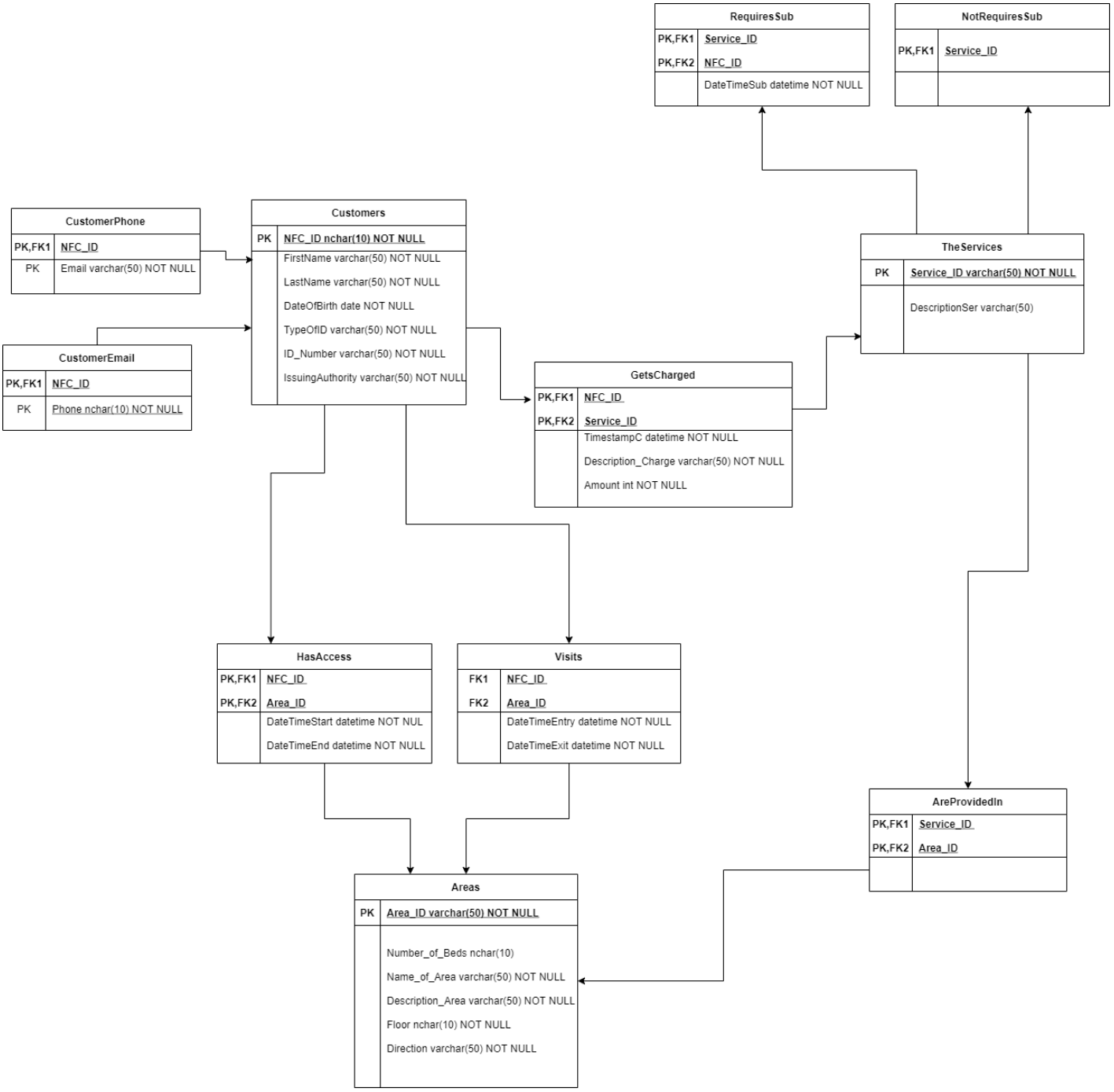
ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ
στο Μάθημα Βάσεις Δεδομένων

Ονοματεπώνυμα Μελών Ομάδας :

Ξυλιά Παναγιώτα Μικαέλα (ΑΜ : el18859)
Σερλής Εμμανουήλ-Αναστάσιος (ΑΜ : 03118125)
Χανή Βασιλική (ΑΜ : el18213)

ΠΕΡΙΕΧΟΜΕΝΑ

Ερώτημα 1 : Το σχεσιακό διάγραμμα της ΒΔ



Ερώτημα 1a :

Οι περιορισμοί που έχουμε ορίσει αφορούν τα χαρακτηριστικά τα οποία ορίσαμε ως NOT_NULL έτσι ώστε κάποια οντότητα ενός πίνακα (πχ. Customers) να μην μπορεί να αρχικοποιηθεί χωρίς αυτά (πχ. First Name).

