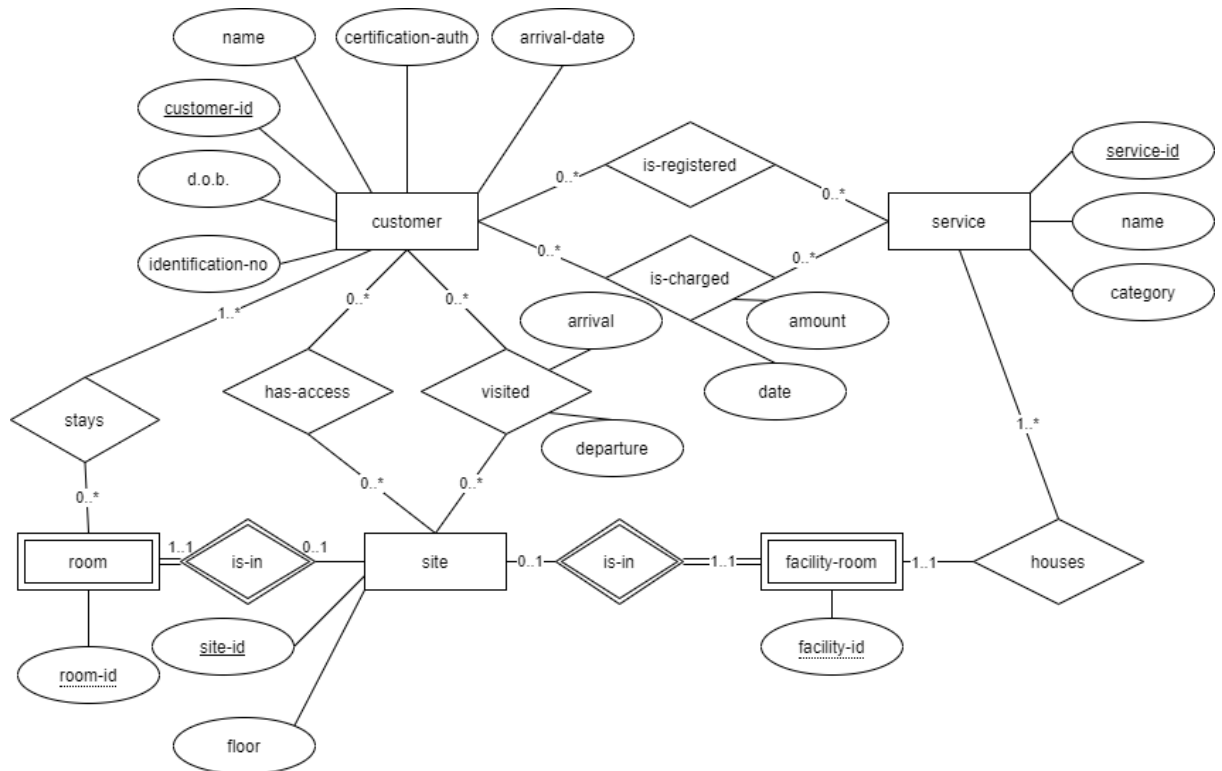
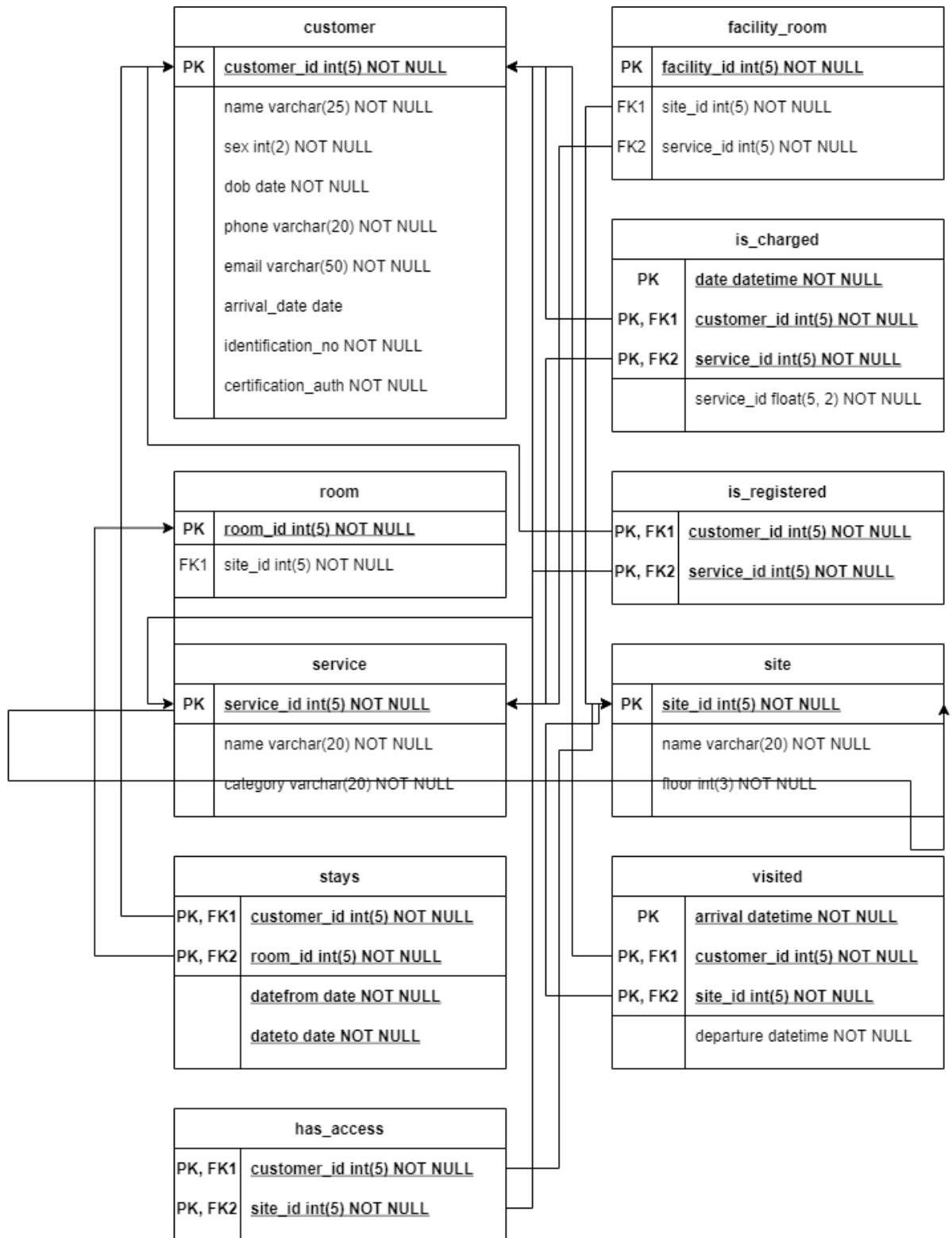


