



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών

Υπολογιστών

Ανδρέας Ευαγγελάτος , 03118069

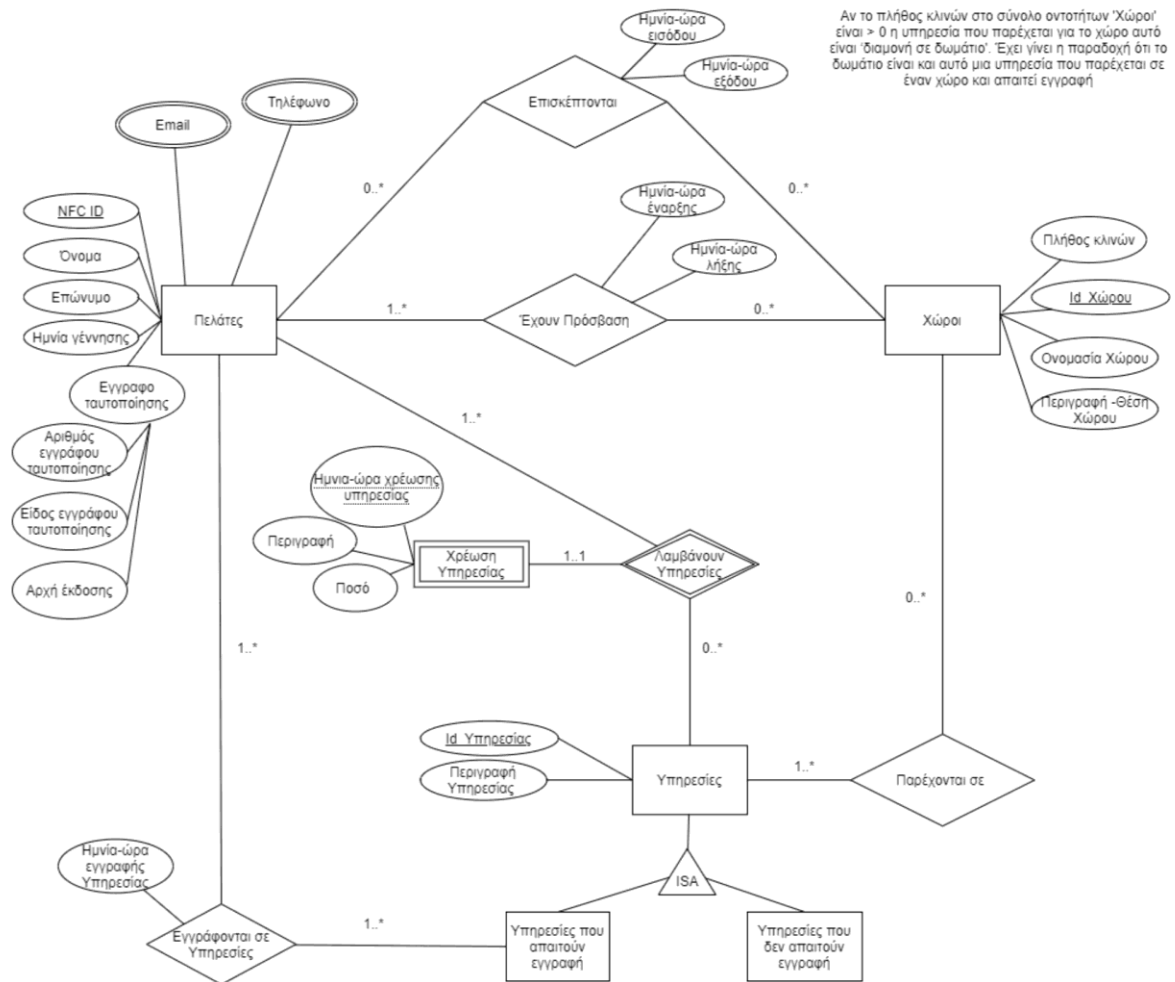
Email: el18069@mail.ntua.gr