Ερώτημα 1b :

Με τη δημιουργία των πινάκων και τον ορισμό των Primary Keys αυτομάτως η SQL δημιουργεί ευρετήρια τα οποία αφορούν αυτά τα κλειδιά. Ωστόσο για τη διευκόλυνση χρήσης της βάσης μας επιλέξαμε να δημιουργήσουμε τα παρακάτω τρία ευρετήρια.

```
CREATE INDEX Name_Of_Service ON TheServices(DescriptionSer);
CREATE INDEX Visits_ID ON Visits(Area_ID);
CREATE INDEX Provides_ID ON AreProvidedIn(Area_ID);
```

Τα οποία αποδείχθηκαν χρήσιμα στην βελτιστοποίηση της βάσης μας καθώς τα παραπάνω στοιχεία χρησιμοποιούνται σε διάφορα queries όπως το παρακάτω :

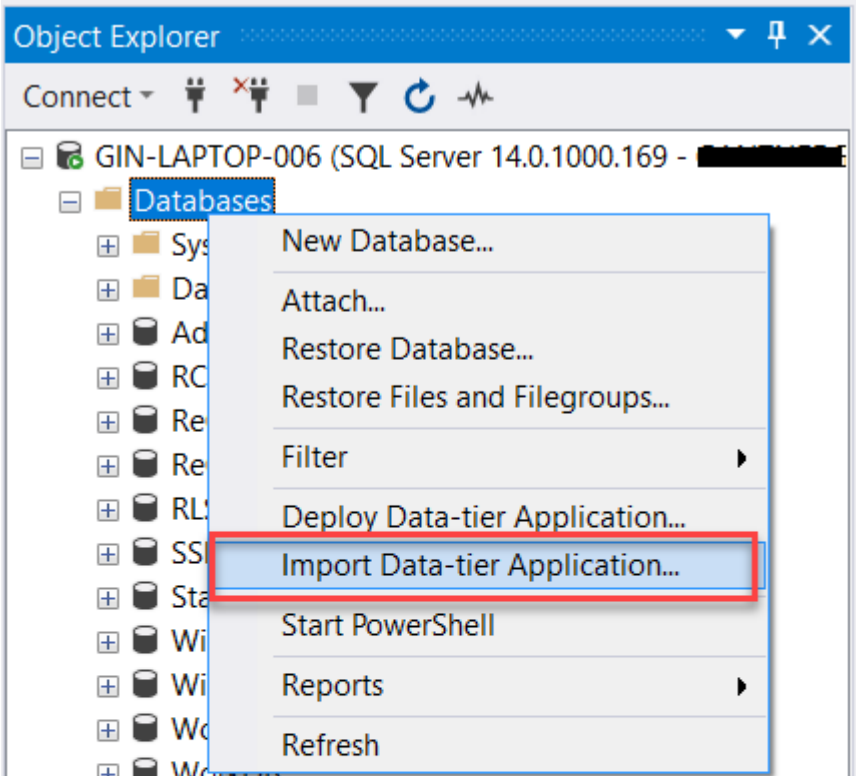
```
SELECT a.Service_ID, a.DescriptionSer, c.DateTimeEntry, c.DateTimeExit,
d.Description_Of_Area
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d
WHERE a.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND CAST(c.DateTimeEntry AS DATE) = '2010-06-08'
ORDER BY a.DescriptionSer;
```

Ερώτημα 1c:

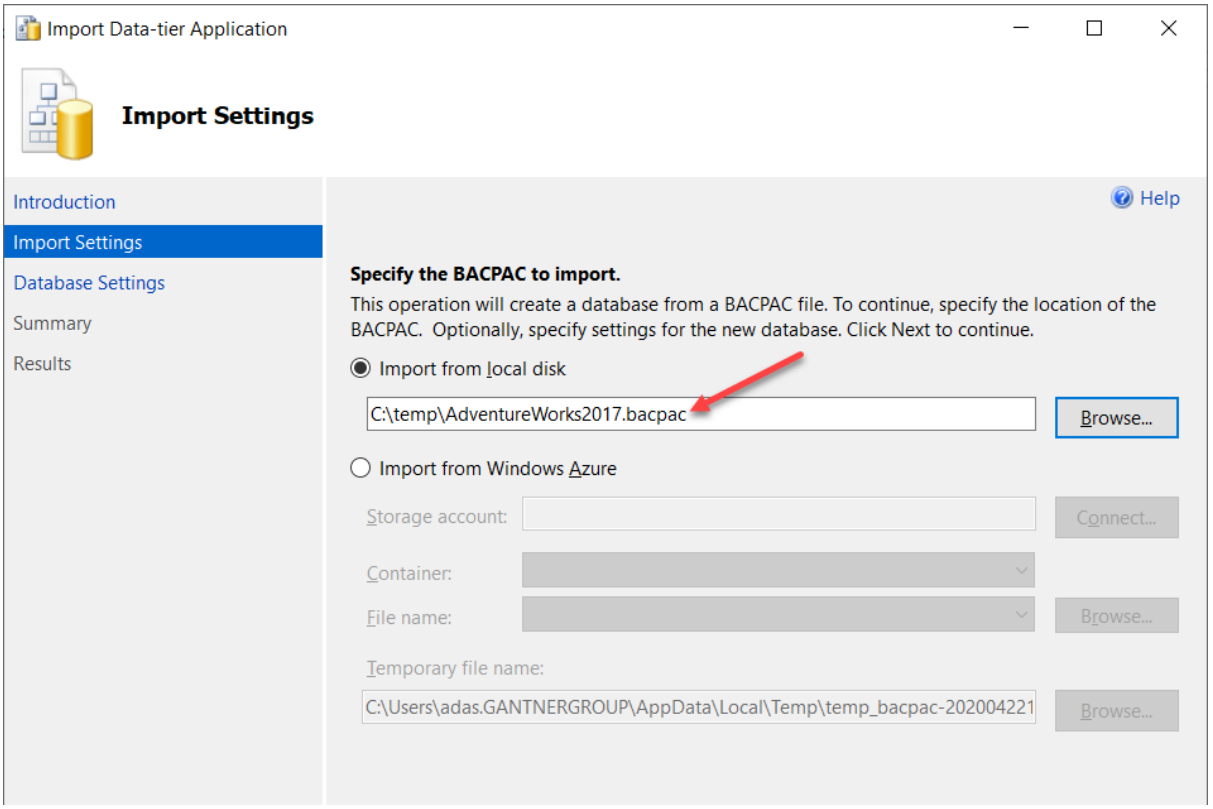
Για τη δημιουργία της βάσης μας χρησιμοποιήθηκε η εφαρμογή Microsoft Server Management Studio στην οποία συντάχθηκε σε SQL γλώσσα ο κώδικας που δημιουργεί τη βάση και τους πίνακες, τα ευρετήρια και τις όψεις της καθώς και τα διάφορα Queries που χρησιμοποιήσαμε. Η κατάλληλη διεπαφή χρήστη (User Interface) συντάχθηκε σε Python με χρήση της βιβλιοθήκης tkinter, τόσο για την σύνδεση βάσης και python όσο και για την δημιουργία του User Interface καθεαυτού.

