Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ροή Λ: Βάσεις Δεδομένων (6ο Εξάμηνο)

Αναφορά Project 2021

Συμμετέχοντες:

Δημήτρης Καλαθάς (el18016)

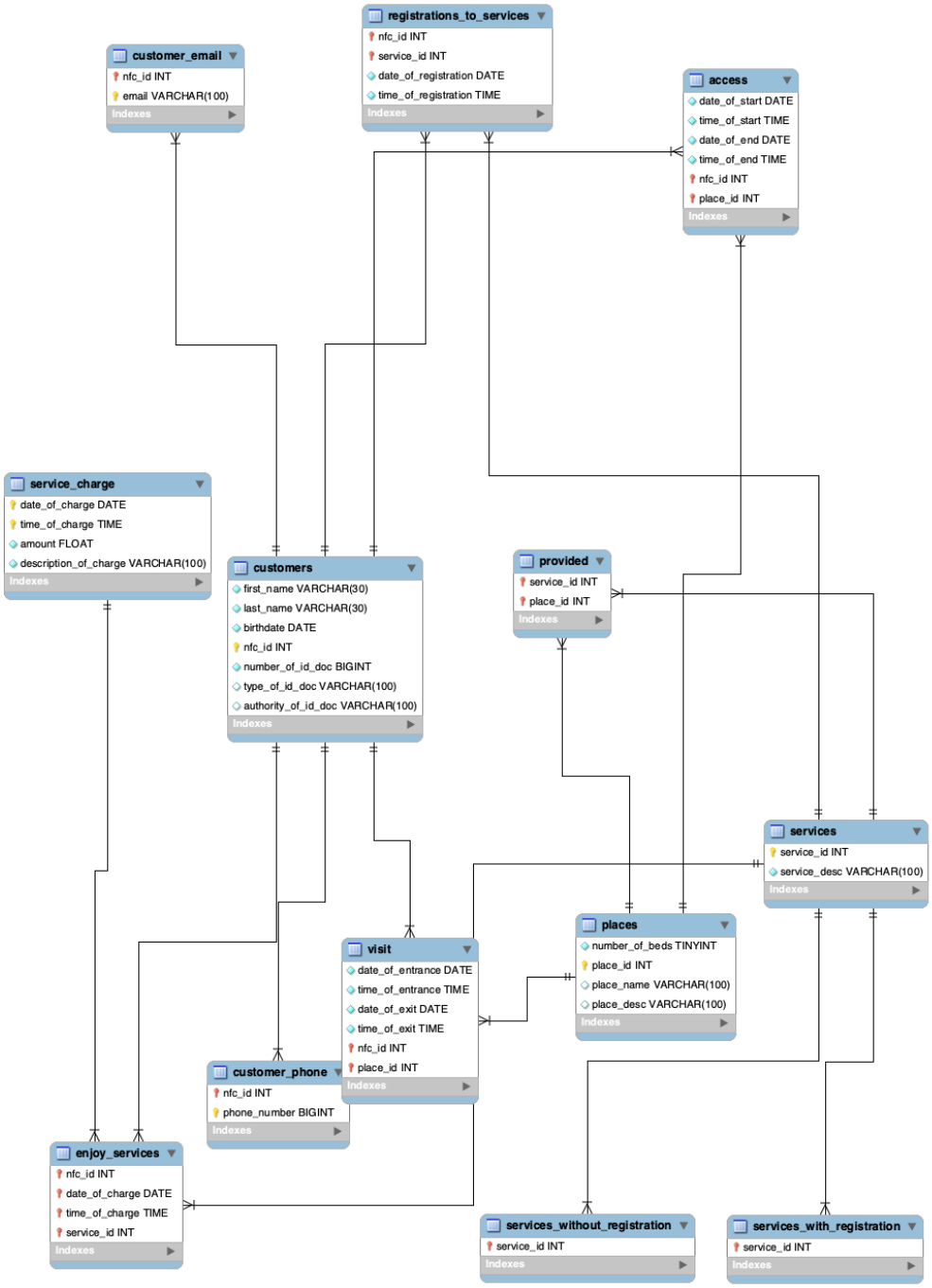
Δημήτρης Μπακάλης (el18163)