Νικόλας Τασιόπουλος , 03118858

Email: el18858@mail.ntua.gr

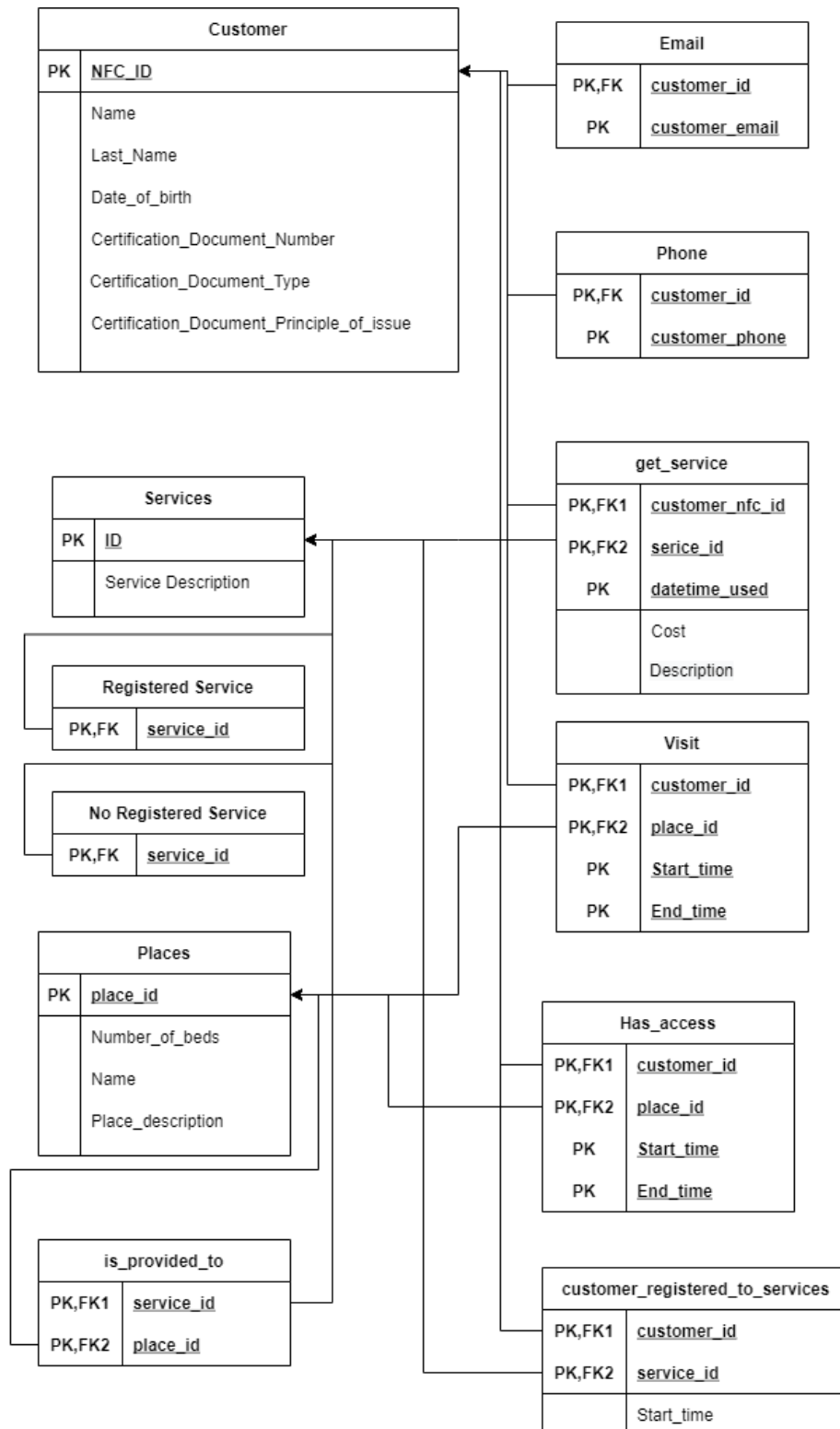
Αναφορά Εξαμηνιαίας εργασίας - 6^{ου} Εξαμήνου - Βάσεις Δεδομένων