Ερώτημα 1d:

Αρχικά, ο χρήστης καλείται να κάνει Import την βάση μας στον SQL Server, πατώντας δεξί κλικ στο Folder “Databases” του Object Explorer(στα αριστερά του SQL Server) και επιλέγει το “Import Data-Tier Application”.

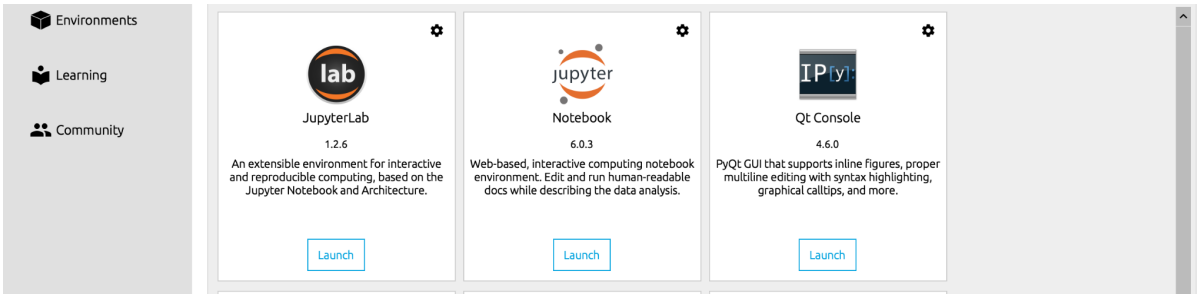


Στην συνέχεια, ακολουθεί τα βήματα του wizard και εισάγει το BACPAC αρχείο Hotel_Official.bacpac που περιλαμβάνεται στο zip παραδοτέο αρχείο.



Αφού ολοκληρωθεί η εν λόγω διαδικασία, θα έχει εγκατασταθεί η βάση δεδομένων μας στον SQL Server του εκάστοτε χρήστη και θα περιλαμβάνει τόσο τους εκάστοτε περιορισμούς (Primary και Foreign Keys, περιορισμούς ακεραιότητας, datatypes) όσο και τα tables συμπληρωμένα με τα απαιτούμενα δεδομένα

Μετ'έπετα, ο χρήστης πρέπει να εγκαταστήσει-αν δεν το έχει ήδη-την επιστημονική πλατφόρμα Anaconda από το δοθέν link: https://www.anaconda.com/products/individual και αφού την ανοίξει να επιλέξει το Launch Button στο Jupyter Notebook, ώστε να τρέξει η Python 3 στον browser της επιλογής του.



Τέλος, ανοίγει-με το Jupyter Notebook-το αρχείο Hotel_Official_UI.ipynb που βρίσκεται στον φάλεκο Project.zip και αλλάζει το όνομα του SQL Server στην γραμμή με το σχόλιο:

```
conn = pyodbc.connect(
    'Driver={SQL Server};'
    'Server=LAPTOP-P23KDOA5\SQLEXPRESS;' #allazei analoga me to pc
    'Database=Hotel_Official;'
    'Trusted_Connection=yes'

)
```

και πατάει το Run Button στο 1ο κελί. Έτσι, θα γίνει pop-up ένα παράθυρο στο Task Bar, το οποίο αν επιλεγεί με αριστερό κλικ θα ανοίξει το UI.