Γρηγόρης Παπανικολάου (el18649)

Περιεχόμενα

1. Σχεσιακό Διάγραμμα

Παρακάτω δίνεται το σχεσιακό διάγραμμα της ΒΔ:



A. Σύμφωνα και με το ER, που δόθηκε ως απλή λύση, χρησιμοποιήσαμε τις παρακάτω εντολές για τη δημιουργία των tables:

```
create database COVID;

create table customers
(
first_name varchar(30) not null,
last_name varchar(30) not null,
birthdate date not null,
nfc_id int not null auto_increment,
number_of_id_doc bigint(100) not null,
type_of_id_doc varchar(100),
authority_of_id_doc varchar(100),

constraint PKcustomers primary key(nfc_id)
);

create table places
(
number_of_beds tinyint not null,
place_id int not null,
place_name varchar(100),
place_desc varchar(100),

constraint PKplaces primary key (place_id)
);

create table visit
(
date_of_entrance date not null,
time_of_entrance time not null,
date_of_exit date not null,
time_of_exit time not null,
nfc_id int not null,
place_id int not null,

constraint PKvisit primary key (nfc_id,place_id),
constraint FKcustomers foreign key (nfc_id) references
customers(nfc_id) on update cascade on delete cascade,
constraint FKplaces foreign key (place_id) references
places(place_id) on update cascade on delete cascade

);

create table access
(
date_of_start date not null,
time_of_start time not null,
date_of_end date not null,
time_of_end time not null,
nfc_id int not null,
place_id int not null,

constraint PKaccess primary key (nfc_id,place_id),
constraint FK1customers foreign key (nfc_id) references
customers(nfc_id) on update cascade on delete cascade,
```

```
constraint FK1places foreign key (place_id) references
places(place_id) on update cascade on delete cascade

);

create table services
(
service_id int not null,
service_desc varchar(100) not null,

constraint PKservices primary key (service_id)
);

create table service_charge
(
date_of_charge date not null,
time_of_charge time not null,
amount float not null,
description_of_charge varchar(100) not null,

constraint PKservice_charge primary key
(date_of_charge,time_of_charge)

);

create table enjoy_services
(
nfc_id int not null,
date_of_charge date not null,
time_of_charge time not null,
service_id int not null,

constraint PKenjoy_services primary key
(nfc_id,date_of_charge,time_of_charge,service_id),
constraint FK2customers foreign key (nfc_id) references
customers(nfc_id) on update cascade on delete cascade,
constraint FKservice_charge foreign key
(date_of_charge,time_of_charge) references
service_charge(date_of_charge,time_of_charge) on update cascade on
delete cascade,
constraint FKservices foreign key (service_id) references
services(service_id) on update cascade on delete cascade

);

create table provided
(
service_id int not null,
place_id int not null,

constraint PKprovided primary key (service_id,place_id),
constraint FK1services foreign key (service_id) references
services(service_id) on update cascade on delete cascade,
constraint FK2places foreign key (place_id) references
places(place_id) on update cascade on delete cascade

);

create table services_with_registration
(
service_id int not null,
```

```
constraint PKservices_with_registration primary key (service_id),
constraint FK2services foreign key (service_id) references
services(service_id) on update cascade on delete cascade
);

create table services_without_registration
(
service_id int not null,

constraint PKservices_without_registration primary key (service_id),
constraint FK3services foreign key (service_id) references
services(service_id) on update cascade on delete cascade
);

create table registrations_to_services
(
nfc_id int not null,
service_id int not null,
date_of_registration date not null,
time_of_registration time not null,

constraint PKregistrations_to_services primary
key(nfc_id,service_id),
constraint FK4services foreign key (service_id) references
services(service_id) on update cascade on delete cascade,
constraint FK3customers foreign key (nfc_id) references
customers(nfc_id) on update cascade on delete cascade

);

create table customer_phone
(
        nfc_id int not null,
        phone_number bigint not null,
        constraint PKcustomer_phone primary key (nfc_id,
phone_number),
        constraint FKcustomer_phone foreign key (nfc_id) references
customers(nfc_id) on update cascade on delete cascade
);

create table customer_email
(
        nfc_id int not null,
        email varchar(100) not null,
        constraint PKcustomer_email primary key (nfc_id, email),
        constraint FKcustomer_email foreign key (nfc_id) references
customers(nfc_id) on update cascade on delete cascade
);
```