Το ER που επιλέχθηκε ήταν το «ER-απλή λύση» που δόθηκε ως προτεινόμενη λύση.
Συγκεκριμένα το ακόλουθο :



1. Σχεσιακό Διάγραμμα

Το Σχεσιακό Διάγραμμα :



Η βασική διαφορά με το ER-Διάγραμμα είναι ότι επιλέχθηκε να απορριφθεί η μοντελοποίηση της λήψης υπηρεσίας με επιπλέον χαρακτηριστικά δηλωμένα ως ένα επιπλέον weak entity και τα επιπλέον χαρακτηριστικά προστέθηκαν ως απλά attributes στη σχέση λαμβάνουν υπηρεσία (get_service).

1.α. Οι περιορισμοί που έχουν οριστεί είναι οι ακόλουθοι:

Για τον πίνακα costumers έχει χρησιμοποιηθεί ως primary key ένας ακέραιος nfc_id ο οποίος χρησιμοποιείται ως foreign key για την συσχέτιση του πελάτη με τους πίνακες costumer_email, costumer_mobile οι οποίοι αντιπροσωπεύουν τα αντίστοιχα multivalued attributes του ER και στους costumer_registered_to_services, get_service, have_access, visit όπου εκφράζει την εγγραφή ενός πελάτη σε μια υπηρεσία , την χρήση μιας υπηρεσίας , την πρόσβαση σε ένα χώρο και την επισκέψει σε έναν χώρο αντίστοιχα.

Για τον πίνακα service, έχει επιλεγθεί πάλι ένας θετικός ακέραιος ID ο οποίος χρησιμοποιείται ως primary key και στους πίνακες registed_services και no_registered_services (κάθε ID σε ένα από τα δύο για την μοντελοποίηση της σχέσης ISA) και ως foreign key στους πίνακες costumer_registered_to_services, get_service, is_provided_to όπου εκφράζει την εγγραφή ενός πελάτη στην υπηρεσία , την λήψη της υπηρεσίας και την συσχέτιση μια υπηρεσίας σε ένα χώρο αντίστοιχα.

Για τον πίνακα places έχει επιλεγθεί και πάλι ένας θετικός ID ως primary key ,ο οποίος χρησιμοποιείται ως foreign key στους πίνακες is_provided_to, have_access, visit όπου εκφράζει την συσχέτιση του χώρου με την υπηρεσία , την πρόσβαση στον χώρο και την επίσκεψη στον χώρο αντίστοιχα.

Σύνθετα γνωρίσματα :

Ονοματεπώνυμο σε Όνομα και Επώνυμο στον πίνακα costumers

Έγγραφο ταυτοποίησης σε αριθμός και είδος εγγράφου ταυτοποίησης και αρχή έκδοσης στον πίνακα costumers.

b.

```
CREATE INDEX index_costumer ON
costumer(nfc_id, first_name, last_name);

CREATE INDEX index_get_service ON
get_service(costumer_id, service_id, datetime_used);

CREATE INDEX index_visit ON
visit(start_time, end_time);

CREATE INDEX index_place ON
places(place_id, name);
```

Η επιλογή των ευρετηρίων έγινε με σκοπό την καλύτερη αναζήτηση για τα views που θα αναφερθούμε πιο κάτω.

c.

Έγινε χρήση των παρακάτω γλωσσών προγραμματισμού : Python, SQL ,HTML ,PHP ,CSS. Συγκεκριμένα για την python πρέπει να έχουμε και τις βιβλιοθήκες numpy και pandas.

d.

