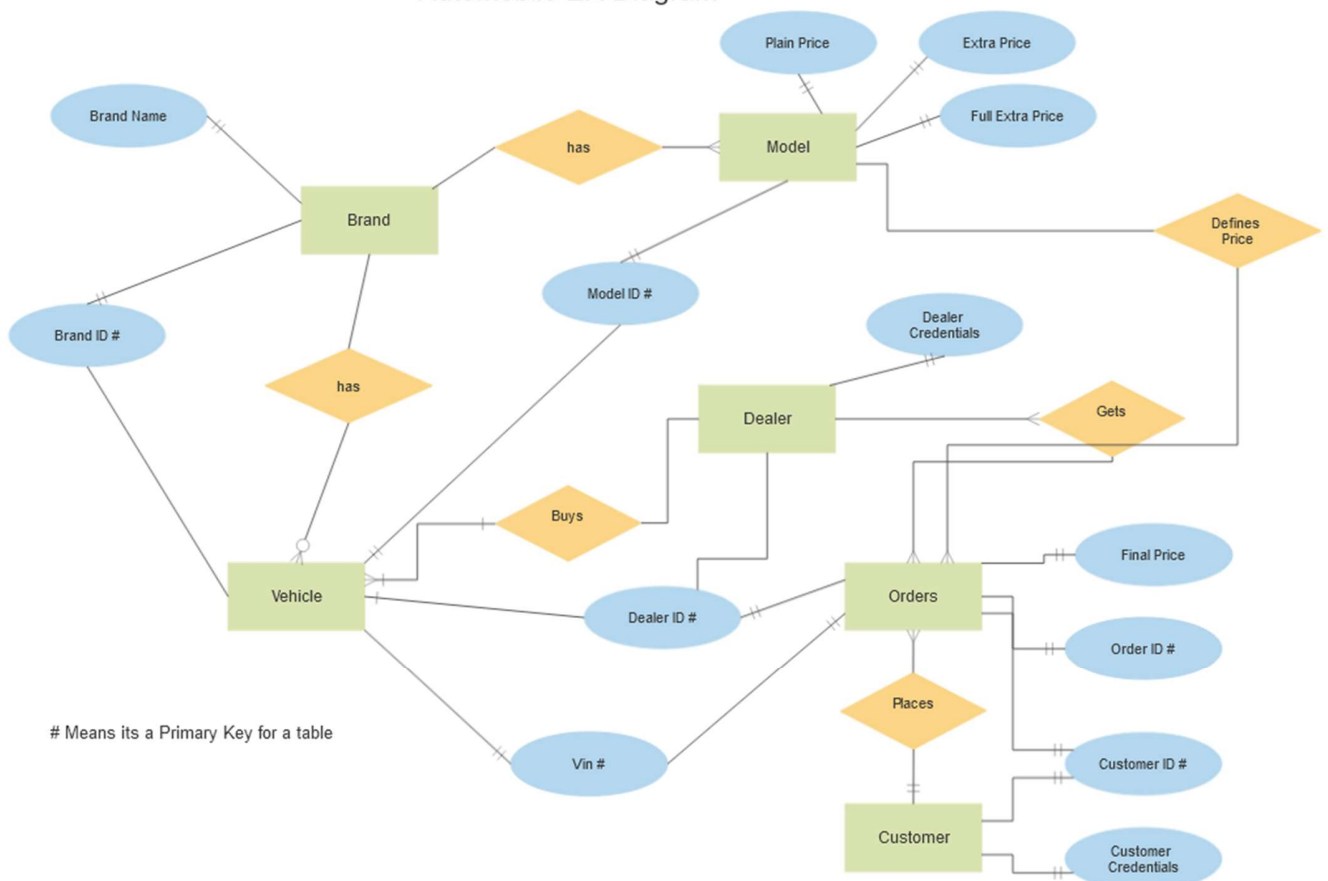
**Student ID:** f3352118

## Introduction to Data Management and Engineering

### PROJECT 2 – E-R Diagrams and Database Programming with Python

1)

Automobile ER Diagram



Στόχος του πρώτου ερωτήματος ήταν να φτιαχτεί ένα σχετικά απλό E-R diagram για μια εταιρεία automobile με σκοπό να βοηθήσει τους dealers στο να κρατάνε καλύτερα customer records/ευκολότερα. Το παραπάνω διάγραμμα σχεδιάστηκε στο cloud.smartdraw.com

Και αποτελείται από 6 βασικούς πίνακες, το Vehicle, Brand, Model, Dealer, Orders και Customer. Κάθε ένας από αυτούς τους πίνακες έχει ένα primary key το οποίο φαίνεται και στον πίνακα(σύμβολο δίσκου). Συγκεκριμένα έχουμε: (Πίνακας-Primary Key)

Vehicle-VIN, Brand-Brand ID, Model-Model ID, Dealer -Dealer ID, Orders -Order ID,  
Customer-Customer ID

## 2)

Σε αυτό το ερώτημα καλούμαστε να υλοποιήσουμε σε πρώτη φάση το database αυτό στην PostgreSQL. Δηλαδή, να τρέξουμε τα create tables για κάθε πίνακα που βλέπουμε στο διάγραμμα, να φτιάξουμε τα απαραίτητα relationships/constraints που πρέπει να έχει ο πίνακας καθώς και να δημιουργήσουμε τις κατάλληλες θέσεις για attributes που θα βοηθήσουν στην επίτευξη του στόχου του database να διευκολύνει την διατήρηση/οργάνωση των δεδομένων των customers.