Αναφορά εξαμηνιαίας εργασίας  
Βάσεις Δεδομένων  
Έτος 2020-2021

Αστρινάκης Νικόλαος el17008  
Ταμπακάκης Χρήστος el17042

1. ER και σχεσιακό διάγραμμα (περιέχονται και σε χωριστά pdf):





A. Όλοι οι περιορισμοί που έχουν οριστεί ανά κατηγορία:

a. Domain Constraints:

i. customer

1. customer\_id int(5) Ακέραιος με αρκετά μεγάλο μέγεθος για να έχουμε περιθώριο στη βάση μας
2. name varchar(25)

3. sex int(1) κωδικοποιούμε τα φύλα με αριθμητικό κωδικό (1, 2, 3)
4. dob date
5. phone varchar(20) περιθώριο για σημεία στίξης ανάλογα με τη χώρα
6. email varchar(50) για να χωράει τις αυτόματα generated διευθύνσεις από lorem ipsum
7. arrival\_date date
8. identification\_no varchar(7)
9. certification\_auth varchar(25)
- ii. service
  1. service\_id int(5)
  2. name varchar(20)
  3. category varchar(25)
- iii. site
  1. site\_id int(5)
  2. name varchar(30)
  3. floor int(3)
- iv. room
  1. room\_id int(5)
  2. site\_id int(5)
- v. facility\_room
  1. facility\_id int(5)
  2. site\_id int(5)
  3. service\_id int(5)
- vi. has\_access
  1. customer\_id int(5)
  2. site\_id int(5)
- vii. is\_registered
  1. customer\_id int(5)
  2. service\_id int(5)
- viii. is\_charged
  1. customer\_id int(5)
  2. service\_id int(5)
  3. amount float(5,2)
  4. date datetime
- ix. stays
  1. customer\_id int(5)
  2. room\_id int(5)
  3. datefrom date
  4. dateto date
- x. visited
  1. customer\_id int(5)
  2. site\_id int(5)
  3. arrival datetime
  4. departure datetime
- b. Key constraints
  - i. customer: Primary key (customer\_id)