Σημείωση:Το όνομα του server μπορεί να βρεθεί στην πρώτη γραμμή του Object Explorer του SQL Server(βλέπε 1η εικόνα), ενώ το όνομα της βάσης παραμένει Hotel_Official (δεν χρειάζεται να αλλάξει)

Ερώτημα 2 : Ο Κώδικας SQL

Μέρος Α' : Δημιουργία της Βάσης

```
CREATE DATABASE Hotel_Official;
GO

CREATE TABLE Customers (
    NFC_ID nchar(5) NOT NULL PRIMARY KEY, /*AUTO_INCREMENT ?? */
    FirstName varchar(50) NOT NULL,
    Surname varchar(50) NOT NULL,
    DateOfBirth date NOT NULL,
    TypeOfID varchar(50) NOT NULL,
      ID_Number varchar(50) NOT NULL,
      IssuingAuthority varchar(50) NOT NULL
);

CREATE TABLE CustomerPhone (
      NFC_ID nchar(5) NOT NULL,
      phone_number nchar(10) NOT NULL,
      CONSTRAINT PKCustomerPhone PRIMARY KEY(NFC_ID, phone_number),
      CONSTRAINT FKCustomerPhone FOREIGN KEY (NFC_ID) REFERENCES
Customers (NFC_ID)
      ON DELETE CASCADE
      ON UPDATE CASCADE
      );

CREATE TABLE CustomerEmail (
      NFC_ID nchar(5) NOT NULL,
```

```
	email varchar(50) NOT NULL,
	CONSTRAINT PKCustomerEmail PRIMARY KEY(NFC_ID, email),
	CONSTRAINT FKCustomerEmail FOREIGN KEY (NFC_ID) REFERENCES
Customers (NFC_ID)
	ON DELETE CASCADE
	ON UPDATE CASCADE
	);

CREATE TABLE TheServices (
	Service_ID varchar(50) NOT NULL PRIMARY KEY,
	DescriptionSer varchar(100) NOT NULL
	);

CREATE TABLE AREA(
	Area_ID varchar(50) NOT NULL PRIMARY KEY,
	Number_Of_Beds nchar(10),
	Name_Of_Area varchar(50) NOT NULL,
	Description_Of_Area varchar(50) NOT NULL,
	Floor nchar(10) NOT NULL,
	Direction varchar(50) NOT NULL,

	);

CREATE TABLE HasAccess (
	NFC_ID nchar(5) NOT NULL,
	Area_ID varchar(50) NOT NULL,
	DateTimeStart datetime NOT NULL,
	DateTimeEnd datetime NOT NULL,
	CONSTRAINT PKHasAccess PRIMARY KEY(NFC_ID, Area_ID),
	CONSTRAINT FK1HasAccess FOREIGN KEY (NFC_ID) REFERENCES Customers
(NFC_ID)
	ON DELETE CASCADE
	ON UPDATE CASCADE,
	CONSTRAINT FK2HasAccess FOREIGN KEY (Area_ID) REFERENCES AREA
(Area_ID)
	ON DELETE CASCADE
	ON UPDATE CASCADE
 );

 CREATE TABLE Visits (
	 NFC_ID nchar(5) NOT NULL,
	 Area_ID varchar(50) NOT NULL,
	 DateTimeEntry datetime NOT NULL,
	 DateTimeExit datetime NOT NULL,
	/* CONSTRAINT PKVisits PRIMARY KEY(NFC_ID, Area_ID),*/
	 CONSTRAINT FK1Visits FOREIGN KEY (NFC_ID) REFERENCES Customers
```

```
(NFC_ID)
       ON DELETE CASCADE
       ON UPDATE CASCADE,
       CONSTRAINT FK2Visits FOREIGN KEY (Area_ID) REFERENCES AREA
(Area_ID)
       ON DELETE CASCADE
       ON UPDATE CASCADE
 );

 CREATE TABLE GetsCharged (
       NFC_ID nchar(5) NOT NULL,
       Service_ID varchar(50) NOT NULL,
       TimestampC datetime NOT NULL,
       Descpription_Charge varchar(50) NOT NULL,
       Amount INT NOT NULL,
       CONSTRAINT PKGetsCharged PRIMARY KEY(NFC_ID, Service_ID),
       CONSTRAINT FK1GetsCharged FOREIGN KEY (NFC_ID) REFERENCES
Customers (NFC_ID)
       ON DELETE CASCADE
       ON UPDATE CASCADE,
       CONSTRAINT FK2GetsCharged FOREIGN KEY (Service_ID) REFERENCES
TheServices (Service_ID)
       ON DELETE CASCADE
       ON UPDATE CASCADE
 );

CREATE TABLE AreProvidedIn (
      Service_ID varchar(50) NOT NULL,
      Area_ID varchar(50) NOT NULL,
      CONSTRAINT PKAreProvidedIn PRIMARY KEY(Service_ID, Area_ID),
      CONSTRAINT FK1AreProvidedIn FOREIGN KEY (Service_ID) REFERENCES
TheServices (Service_ID)
      ON DELETE CASCADE
      ON UPDATE CASCADE,
      CONSTRAINT FK2AreProvidedIn FOREIGN KEY (Area_ID) REFERENCES AREA
(Area_ID)
      ON DELETE CASCADE
      ON UPDATE CASCADE
);

CREATE TABLE RequiresSub (
       NFC_ID nchar(5) NOT NULL,
       Service_ID varchar(50) NOT NULL,
       DateTimeSub datetime NOT NULL,
       CONSTRAINT PKRequiresSub PRIMARY KEY(NFC_ID, Service_ID),
       CONSTRAINT FK1RequiresSub FOREIGN KEY (NFC_ID) REFERENCES
```

```
Customers (NFC_ID)
       ON DELETE CASCADE
       ON UPDATE CASCADE,
       CONSTRAINT FK2RequiresSub FOREIGN KEY (Service_ID) REFERENCES
TheServices (Service_ID)
       ON DELETE CASCADE
       ON UPDATE CASCADE
 );

CREATE TABLE NotRequiresSub (

       Service_ID varchar(50) NOT NULL,
       CONSTRAINT PKNotRequiresSub PRIMARY KEY(Service_ID),
       CONSTRAINT FKNotRequiresSub FOREIGN KEY (Service_ID) REFERENCES
TheServices (Service_ID)
       ON DELETE CASCADE
       ON UPDATE CASCADE
);


CREATE INDEX Name_Of_Service ON TheServices(DescriptionSer);
CREATE INDEX Visits_ID ON Visits(Area_ID);
CREATE INDEX Provides_ID ON AreProvidedIn(Area_ID);
```