Απαραίτητη προϋπόθεση για τα παρακάτω είναι η εγκατάσταση του XAMPP , η θέση του φάκελου εγκατάστασης και η εκτέλεση του XAMPP Control Panel με δικαιώματα διαχειριστή.

ΠΡΟΗΔΟΠΟΙΗΣΗ: Τα δεδομένα στο αρχείο costumer.sql έχουν παραχθεί από τους κώδικες python που θα περιληφθούν στην παράδοση.

Βήματα για την δημιουργία της βάσης:

1. Στο phpMyAdmin δημιουργούμε μια βάση με όνομα projectdb και ένα χρήστη με όνομα 'root', ΧΩΡΙΣ κωδικό και επιλέγουμε όλα τα δικαιώματα ,αν υπάρχει ήδη ο χρήστης παραλείψτε αυτό το βήμα. Αν αντιμετωπίσετε άλλο πρόβλημα πρέπει να δημιουργήσετε εσείς δικό σας χρήστη και το αλλάξετε από τα αρχεία που σας δίνουμε.
2. Εισαγωγή από το PhpMyAdmin στην βάση projectdb τα αρχεία με την σειρά που δίνονται projectDB.sql, costumer.sql .

Βήματα για το UI:

ΠΡΟΣΟΧΗ : Αν έχετε άλλο όνομα χρήστη ή άλλη βάση δεδομένων μπορείτε να αλλάξετε τα ονόματα από το αρχείο credentials.php στον ομώνυμο

1. Προσθέτουμε στον φάκελο httdocs ,του φακέλου εγκατάστασης XAMPP, τον φάκελο με όνομα DB.

Θα μπορούσατε να χρησιμοποιήσετε και έναν εναλλακτικό τρόπο ανάλογα με το λειτουργικό σύστημα της επιλογής σας και τον τοπικό server σας.

Για να αποδείξουμε ότι τα αρχεία python δουλεύουν θα μπορούσαμε να κάνουμε το εξής:

ΠΡΟΣΟΧΗ : Για να λειτουργήσει το παρακάτω πρέπει να έχουμε ήδη την βάση μας με τους πίνακες χωρίς δεδομένα και με τα ίδια ονόματα όπως στο αρχείο projectDB.sql.

Επιπλέον , πρέπει να έχουμε τις βιβλιοθήκες Python numpy και pandas, αν δεν τις έχουμε κάνει install .

Τέλος πρέπει να υπάρχουν στο ίδιο φάκελο με τα αρχεία deleteall.py , addPlaces&Services.py , addcostumers.py τα αρχεία costumers.csv places.csv & services.csv.

1. Δημιουργήστε μια βάσης με όνομα projectdb και με όνομα χρήστη root, χωρίς κωδικό , αν υπάρχει ήδη χρήστη root τότε μην κάνετε το βήμα 1.
Μπορείτε να τροποποιήσετε τα δεδομένα της βάσης σας με ανάλογη τροποποίηση στα δεδομένα του παρακάτω κώδικα σε ΟΛΑ τα scripts Python.

```
db = pymysql.connect(host='localhost',user='root',passwd='', database="projectdb")
```

2. Αν η βάση μας δεν είναι αδεια πρώτα τρέχουμε το αρχείο deleteall.py και να φτιάξουμε ξανά την βάση με το αρχείο projectDB.sql.
3. Στη συνέχεια τρέχουμε το αρχείο python addPlaces&Services.py
4. Τέλος, το αρχείο addcostumers.py

2.