Ο πίνακας **customers**, συνδέεται μέσω των relationship sets, **visit** (αποθήκευση ώρας εισόδου-εξόδου) και **access**(αποθήκευση ώρας εισόδου-εξόδου) με τον πίνακα **places**, ενώ, παράλληλα, συνδέεται μέσω του relationship set **enjoy_services** (επιπλέον αποθήκευση χρόνου, περιγραφής, ποσού και ημερομηνίας χρέωσης από το weak entity **service_charge**) με το table **services**. Ακόμα, μέσω του relationship set **registrations_to_services** (επιπλέον αποθήκευση χρόνου εγγραφής) συνδέεται με το table των υπηρεσιών, που χρειάζονται εγγραφή. Το table **customers**, επιπλέον, συνδέεται με τα tables, που περιέχουν τα **τηλέφωνα** και τα **email,** τα οποία θεωρήσαμε **multivalued attributes**, καθώς σε ένα πελάτη μπορούν να αντιστοιχούν περισσότερα από ένα στοιχεία επικοινωνίας ή και κανένα. Σε όλα τα παραπάνω, οι συνδέσεις χρησιμοποιούν το primary key του **customers**, δηλαδή το nfc_id, το οποίο είναι τύπου integer (με τέσσερα ψηφία, όχι όμως υποχρεωτικά), ενώ η ημερομηνία γέννησης είναι δεδομένο τύπου date. Επίσης, τα υπόλοιπα δεδομένα, που περιέχει αποτελούνται, μόνο από χαρακτήρες και έχουν μεταβλητό μέγεθος (έως 30 ή 100).

Στη συνέχεια, θα αναφερθούμε στο table **services**, το οποίο κληροδοτεί τα service_id στο επιμέρους **services_with_registration** το οποίο περιέχει μόνο υπηρεσίες με εγγραφή, καθώς και το **services_without_registration**, το οποίο θα περιέχει μόνο υπηρεσίες χωρίς εγγραφή. Τα παραπάνω tables έχουν ως primary key το service_id (μονοψήφιο αναγνωριστικό) με μόνη διαφορά ότι στο **services** υπάρχει και ένα attribute με χαρακτήρες το οποίο δίνει την περιγραφή της υπηρεσίας. Ενώ στις υπηρεσίες έχει συμπεριληφθεί και το hotel

accommodation.

Έπειτα, θα αναφερθούμε στο τελευταίο «ισχυρό» entity set που απέμεινε, το table **places** το οποίο περιέχει τα 400 δωμάτια, που δημιουργήσαμε, τους χώρους προσφοράς υπηρεσιών, καθώς και τα μοναδικά αναγνωριστικά τους place_id (integer). Επίσης έχει συμπεριληφθεί, η πτέρυγα μέσα στην οποία βρίσκονται, όπου κρίνεται αναγκαίο, ως place_desc (char), ο αριθμός κρεβατιών (στους χώρους παροχής υπηρεσιών είναι 0, ενώ στα δωμάτια των επισκεπτών, ποικίλει με τυχαίο τρόπο από 1-4) και η ονομασία του χώρου (char). Ενώ το **places**, συνδέεται με το table **services** μέσω του relationship set **provided**. Σχετικά με τον όροφο, στον οποίο ανήκει κάθε δωμάτιο, έχει γίνει η παραδοχή πως τα πρώτα 80 ανήκουν στον 1^{ο} όροφο, τα επόμενα 80 στον 2^{ο} όροφο (400 δωμάτια με 5 ορόφους) κλπ.

Β. Τα ευρετήρια, που δημιουργήθηκαν, είναι τα εξής:

```
CREATE INDEX popular_places ON visit(place_id);
CREATE INDEX most_used_services ON service_charge(description_of_charge);
CREATE INDEX popular_services ON enjoy_services(nfc_id);
```

Τα παραπάνω ευρετήρια δημιουργήθηκαν με τρόπο, τέτοιο ώστε, να χρησιμοποιηθούν στο τελευταίο ερώτημα, προκειμένου να έχουμε καλύτερη απόδοση του προγράμματος από άποψη χρόνου.