Μέρος Β΄ : Queries

Τα παρακάτω Queries επιστρέφουν πίνακες που μέσω της διεπαφής επιτρέπουν στον χρήστη να απαντήσει σε χρήσιμα ερωτήματα σχετικά με τη βάση μας.

Query για το ερώτημα 7 :

```
SELECT h.Service_ID, p.DescriptionSer, g.Amount
FROM TheServices AS p, RequiresSub AS h, GetsCharged AS g
WHERE p.Service_ID = h.Service_ID
AND h.Service_ID = g.Service_ID
ORDER BY p.DescriptionSer;

SELECT h.Service_ID, p.DescriptionSer
FROM TheServices AS p, NotRequiresSub AS h
WHERE p.Service_ID = h.Service_ID
ORDER BY p.DescriptionSer;

DROP TABLE array_7a
DROP TABLE array_7b
DROP TABLE array_7c
```

```
DROP TABLE array_7d
DROP TABLE array_7e
DROP TABLE array_7f
DROP TABLE array_7g

/*Visits by Name of Service*/

SELECT a.Service_ID as Service_ID, a.DescriptionSer as DescriptionSer,
c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as DateTimeExit,
d.Floor as Floor, d.Direction as Direction into array_7a
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d
WHERE a.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND a.DescriptionSer =  'Restaurant' --input01
ORDER BY a.DescriptionSer;

/*Visits by Date*/

SELECT a.Service_ID as Service_ID, a.DescriptionSer as DescriptionSer,
c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as DateTimeExit,
d.Floor as Floor, d.Direction as Direction into array_7b
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d
WHERE a.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND CAST(c.DateTimeEntry AS DATE) =  '2021-06-10'--input02
ORDER BY a.DescriptionSer;

/*Visits by Cost*/

SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as
Amount into array_7c
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND h.NFC_ID = c.NFC_ID
AND h.Service_ID = a.Service_ID
AND h.Amount <= 35 --input03
ORDER BY a.DescriptionSer;
```

```
--visits by date and cost

SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as
Amount into array_7d
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND h.NFC_ID = c.NFC_ID
AND h.Service_ID = a.Service_ID
AND h.Amount <= 35 --input03
AND CAST(c.DateTimeEntry AS DATE) =  '2021-06-10'--input02
ORDER BY a.DescriptionSer;

--visits by Cost and Service_Name

SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as
Amount into array_7e
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND h.NFC_ID = c.NFC_ID
AND h.Service_ID = a.Service_ID
AND h.Amount <= 35 --input03
AND a.DescriptionSer = 'Restaurant'--input02
ORDER BY a.DescriptionSer;

--visits by date and service_name

SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as
Amount into array_7f
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
```

```
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND h.NFC_ID = c.NFC_ID
AND c.Area_ID = d.Area_ID
AND h.Service_ID = a.Service_ID
AND a.DescriptionSer = 'Bar'
AND CAST(c.DateTimeEntry AS DATE) =  '2021-05-09'--input02
ORDER BY a.DescriptionSer;

--visits by all criteria

SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as
Amount into array_7g
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND h.NFC_ID = c.NFC_ID
AND h.Service_ID = a.Service_ID
AND a.DescriptionSer = 'Bar'
AND CAST(c.DateTimeEntry AS DATE) =  '2021-05-09'--input02
AND h.Amount <=20
ORDER BY a.DescriptionSer;

SELECT *
FROM array_7g

/*All Visits*/

SELECT a.Service_ID, a.DescriptionSer, c.DateTimeEntry,
c.DateTimeExit,d.Floor, d.Direction
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d
WHERE a.Service_ID = b.Service_ID
AND b.Area_ID  = c.Area_ID
AND c.Area_ID = d.Area_ID
ORDER BY a.DescriptionSer;
```