Ο SQL κώδικας που γράφθηκε για την απάντηση των ερωτημάτων βρίσκεται στο αρχείο projectDB.sql .

```
CREATE TABLE costumer (  
    nfc_id int(10) AUTO_INCREMENT NOT NULL,  
    first_name varchar(50), /* First name */  
    last_name varchar(50), /* Last name */  
    date_of_birth DATE NOT NULL ,  
    certification_document_number varchar(50) NOT NULL,  
    certification_document_type varchar(50) NOT NULL,  
    certification_document_principle_of_issue varchar(50) NOT NULL,  
    PRIMARY KEY (nfc_id)  
);
```

```

/* MULTIVALUED ATTRIBUTE EMAIL */
CREATE TABLE costumer_email(
    costumer_nfc_id int(10) NOT NULL,
    costumer_email varchar(50) NOT NULL,
    FOREIGN KEY (costumer_nfc_id) REFERENCES costumer(nfc_id) ON
DELETE CASCADE ON UPDATE CASCADE ,
    PRIMARY KEY (costumer_email,costumer_nfc_id)
);

/* MULTIVALUED ATTRIBUTE MOBILE */
CREATE TABLE costumer_mobile(
    costumer_nfc_id INT(10) NOT NULL,
    costumer_mobile INT(12) NOT NULL,
    FOREIGN KEY (costumer_nfc_id) REFERENCES costumer(nfc_id) ON
DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (costumer_mobile,costumer_nfc_id)
);

/* PLACES ENTITY */
CREATE TABLE places (
    number_of_beds INT(4) ,
    place_id INT (10) NOT NULL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL ,
    place_description VARCHAR(255) NOT NULL,
    PRIMARY KEY (place_id)
);

/* HAVE ACCESS RELATIONSHIP */
CREATE TABLE have_access(
    costumer_nfc_id INT (10) NOT NULL,
    place_id INT (10) NOT NULL ,
    start_time datetime,
    end_time datetime,
    FOREIGN KEY (place_id) REFERENCES places(place_id) ON DELETE
CASCADE ON UPDATE CASCADE ,
    FOREIGN KEY (costumer_nfc_id) REFERENCES costumer(nfc_id) ON
DELETE CASCADE ON UPDATE CASCADE ,
    PRIMARY KEY (costumer_nfc_id,place_id,start_time,end_time)
);

/* VISITS RELATIONSHIP */
CREATE TABLE visit(
    costumer_nfc_id INT (10) NOT NULL ,
    place_id INT (10) NOT NULL ,
    start_time datetime,
    end_time datetime,
    FOREIGN KEY (place_id) REFERENCES places(place_id) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (costumer_nfc_id) REFERENCES costumer(nfc_id) ON
DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (costumer_nfc_id,place_id,start_time,end_time)
);

/* SERVICES ENTITY */
CREATE TABLE services (
    service_id INT(10) NOT NULL ,
    service_description VARCHAR(120),

    PRIMARY KEY (service_id)
);

```

```

/* ISA REGISTERED */
CREATE TABLE registered_services(
    service_id INT(10) NOT NULL,

    PRIMARY KEY (service_id),
    FOREIGN KEY (service_id) REFERENCES services(service_id) ON
DELETE CASCADE ON UPDATE CASCADE
);

/* ISA NOT REGISTERED */
CREATE TABLE no_registered_services(
    service_id INT(10) NOT NULL,

    PRIMARY KEY (service_id),
    FOREIGN KEY (service_id) REFERENCES services(service_id) ON
DELETE CASCADE ON UPDATE CASCADE
);

/* REGISTER TO SERVICE */
CREATE TABLE costumer_registered_to_services(
    costumer_id int(10) NOT NULL,
    service_id INT (10),
    start_time datetime NOT NULL,
    PRIMARY KEY (costumer_id,service_id),
    FOREIGN KEY (service_id) REFERENCES
registered_services(service_id) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (costumer_id) REFERENCES costumer(nfc_id) ON DELETE
CASCADE ON UPDATE CASCADE
);

/* GET SERVICE RELATIONSHIP */ /*(included SERVICE CHARGE WEAK ENTITY
*/
CREATE TABLE get_service (
    costumer_id INT(10) NOT NULL,
    service_id INT(10) NOT NULL,
    amount INT(9) NOT NULL,
    service_detailed_description VARCHAR(255),
    datetime_used DATETIME ,
    PRIMARY KEY (datetime_used,costumer_id,service_id),
    FOREIGN KEY (costumer_id) REFERENCES costumer(nfc_id) ON DELETE
CASCADE ON UPDATE CASCADE ,
    FOREIGN KEY (service_id) REFERENCES services(service_id) ON
DELETE CASCADE ON UPDATE CASCADE
);

/* IS PROVIDED TO RELATION */
CREATE TABLE is_provided_to(
    service_id int(10),
    place_id int(10),
    PRIMARY KEY (service_id,place_id),
    FOREIGN KEY (service_id) REFERENCES services(service_id),
    FOREIGN KEY (place_id) REFERENCES places(place_id)
);

CREATE INDEX index_costumer ON
costumer(nfc_id, first name, last name);