Γ. Το λογισμικό που είναι ορατό και προσβάσιμο από τον χρήστη είναι η ιστοσελίδα, για τη δημιουργία της οποίας, έχει χρησιμοποιηθεί HTML,CSS και JavaScript . Για το κομμάτι της δημιουργίας της διεπαφής έχει χρησιμοποιηθεί

PHP και SQL για τη δημιουργία αιτημάτων, για δεδομένα από τη βάση. Ενώ, για τον σχηματισμό της βάσης χρησιμοποιήθηκε η πλατφόρμα phpMyAdmin. Για την εισαγωγή τυχαίων δεδομένων στη βάση χρησιμοποιήθηκε script γραμμένο σε Python με τυχαίες τιμές που κυμαίνονται σε συγκεκριμένα όρια, προκειμένου να έχουμε καλύτερα αποτελέσματα από τα αιτήματα (επισυνάπτεται στο τελικό zip αρχείο), καθώς και ιστοσελίδα παραγωγής τυχαίων δεδομένων, της οποίας ο σύνδεσμος δίνεται παρακάτω:

https://extendsclass.com/csv-generator.html

Δ. Για την εγκατάσταση είναι ανάγκη, μέσω XAMPP, να γίνει χρήση του server της Apache και να συνδεθούμε στην πλατφόρμα phpMyAdmin, ενώ, στη συνέχεια, θα πρέπει να δημιουργηθεί η βάση μέσω του κώδικα σε SQL. Έπειτα, θα εισαχθούν τα δεδομένα από τα έτοιμα CSV που αποστέλλουμε και δημιουργήθηκαν από γεννήτρια, ενώ, παράλληλα, είναι ανάγκη να τρέξει και το script το οποίο συνδέεται με τη βάση και επίσης την γεμίζει με στοιχεία. Επίσης είναι σημαντικό να δημιουργηθούν τα indexes και τα views με τον κώδικα, που δίνεται μέσω της πλατφόρμας phpMyAdmin. Τέλος, είναι ανάγκη τα αρχεία σε HTML,CSS,JavaScript και PHP να βρίσκονται στον ίδιο φάκελο htdocs του XAMPP. Στη συνέχεια, ανοίγουμε με οποιοδήποτε browser (συστήνεται Chrome), όποιο αρχείο HTML επιθυμούμε και μπορούμε να περιηγηθούμε σ’ όλες τις επιμέρους ιστοσελίδες και να δούμε τα ερωτήματα της άσκησης.

2.Κώδικας SQL

Ερώτημα 7

Ανάλογα με τα κριτήρια που θέτει ο χρήστης, η PHP φτάνει στο αντίστοιχο από τα παρακάτω αιτήματα σε SQL. Πιο συγκεκριμένα, παρακάτω επισυνάπτονται οι 6 κώδικες σε sql, που καλύπτουν όλες τις περιπτώσεις της εισόδου:

Όταν έχουμε ως είσοδο μόνο την ημερομηνία:

```
"SELECT nfc_id,first_name,last_name,place_id,place_name
        from customers,places
        where (nfc_id,place_id) in
        (SELECT nfc_id, place_id from visit where date_of_entrance = '$date'
)"
```

Όταν έχουμε ως είσοδο μόνο την υπηρεσία:

```
 "SELECT nfc_id,first_name,last_name,place_id,place_name
        from customers,places
        where (nfc_id,place_id) in
        (SELECT nfc_id, place_id from visit  where place_id in (SELECT place
_id  from provided  where service_id in (SELECT service_id from services whe
re service_desc = '$services') ))"
```

Όταν έχουμε ως είσοδο μόνο την τιμή:

```
 "SELECT nfc_id,first_name,last_name,place_id,place_name
        from customers,places
        where (nfc_id,place_id) in
        (SELECT nfc_id, place_id

        from visit

        where nfc_id in

        (select nfc_id
```

```
        from enjoy_services

        where (date_of_charge, time_of_charge) in (select date_of_charge, ti
me_of_charge

              from service_charge

              where amount ='$price')

        )) "
```