Query για το ερώτημα 9 :

```
DROP TABLE array_9
SELECT CU.Firstname as A, CU.Surname as B, V.Area_ID as C,
```

```
V.DateTimeEntry as D, V.DateTimeExit as E into array_9
FROM AREA as AR, Visits as V, Customers as CU
WHERE (CU.NFC_ID = V.NFC_ID) and (AR.Area_ID=V.Area_ID) and (CU.NFC_ID =
000015) --Insert NFC_ID of covid case

SELECT *
FROM array_9
```

Query για το ερώτημα 10 :

```
DROP TABLE array_mid
SELECT DATEADD(hour,1,array_9.E) as DateExitplus1 into array_mid
FROM array_9

SELECT DISTINCT CU.Firstname, CU.Surname
FROM Customers as CU, Visits as V, array_9, array_mid
WHERE (CU.NFC_ID=V.NFC_ID) and ( V.Area_ID = array_9.C) and ( (
(V.DateTimeExit >= array_9.D) and (V.DateTimeExit <=
array_mid.DateExitplus1) ) or

( (V.DateTimeEntry >= array_9.D) and (V.DateTimeEntry <=
array_mid.DateExitplus1) ) or ( (V.DateTimeExit >=
array_mid.DateExitplus1) and (V.DateTimeEntry <= array_9.D) ) )
```

Query για το ερώτημα 11 :

```
DROP TABLE #tempCustomers;
CREATE TABLE #tempCustomers (
    NFC_ID int NOT NULL PRIMARY KEY,
    FirstName varchar(50) NOT NULL,
    Surname varchar(50) NOT NULL,
    Age varchar(50),
);

INSERT INTO #tempCustomers
 SELECT NFC_ID, FirstName, Surname,
           CASE
                 WHEN Age BETWEEN 20 AND 40 THEN '20-40'
                 WHEN Age BETWEEN 40 AND 60 THEN '40-60'
                 WHEN Age > 60 THEN '60+'
           END AS 'Ages'

 FROM (SELECT NFC_ID, FirstName, Surname, Floor(DateDiff(d, DateOfBirth,
GetDate()) / 365.25) AS Age
        FROM dbo.customers )
         AS Ages
```

```
SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 20-40"
FROM TheServices AS a, (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-01-01' AND '2021-12-31')
AS b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='20-40'
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 20-40" DESC;

SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 40-60"
FROM TheServices AS a, (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-01-01' AND '2021-12-31')
AS b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='40-60'
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 40-60"  DESC;

SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 60+"
FROM TheServices AS a, (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-01-01' AND '2021-12-31')
AS b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='60+'
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 60+"  DESC;

SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used for 20-40"
FROM TheServices AS a,
            (SELECT * FROM Visits
```

```
            WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS
b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age = '20-40'
GROUP BY DescriptionSer
ORDER BY "Most Used for 20-40" DESC;


SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used for 40-60"
FROM TheServices AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS
b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='40-60'
GROUP BY DescriptionSer
ORDER BY "Most Used for 40-60" DESC;


SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used for 60+"
FROM TheServices AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS
b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='60+'
GROUP BY DescriptionSer
ORDER BY "Most Used for 60+" DESC;

SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used for 20-40"
FROM TheServices AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS
b,
```

```
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age = '20-40'
GROUP BY DescriptionSer
ORDER BY "Most Used for 20-40" DESC;


SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used for 40-60"
FROM TheServices AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS
b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='40-60'
GROUP BY DescriptionSer
ORDER BY "Most Used for 40-60" DESC;


SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used for 60+"
FROM TheServices AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS
b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='60+'
GROUP BY DescriptionSer
ORDER BY "Most Used for 60+" DESC;

SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for
20-40"
FROM  AREA AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS
b,
            #tempCustomers AS d
```

```
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age='20-40'
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 20-40" DESC;

SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for
40-60"
FROM  AREA AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS
b,
            #tempCustomers AS d
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age='40-60'
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 40-60" DESC;;

SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for 60+"
FROM  AREA AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS
b,
            #tempCustomers AS d
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age='60+'
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 60+" DESC;;

SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for
20-40"
FROM  AREA AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS
b,
            #tempCustomers AS d
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age='20-40'
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 20-40" DESC;

SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for
40-60"
```

```
FROM  AREA AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS
b,
            #tempCustomers AS d
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age='40-60'
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 40-60" DESC;;

SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for 60+"
FROM  AREA AS a,
            (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS
b,
            #tempCustomers AS d
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age='60+'
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 60+" DESC;;

SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 20-40"
FROM TheServices AS a, (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-01-01' AND '2021-12-31')
AS b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='20-40'
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 20-40" DESC;

SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 40-60"
FROM TheServices AS a, (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-01-01' AND '2021-12-31')
AS b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
```

```
AND b.NFC_ID = d.NFC_ID
AND d.Age='40-60'
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 40-60"  DESC;

SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 60+"
FROM TheServices AS a, (SELECT * FROM Visits
            WHERE DateTimeEntry BETWEEN '2021-01-01' AND '2021-12-31')
AS b,
            AreProvidedIn AS c,
            #tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age='60+'
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 60+"  DESC;
```