```

```

CREATE INDEX index_get_service ON
get_service(costumer_id,service_id,datetime_used);

CREATE INDEX index_visit ON
visit(start_time,end_time);

CREATE INDEX index_place ON
places(place_id,name);

CREATE VIEW erwthma7 AS
SELECT
first_name,last_name,nfc_id,services.service_id,service_description,a
mount,datetime_used
FROM costumer , get_service , services
WHERE costumer.nfc_id=get_service.costumer_id AND
get_service.service_id=services.service_id ;

CREATE VIEW erwthma9 AS
SELECT
nfc_id,last_name,first_name,places.place_id,name,start_time,end_time
FROM costumer , visit , places
WHERE costumer.nfc_id=visit.costumer_nfc_id AND
visit.place_id=places.place_id;

CREATE VIEW erwthma10 AS
SELECT ER_9_1.nfc_id , ER_9_2.nfc_id as
no_covid_id,ER_9_2.first_name,ER_9_2.last_name,ER_9_2.name ,
ER_9_2.place_id ,ER_9_2.start_time AS non_covid_start_time ,
ER_9_1.start_time , ER_9_1.end_time , ER_9_2.end_time AS
non_covid_end_time
FROM erwthma9 as ER_9_1 ,erwthma9 as ER_9_2
WHERE ER_9_2.place_id=ER_9_1.place_id AND
ER_9_1.nfc_id<>ER_9_2.nfc_id AND (
(ER_9_2.start_time<=ER_9_1.start_time AND
ER_9_2.end_time<=DATE_ADD(ER_9_1.end_time, INTERVAL 1 HOUR) AND
ER_9_2.end_time>=ER_9_1.start_time) OR
(ER_9_2.start_time>=ER_9_1.start_time AND
ER_9_2.end_time<=DATE_ADD(ER_9_1.end_time, INTERVAL 1 HOUR) ) OR
(ER_9_2.start_time<=ER_9_1.end_time AND
ER_9_2.end_time>=DATE_ADD(ER_9_1.end_time, INTERVAL 1 HOUR))
) ;

CREATE VIEW erwthma8a AS
SELECT SUM(amount) AS sum_amount,get_service.service_id,
services.service_description
FROM get_service,services
WHERE services.service_id=get_service.service_id
GROUP BY service_id ASC ;

CREATE VIEW erwthma8b AS
SELECT nfc_id, first_name, last_name, date_of_birth,
certification_document_number,certification_document_type,
certification_document_principle_of_issue, costumer_mobile,
costumer_email
FROM costumer, costumer_mobile ,costumer_email
WHERE costumer.nfc_id=costumer_mobile.costumer_nfc_id and
costumer.nfc_id=costumer_email.costumer_nfc_id
ORDER BY nfc_id ASC;

```


3.

Τα αρχεία περιλαμβάνονται στην αναφορά, πιο αναλυτικά:

SQL και στον υπό-φάκελο add είναι τα δεδομένα για το φτιάξιμο της βάσης και τα δεδομένα σε αυτή αντίστοιχα σε κάθε έναν φάκελο.

Το UI είναι διαμοιρασμένο με όλους τους υπόλοιπους φακέλους.