- ii. service: Primary key (service\_id)
  - iii. site: Primary key (site\_id)
  - iv. room: Primary key (room\_id)
  - v. facility\_room: Primary key (facility\_id)
  - vi. has\_access: Primary key (customer\_id, site\_id)
  - vii. is\_registered: Primary key (customer\_id, service\_id)
  - viii. is\_charged: Primary key (customer\_id, service\_id, date)
  - ix. stays: Primary key (customer\_id, room\_id)
  - x. visited: Primary key (customer\_id, site\_id, arrival)
- c. Integrity constraints
- i. customer: Όλες οι τιμές NOT NULL εκτός από το arrival\_date, διότι θέλουμε να έχουμε όλα τα στοιχεία των πελατών στη βάση μας. Η τιμή arrival\_date εξαιρείται διότι μπορεί να μην έχει φτάσει ακόμα ο πελάτης
  - ii. service: Όλες οι τιμές NOT NULL και το name UNIQUE
  - iii. site: Όλες οι τιμές NOT NULL
  - iv. room: Όλες οι τιμές NOT NULL
  - v. facility\_room: Όλες οι τιμές NOT NULL εκτός του service\_id (με την λογική ότι ένα δωμάτιο θα παραμείνει facility room ακόμα και άμα σβήσουμε την υπηρεσία που φιλοξενούσε)
  - vi. has\_access: Όλες οι τιμές NOT NULL (όλες μέρος του primary key)
  - vii. is\_registered: Όλες οι τιμές NOT NULL (όλες μέρος του primary key)
  - viii. is\_charged: Όλες οι τιμές NOT NULL, διότι οι 3 είναι μέρος του primary key και η τιμή amount είναι η κρίσιμη πληροφορία που θέλουμε να αποθηκεύσουμε
  - ix. stays: Όλες οι τιμές NOT NULL, customer\_id και room\_id είναι το primary key και η ημερομηνίες είναι απαραίτητες πληροφορίες στη βάση
  - x. visited: Όλες οι τιμές NOT NULL διότι είναι μέρος του primary key, ενώ το departure μπορεί να γίνει NULL, δείχνοντας ότι ο πελάτης είναι ακόμα εκεί
- d. Referential integrity constraints
- i. room:
    - 1. site\_id αναφέρεται στην αντίστοιχη στήλη στον πίνακα site με συμπεριφορά ON UPDATE CASCADE ON DELETE CASCADE  
Ο λόγος που επιλέχθηκε το CASCADE είναι διότι σε αυτό τον πίνακα όπως και στις περισσότερες παρακάτω περιπτώσεις, το foreign key δε μπορεί να γίνει null, οπότε προτιμάμε να αφήσουμε να γίνονται τα operations με CASCADE αντί να κάνουμε RESTRICT, εφόσον το SET NULL δεν είναι διαθέσιμο
  - ii. facility\_room:
    - 1. service\_id αναφέρεται στην αντίστοιχη στήλη του πίνακα service ON UPDATE CASCADE ON DELETE SET NULL  
Ιδανικά θα θέλαμε ένα facility room να παραμείνει παρότι δεν ανήκει πλέον σε μια συγκεκριμένη υπηρεσία, διότι είναι πιθανό στο μέλλον να στεγάσει μια διαφορετική υπηρεσία.

2. `site_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `site` ON UPDATE CASCADE ON DELETE CASCADE
  - iii. `has_access`:
    1. `customer_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `customer` ON UPDATE CASCADE ON DELETE CASCADE
    2. `site_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `site` ON UPDATE CASCADE ON DELETE CASCADE
  - iv. `is_registered`:
    1. `service_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `service` ON UPDATE CASCADE ON DELETE CASCADE
    2. `customer_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `customer` ON UPDATE CASCADE ON DELETE CASCADE
  - v. `is_charged`:
    1. `service_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `service` ON UPDATE CASCADE ON DELETE CASCADE
    2. `customer_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `customer` ON UPDATE CASCADE ON DELETE CASCADE
  - vi. `stays`:
    1. `room_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `service` ON UPDATE RESTRICT ON DELETE RESTRICT  
 Ο λόγος που στην συγκεκριμένη περίπτωση επιλέχθηκε το RESTRICT είναι διότι στην περίπτωση που για οποιοδήποτε λόγο πρέπει να σβηστεί κάποιο δωμάτιο, δε θέλουμε να μπορέσουμε να το σβήσουμε πριν να μπορέσουμε να αποκαταστήσουμε όλους τους πελάτες που έχουν δηλώσει ότι θα μείνουν εκεί ή μένουν ήδη εκεί. Θέλουμε λοιπόν να έχουμε ήδη διευθετήσει αυτά τα ζητήματα και μετά να μπορούμε να κάνουμε τη διαγραφή.
    2. `customer_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `customer` ON UPDATE CASCADE ON DELETE CASCADE
  - vii. `visited`:
    1. `site_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `site` ON UPDATE CASCADE ON DELETE CASCADE
    2. `customer_id` αναφέρεται στην αντίστοιχη στήλη του πίνακα `customer` ON UPDATE CASCADE ON DELETE CASCADE
- e. User-defined constraints