Μέρος Γ' : Όψεις

```
CREATE VIEW Services_Charged
AS
SELECT b.Service_ID, b.DescriptionSer, c.Amount, c.TimestampC
FROM TheServices b, GetsCharged c
WHERE b.Service_ID = c.Service_ID;

CREATE VIEW Customer_Info
AS
SELECT DISTINCT b.NFC_ID, b.FirstName, b.Surname, b.DateOfBirth,
b.ID_Number, b.TypeOfID, b.IssuingAuthority, c.email, d.phone_number
FROM Customers b, CustomerEmail c, CustomerPhone d
WHERE b.NFC_ID = c.NFC_ID
AND c.NFC_ID = d.NFC_ID;
```

Ερώτημα 3 :

Τα αρχεία τεχνοδιαμόρφωσης (configuration) που χρειάζονται για να εγκατασταθεί από την αρχή την εφαρμογή μας έχουν ως εξής:

```
from tkinter import *
from PIL import ImageTk, Image
import pyodbc

#connect to SQL Server
```

```
conn = pyodbc.connect(
       'Driver={SQL Server};'
       'Server=LAPTOP-P23KDOA5\SQLEXPRESS;' #allazei analoga me to pc
       'Database=Hotel_Official;'
       'Trusted_Connection=yes'

)

c=conn.cursor()

root = Tk()
root.title("Welcome!")
root.geometry("1700x1700")

req_label_1=Label(root)
req_label_2=Label(root)
label71=Label(root)
label72=Label(root)
label73=Label(root)
label74=Label(root)
label75=Label(root)
label76=Label(root)
label77=Label(root)
nreq_label_1=Label(root)
nreq_label_2=Label(root)

def q7():

    global req_label_1
    global req_label_2
    global nreq_label_1
    global nreq_label_2
    global label71
    global label72
    global label73
    global label74
    global label75
    global label76
    global label77
    req_label_1.destroy()
    req_label_2.destroy()
    label72.destroy()
    label73.destroy()
    label74.destroy()
    label75.destroy()
    label76.destroy()
    label77.destroy()
    label71.destroy()
```

```
    nreq_label_1.destroy()
    nreq_label_2.destroy()

    c=conn.cursor()
    date=n7a.get()
    service=n7b.get()
    cost=n7c.get()

    if(cost=='' and service=='' and date==''): #general case
        req_string='''SELECT DISTINCT h.Service_ID, p.DescriptionSer
        FROM TheServices AS p, RequiresSub AS h
        WHERE p.Service_ID = h.Service_ID
        ORDER BY p.DescriptionSer;'''
        c.execute(req_string)
        req_rec=c.fetchall()
        print_req_2=str("Description of Sub Services")+"\n"
        print_req_1=str("Service ID") + "\n"
        for record in (req_rec):
             print_req_1+=str(record[0]) + "\n"
             print_req_2+=str(record[1]) + "\n"
        req_label_1=Label(root,text=print_req_1)
        req_label_1.grid(row=1,column=10)
        req_label_2=Label(root,text=print_req_2)
        req_label_2.grid(row=1,column=20)

        nreq_string='''SELECT DISTINCT h.Service_ID, p.DescriptionSer
        FROM TheServices AS p, NotRequiresSub AS h
        WHERE p.Service_ID = h.Service_ID
        ORDER BY p.DescriptionSer;'''
        c.execute(nreq_string)
        nreq_rec=c.fetchall()
        print_nreq_2=str("Description of Non_Sub Services")+"\n"
        print_nreq_1=str("Service ID") + "\n"
        for record in (nreq_rec):
             print_nreq_1+=str(record[0]) + "\n"
             print_nreq_2+=str(record[1]) + "\n"
        nreq_label_1=Label(root,text=print_nreq_1)
        nreq_label_1.grid(row=1,column=200)
        nreq_label_2=Label(root,text=print_nreq_2)
        nreq_label_2.grid(row=1,column=210)


    c3=conn.cursor()

    if(cost=='' and service=='' and date!=''): #case 1
        case7_rec = c3.execute('''
                    SELECT *
FROM (
```

```
SELECT a.Service_ID as Service_ID, a.DescriptionSer as DescriptionSer,
c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as DateTimeExit, d.Floor as
Floor, d.Direction as Direction
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d
WHERE a.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND CAST(c.DateTimeEntry AS DATE) = ?
) as array_7b
ORDER BY DescriptionSer;''', str(date) )
        case7_rec=c3.fetchall()
        print71=str("Service Id")+"\n"
        print72=str("Description of Service")+"\n"
        print73=str("Date of entry")+"\n"
        print74=str("Date of exit")+"\n"
        print75=str("Floor")+"\n"
        print76=str("Direction")+"\n"
        for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
            print76+=str(record[5])+"\n"
        label71=Label(root,text=print71)
        label71.grid(row=1,column=300)
        label72=Label(root,text=print72)
        label72.grid(row=1,column=310)
        label73=Label(root,text=print73)
        label73.grid(row=1,column=320)
        label74=Label(root,text=print74)
        label74.grid(row=1,column=330)
        label75=Label(root,text=print75)
        label75.grid(row=1,column=340)
        label76=Label(root,text=print76)
        label76.grid(row=1,column=350)

    if(cost=='' and date=='' and service!=''): #case2
        case7_rec = c3.execute('''
                    SELECT *
                    FROM (
                    SELECT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction
                    FROM TheServices AS a, AreProvidedIn AS b, Visits AS c,
AREA AS d