Στην συνέχεια χρειάζεται να γεμίσουμε τους παραπάνω πίνακες με διάφορα data(στην περίπτωση αυτή με randomly generated data) έτσι ώστε ο κάθε πίνακας να έχει από κάποια δεδομένα για να μπορούμε να τρέξουμε queries στην βάση και να βγάλουμε κάποιο αποτέλεσμα. (Αλλά κοιτάμε ταυτόχρονα να υπάρχει νόημα στα data ως προς τις συνδέσεις τους με τους άλλους πίνακες) Κάθε table έχει από ένα primary key και η πλειοψηφία έχει foreign keys για την σύνδεσή τους με τους άλλους πίνακες.

Το τρίτο ερώτημα ζητάει να φτιάξουμε ευρετήρια για τα ερωτήματα που περιμένουμε να ερωτηθούμε τις περισσότερες φορές έτσι ώστε αυτά να διενεργηθούν γρηγορότερα μέσω του index που θα

φτιάξουμε. Θα ήθελα να αναφέρω πως: Ο προγραμματισμός ερωτημάτων της PostgreSQL είναι αρκετά αποτελεσματικός, επομένως η προσθήκη νέου ευρετηρίου μπορεί να μην επηρεάσει την ταχύτητα εκτέλεσης των ερωτημάτων. Επίσης, η PostgreSQL δημιουργεί ήδη ευρετήρια για κάθε πρωτεύον κλειδί, επομένως η δημιουργία ευρετηρίου σε ένα πρωτεύον κλειδί δεν είναι χρήσιμη. Τα παραπάνω σε συνδυασμό με το μέγεθος των data που έχουμε στην διάθεσή μας πιστεύω πως δεν θα εμφανίσει καλύτερους χρόνους με την δημιουργία indexes για την συγκεκριμένη περίπτωση.(Αλλά θα βοηθήσει αν είχαμε πολύ μεγαλύτερο όγκο δεδομένων).

Για αυτό τον λόγο έφτιαξα 3 index:

i)Για το customer name (Πιστεύω θα υπάρχουν πολλές περιπτώσεις στις οποίες κάποιος θα έψαχνε την βάση δεδομένων με φίλτρο το όνομα του πελάτη για να δει ίσως τις παραγγελίες του ή για να βρει τα στοιχεία του.

ii)Για το dealer name (Όπως και πάνω, η εταιρεία μπορεί να θέλει να ψάξει τα στοιχεία ενός συγκεκριμένου dealer για επικοινωνία).

iii)Final price (Κάθε dealer θα μπορεί να φιλτράρει τις πωλήσεις του και να δει π.χ. τον πελάτη με τις μεγαλύτερες αγορές (Η βάση δεδομένων έχει φτιαχτεί έτσι ώστε ένας πελάτης να μπορεί να κάνει παραπάνω από μία παραγγελίες (Με διαφορετικό order id))

#### 4)

Στο τελευταίο και πιο complex από τα 3 ερωτήματα ζητείται να φτιάξουμε ένα menu of choices in Python το οποίο να συνδέεται στην βάση δεδομένων που έχουμε φτιάξει και να μπορούμε μέσω αυτού να εκτελούμε κάποιες ενέργειες.(Στην συγκεκριμένη περίπτωση φτιάχτηκε στο jupyter notebook) που να περιέχει 3 επιλογές:

##### 1)Entity Management

Το οποίο οδηγεί τον χρήστη στις επιλογές του insert,delete και entity update οι οποίες δουλεύουν στην εισαγωγή νέων data,διαγραφή και ενημέρωση ήδη υπάρχοντων δεδομένων αντίστοιχα.

##### 2)Entity Search

Το οποίο δίνει στον χρήστη 4 επιλογές :

- 1) Search Using Customer's Name
- 2) Search Using Order ID
- 3) Search Using Customer's ID
- 4) Search Using VIN

Δηλαδή του δίνει την δυνατότητα να ψάξει την βάση δεδομένων με κάποιο από τα παραπάνω φίλτρα.

##### 3)Perform Specific Query

Το οποίο δίνει στον χρήστη 2 επιλογές:

- 1) Find Final Price using Customer's Name & Customer's ID
- 2) Find Model's Prices per version using Model ID

Τα παραπάνω αποτελούν queries που θα γίνονται συχνά και για αυτό επιλέχθηκαν για αυτό το ερώτημα και σε συνδυασμό με τα indexes που έχουν φτιαχτεί θα υπάρχει το βέλτιστο αποτέλεσμα για την εκτέλεση αυτών.

Το μενού των επιλογών έχει φτιαχτεί με τέτοιο τρόπο ώστε αν ο χρήστης πληκτρολογήσει κάτι λάθος να του πετάει το αντίστοιχο μήνυμα αναφέροντας πως έχει κάνει λάθος και αν θέλει να ξαναπροσπαθήσει. Επίσης, αν επιχειρήσει να εισάγει νέες πληροφορίες που καταπατάνε κάποιο constraint θα εμφανιστεί το αντίστοιχο error στον χρήστη. Τέλος, ο χρήστης μπορεί από οποιοδήποτε «επίπεδο» του μενού να επιστρέψει στο προηγούμενο (με την εντολή back) ή και να βγει εντελώς από το μενού με την εντολή exit.