Όταν έχουμε ως είσοδο την ημερομηνία και την υπηρεσία:

```
"SELECT nfc_id,first_name,last_name,place_id,place_name
        from customers,places
        where (nfc_id,place_id) in
        (SELECT nfc_id, place_id

        from visit

        where date_of_entrance = '$date' and place_id in (select place_id

          from provided

         where service_id in

          (select service_id

           from services

           where service_desc = '$services')

        ) )"
```

Όταν έχουμε ως είσοδο την υπηρεσία και την τιμή:

```
"SELECT nfc_id,first_name,last_name,place_id,place_name
        from customers,places
        where (nfc_id,place_id) in
        (SELECT nfc_id, place_id

        from visit

        where place_id in (select place_id
```

```
        from provided

        where service_id in (select service_id

           from services

           where service_desc = '$services')

        )

        and

        nfc_id in

        (select nfc_id

        from enjoy_services

        where (date_of_charge, time_of_charge) in (select date_of_charge, ti
me_of_charge

              from service_charge

              where amount = '$price')

        ) ) "
```

Όταν έχουμε ως είσοδο την ημερομηνία και την τιμή:

```
"SELECT nfc_id,first_name,last_name,place_id,place_name
        from customers,places
        where (nfc_id,place_id) in
        (SELECT nfc_id, place_id

        from visit

        where date_of_entrance = '$date' and nfc_id in

        (select nfc_id

        from enjoy_services

        where (date_of_charge, time_of_charge) in (select date_of_charge, ti
me_of_charge

              from service_charge
```

```
            where amount = '$price')

        ) )"
```

Όταν έχουμε ως είσοδο και τις 3 πιθανές εισόδους:

```
"SELECT nfc_id,first_name,last_name,place_id,place_name
        from customers,places
        where (nfc_id,place_id) in
        (select nfc_id, place_id

        from visit

        where date_of_entrance = '$date'

        and

        place_id in (select place_id

        from provided

        where service_id in (select service_id from services

        where service_desc = '$services') )

        and nfc_id in

        (select nfc_id  from enjoy_services where (date_of_charge, time_of_c
harge)
         in (select date_of_charge, time_of_charge from service_charge where
 amount = '$price') )
        ) "
```

Ερώτημα 8

Παρακάτω δίνεται ο κώδικας σχηματισμού των όψεων:

```
CREATE VIEW customer_info AS
SELECT
  DISTINCT c.first_name,
  c.last_name,
  c.birthdate,
  c.nfc_id,
  c.number_of_id_doc,
  c.type_of_id_doc,
  c.authority_of_id_doc,
  p.phone_number,
  e.email
FROM
  customer c
  LEFT JOIN customer_phone p on c.nfc_id = p.nfc_id
  LEFT JOIN customer_email e on p.nfc_id = e.nfc_id
CREATE VIEW sales_info AS
SELECT
  DISTINCT SUM(s.amount),
  s.description_of_charge
FROM
  service_charge as s
GROUP by
  2
ORDER BY
  1 DESC
```

Στη συνέχεια, δίνεται ο κώδικας SQL, που ενεργοποιείται κάθε φορά από την PHP, ανάλογα με τις επιλογές του χρήστη.

```
"SELECT * from sales_info"
"SELECT * from customer_info"
```

Ερώτημα 9

Στην περίπτωση κρούσματος, θα ενεργοποιηθεί, για συγκεκριμένο nfc_id, το παρακάτω αίτημα, για να δούμε αν έχει επισκεφτεί κάποιο χώρο ο ασθενής.

```
"SELECT * FROM visit WHERE nfc_id='$nfc_id'"
```