Στη σχέση *visited* προσθέτουμε έναν περιορισμό ώστε το *arrival* να είναι μικρότερο του *departure* για να μπορούμε να τα χρησιμοποιήσουμε στην αναζήτηση μολυσμένων χώρων και ανθρώπων χωρίς να χρειάζεται να ελέγχουμε την ορθότητά τους. Να σημειωθεί ότι ο έλεγχος επιτυγχάνει ακόμα και άμα το *departure* είναι NULL:

```
ALTER TABLE visited
ADD CONSTRAINT CHK_TIME
CHECK (arrival < departure)
```

Προσθέσαμε επίσης τον ακόλουθο περιορισμό για να βεβαιωθούμε ότι το φύλο του κάθε πελάτη έχει έγκυρη τιμή (1 για male, 2 για female και 3 για other)

```
ALTER TABLE customer
ADD CONSTRAINT gender
CHECK (sex >= 1 AND sex <= 3)
```

- B. Πέρα από τα ευρήματα με βάση τα primary και foreign keys που δημιουργούνται αυτόματα, δημιουργήσαμε 3 επιπλέον:
- Στον πίνακα *customer* δημιουργήσαμε index με βάση τη στήλη *dob* (date of birth) καθώς υπάρχουν 6 ερωτήματα τα οποία απαιτούν να χωρίσουμε τους πελάτες ανάλογα με ηλικιακή ομάδα, και θεωρήσαμε ότι ο ορισμός του index θα βοηθήσει στην επιτάχυνση των ερωτημάτων αυτών.
  - Στον πίνακα *visited* δημιουργήσαμε index με βάση τη στήλη *arrival* (η οποία ήταν ήδη μέρος του composite key, άρα μέρος ενός index) καθώς και ένα ακόμα με βάση τη στήλη *departure*. Αυτή η επιλογή έγινε διότι στην αναζήτηση για ασθενείς από covid έρχονται πολλές φορές σε σύγκριση μεταξύ τους αυτές οι τιμές για να μπορέσουμε να βρούμε αν συνυπήρξαν στον ίδιο χώρο δύο άτομα. Συνεπώς ιδιαίτερα για ένα τόσο υπολογιστικά εντατικό query θεωρήσαμε τα οφέλη θα είναι σημαντικά.
- C. Η βάση δημιουργήθηκε σε περιβάλλον mariaDB μέσω του phpmyadmin και xampp. Για την επικοινωνία με τη βάση δημιουργήσαμε ένα back-end και για την εμφάνιση των αποτελεσμάτων ένα front-end. Και τα δύο γράφτηκαν σε Javascript, το back-end χρησιμοποιώντας nodemon και express, ενώ το front-end σε react.
- D. Για να εγκαταστήσει κάποιος την εφαρμογή μας πρέπει να ακολουθήσει τα εξής βήματα:
- Εγκατάσταση node js στο σύστημά του.
  - Χρήση της εντολής npm install μέσα στο φάκελο back-end καθώς και στον φάκελο front-end
  - Χρήση mariaDB μέσω xampp στο σύστημά του
  - Δημιουργία βάσης με όνομα hotel
  - Δημιουργία χρήστη για την παραπάνω βάση με δικαιώματα δεδομένων πάνω στη βάση
  - Επεξεργασία του αρχείου .env τόσο στον φάκελο back-end όσο και στον front-end. Συγκεκριμένα για το back-end:
    - Ορισμός του port του db\_port (default 3306)
    - Ορισμός του service\_port (η θύρα στην οποία θα ακούει το back-end, default 4001)
    - Ορισμός του db\_username (default hotelcalifornia)
    - Ορισμός του db\_password (default suchalovelyplace)
  - Για το front-end:
    - Ορισμός του REACT\_APP\_BACKEND\_PORT (πρέπει να ταυτίζεται με το service\_port του backend, default 4001)

- Ορισμός του PORT (η θύρα στην οποία θα ακούει το frontend, default 3000)
- Τελευταίο βήμα είναι το import του database dump hotel.sql μέσα στην αντίστοιχη βάση
- Η εφαρμογή είναι έτοιμη για τρέξιμο. Αρκεί να τρέξει κάποιος npm start μέσα στον φάκελο back-end και μέσα στον φάκελο front-end

2. Όλος ο κώδικας SQL που γράφτηκε για τη βάση περιέχεται στα αρχεία queries.sql και hotel.sql που συμπεριλαμβάνονται με την αναφορά. Περιέχουν αντίστοιχα όλα τα queries που χρειάζονται για την επίλυση των ερωτημάτων της εργασίας, και όλα την πληροφορία για τη δομή και τα δεδομένα της βάσης.

3. Τα αρχεία διαμόρφωσης βρίσκονται μαζί με την αναφορά στους φακέλους back-end και front-end