Ερώτημα 10

Για να βρούμε τους πελάτες με τους οποίους μπορεί να έχει έρθει σ' επαφή ο ασθενής, θα εκτελεστεί το παρακάτω αίτημα σε SQL μετά από επιλογή του χρήστη.

```
"SELECT DISTINCT v.nfc_id,v.date_of_entrance,v.time_of_entrance,v.time_of_exit,v.place_id
        FROM visit as v, visit as s
        WHERE ((v.time_of_entrance <TIMESTAMPADD(HOUR,1,s.time_of_exit) and v.time_of_entrance>s.time_of_entrance) OR (v.time_of_exit <TIMESTAMPADD(HOUR,1,s.time_of_exit) and v.time_of_exit>s.time_of_entrance) )
        AND v.place_id=s.place_id and v.date_of_entrance=s.date_of_entrance and v.nfc_id <>s.nfc_id
        and s.nfc_id='$nfc_id'"
```

Ερώτημα 11

Τέλος, ο χρήστης θα πρέπει να αποφασίσει για ποια ηλικιακή ομάδα ενδιαφέρεται να μάθει κάθε φορά, για ποια χρονική περίοδο και για ποια κατηγορία, ώστε να λάβει αποτελέσματα δημοτικότητας, με φθίνουσα σειρά. Ανάλογα με τις επιλογές του χρήστη, η PHP θα ενεργοποιεί κάποιο από τα παρακάτω αιτήματα:

Για τους πιο πολυσύχναστους χώρους:

Για το έτος 2021 και ηλικιακή ομάδα 20-40:

```
"SELECT DISTINCT place_id,COUNT(place_id) as number_of_visits
        from visit

        where (nfc_id in (select nfc_id

        from customers

        where(birthdate<='2001-01-01' and birthdate>='1981-01-01'))

        and

        date_of_entrance>= '2021-01-01')
        GROUP by place_id
        ORDER by number_of_visits DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 20-40:

```
"SELECT DISTINCT place_id,COUNT(place_id) as number_of_visits
        from visit

        where (nfc_id in (select nfc_id

        from customers

        where(birthdate<='2001-01-01' and birthdate>='1981-01-01'))

        and

        date_of_entrance>= '2021-12-01')
        GROUP by place_id
        ORDER by number_of_visits DESC"
```

Για το έτος 2021 και ηλικιακή ομάδα 41-60:

```
"SELECT DISTINCT place_id,COUNT(place_id) as number_of_visits
        from visit

        where (nfc_id in (select nfc_id

        from customers
```

```
        where(birthdate<='1980-01-01' and birthdate>='1961-01-01'))

        and

        date_of_entrance>= '2021-01-01')
        GROUP by place_id
        ORDER by number_of_visits DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 41-60:

```
"SELECT DISTINCT place_id,COUNT(place_id) as number_of_visits
        from visit

        where (nfc_id in (select nfc_id

        from customers

        where(birthdate<='1980-01-01' and birthdate>='1961-01-01'))

        and

        date_of_entrance>= '2021-12-01')
        GROUP by place_id
        ORDER by number_of_visits DESC"
```

Για το έτος 2021 και ηλικιακή ομάδα 61+:

```
"SELECT DISTINCT place_id,COUNT(place_id) as number_of_visits
        from visit

        where (nfc_id in (select nfc_id

        from customers

        where(birthdate<='1960-01-01'))

        and

        date_of_entrance>= '2021-01-01')
        GROUP by place_id
        ORDER by number_of_visits DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 61+:

```
"SELECT DISTINCT place_id,COUNT(place_id) as number_of_visits
        from visit

        where (nfc_id in (select nfc_id

        from customers

        where(birthdate<='1960-01-01'))

        and

        date_of_entrance>= '2021-12-01')
        GROUP by place_id
        ORDER by number_of_visits DESC"
```

Για τις συχνότερα χρησιμοποιούμενες υπηρεσίες:

Για το έτος 2021 και ηλικιακή ομάδα 20-40:

```
"SELECT DISTINCT service_id, count(service_id) as most_used_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='2001-01-01' and birthdate>='1981-01-01')

        and

        date_of_charge>= '2021-01-01'
        group by service_id
         ORDER BY most_used_services DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 20-40:

```
"SELECT DISTINCT service_id, count(service_id) as most_used_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers
```

```
        where birthdate<='2001-01-01' and birthdate>='1981-01-01')

        and

        date_of_charge>= '2021-12-01'
        group by service_id
         ORDER BY most_used_services DESC"
```

Για το έτος 2021 και ηλικιακή ομάδα 41-60:

```
"SELECT DISTINCT service_id, count(service_id) as most_used_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='1980-01-01' and birthdate>='1961-01-01')

        and

        date_of_charge>= '2021-01-01'
        group by service_id
         ORDER BY most_used_services DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 41-60:

```
"SELECT DISTINCT service_id, count(service_id) as most_used_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='1980-01-01' and birthdate>='1961-01-01')

        and

        date_of_charge>= '2021-12-01'
        group by service_id
         ORDER BY most_used_services DESC"
```

Για το έτος 2021 και ηλικιακή ομάδα 61+:

```
"SELECT DISTINCT service_id, count(service_id) as most_used_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='1960-01-01' )

        and

        date_of_charge>= '2021-01-01'
        group by service_id
         ORDER BY most_used_services DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 61+:

```
"SELECT DISTINCT service_id, count(service_id) as most_used_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='1960-01-01' )

        and

        date_of_charge>= '2021-12-01'
        group by service_id
         ORDER BY most_used_services DESC"
```

Για τις υπηρεσίες που χρησιμοποιούνται από τους περισσότερους πελάτες:

Για το έτος 2021 και ηλικιακή ομάδα 20-40:

```
"SELECT service_id , count(nfc_id) as most_popular_services

        from enjoy_services
```

```
        where nfc_id in (select nfc_id
        from customers
        where birthdate<='2001-01-01' and birthdate>='1981-01-01')
        and
        date_of_charge>= '2021-01-01'
        group by service_id
        order by most_popular_services DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 20-40:

```
"SELECT service_id , count(nfc_id) as most_popular_services
        from enjoy_services
        where nfc_id in (select nfc_id
        from customers
        where birthdate<='2001-01-01' and birthdate>='1981-01-01')
        and
        date_of_charge>= '2021-12-01'
        group by service_id
        order by most_popular_services DESC"
```

Για το έτος 2021 και ηλικιακή ομάδα 41-60:

```
"SELECT service_id , count(nfc_id) as most_popular_services
        from enjoy_services
        where nfc_id in (select nfc_id
        from customers
```

```
        where birthdate<='1980-01-01' and birthdate>='1961-01-01')

        and

        date_of_charge>= '2021-01-01'

        group by service_id

        order by most_popular_services DESC"
```

Για το μήνα 12-2021 και ηλικιακή ομάδα 41-60:

```
"SELECT service_id , count(nfc_id) as most_popular_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='1980-01-01' and birthdate>='1961-01-01')

        and

        date_of_charge>= '2021-12-01'

        group by service_id

        order by most_popular_services DESC"
```

Για το έτος 2021 και ηλικιακή ομάδα 61+:

```
"SELECT service_id , count(nfc_id) as most_popular_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='1960-01-01')

        and

        date_of_charge>= '2021-01-01'
```

```
        group by service_id

        order by most_popular_services DESC"
```

Για τον μήνα 12-2021 και ηλικιακή ομάδα 61+:

```
"SELECT service_id , count(nfc_id) as most_popular_services

        from enjoy_services

        where nfc_id in (select nfc_id

        from customers

        where birthdate<='1960-01-01')

        and

        date_of_charge>= '2021-12-01'

        group by service_id

        order by most_popular_services DESC"
```

3. Configuration Files

Τα αρχεία, που θα χρειαστούμε, θα είναι τα παρακάτω, τα οποία πρέπει να βρίσκονται στον ίδιο φάκελο.

- *app.js*
- *styles.css*
- *covid.jpg.png*
- *Hotel_Covid.html*

- *Hotel_Covid.php*
- *More_info.html*
- *More_info.php*
- *Sales.html*
- *Sales.php*
- *Services.html*
- *Services.php*
- *Statistics.html*
- *Statistics.php*
- *Insert_data1.py*
- *Insert_data2.py*

4. Παρουσίαση

Ερώτημα	Αρχή	Τέλος
(a)	00:00	01:43
(b)	01:43	02:35
(c)	02:35	07:24
(d)	07:24	08:47
(e)	08:47	10:50
(f).7	10:50	12:26
(f).9	12:26	13:28
(f).10	13:28	13:54
(f).11	13:54	15:58
(g)	15:58	18:06