                    WHERE a.Service_ID = b.Service_ID
                    AND b.Area_ID = c.Area_ID
```

```
                    AND c.Area_ID = d.Area_ID
                    AND a.DescriptionSer =  ? ) as array_7a
                    ORDER BY DescriptionSer;''', str(service) )
        case7_rec=c3.fetchall()
        print71=str("Service Id")+"\n"
        print72=str("Description of Service")+"\n"
        print73=str("Date of entry")+"\n"
        print74=str("Date of exit")+"\n"
        print75=str("Floor")+"\n"
        print76=str("Direction")+"\n"
        for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
            print76+=str(record[5])+"\n"
        label71=Label(root,text=print71)
        label71.grid(row=1,column=300)
        label72=Label(root,text=print72)
        label72.grid(row=1,column=310)
        label73=Label(root,text=print73)
        label73.grid(row=1,column=320)
        label74=Label(root,text=print74)
        label74.grid(row=1,column=330)
        label75=Label(root,text=print75)
        label75.grid(row=1,column=340)
        label76=Label(root,text=print76)
        label76.grid(row=1,column=350)

    if(service=='' and date=='' and cost!=''): #case3
        case7_rec = c3.execute('''
                    SELECT *
                    FROM (
                    SELECT DISTINCT a.Service_ID as Service_ID,
a.DescriptionSer as DescriptionSer, c.DateTimeEntry as DateTimeEntry,
c.DateTimeExit as DateTimeExit, d.Floor as Floor, d.Direction as Direction,
h.Amount as Amount
                    FROM TheServices AS a, AreProvidedIn AS b, Visits AS c,
AREA AS d, GetsCharged AS h, RequiresSub AS g
                    WHERE a.Service_ID = b.Service_ID
                    AND g.Service_ID = b.Service_ID
                    AND b.Area_ID = c.Area_ID
                    AND c.Area_ID = d.Area_ID
                    AND h.Service_ID = a.Service_ID
                    AND c.NFC_ID = h.NFC_ID
                    AND h.Amount <= ?) as array_7c
                    ORDER BY DescriptionSer;''' , int(cost) )
```

```
        case7_rec=c3.fetchall()
        print71=str("Service Id")+"\n"
        print72=str("Service Name")+"\n"
        print73=str("Date of entry")+"\n"
        print74=str("Date of exit")+"\n"
        print75=str("Floor")+"\n"
        print76=str("Direction")+"\n"
        print77=str("Amount")+"\n"
        for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
            print76+=str(record[5])+"\n"
            print77+=str(record[6])+"\n"
        print(print77)
        label71=Label(root,text=print71)
        label71.grid(row=1,column=3)
        label72=Label(root,text=print72)
        label72.grid(row=1,column=4)
        label73=Label(root,text=print73)
        label73.grid(row=1,column=5)
        label74=Label(root,text=print74)
        label74.grid(row=1,column=6)
        label75=Label(root,text=print75)
        label75.grid(row=1,column=7)
        label76=Label(root,text=print76)
        label76.grid(row=1,column=8)
        label77=Label(root,text=print77)
        label77.grid(row=1,column=10)

    if(cost=='' and service!='' and date!=''): #case4
        case7_rec = c3.execute('''
                    SELECT *
                    FROM (
                    SELECT DISTINCT a.Service_ID as Service_ID,
a.DescriptionSer as DescriptionSer, c.DateTimeEntry as DateTimeEntry,
c.DateTimeExit as DateTimeExit, d.Floor as Floor, d.Direction as Direction
                    FROM TheServices AS a, AreProvidedIn AS b, Visits AS c,
AREA AS d
                    WHERE a.Service_ID = b.Service_ID
                    AND b.Area_ID = c.Area_ID
                    AND c.Area_ID = d.Area_ID
                    AND a.DescriptionSer = ?
                    AND CAST(c.DateTimeEntry AS DATE) =  ?) as array_7f
                    ORDER BY DescriptionSer;''' , (str(service),str(date) )
)
```

```
        case7_rec=c3.fetchall()
        print71=str("Service Id")+"\n"
        print72=str("Service Name")+"\n"
        print73=str("Date of entry")+"\n"
        print74=str("Date of exit")+"\n"
        print75=str("Floor")+"\n"
        print76=str("Direction")+"\n"
        for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
            print76+=str(record[5])+"\n"
        label71=Label(root,text=print71)
        label71.grid(row=1,column=2,rowspan=10)
        label72=Label(root,text=print72)
        label72.grid(row=1,column=3,rowspan=10)
        label73=Label(root,text=print73)
        label73.grid(row=1,column=4,rowspan=10)
        label74=Label(root,text=print74)
        label74.grid(row=1,column=5,rowspan=10)
        label75=Label(root,text=print75)
        label75.grid(row=1,column=6,rowspan=10)
        label76=Label(root,text=print76)
        label76.grid(row=1,column=7,rowspan=10)

    if(service=='' and cost!='' and date!=''): #case5
        case7_rec = c3.execute('''
        SELECT *
        FROM(
SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as Amount
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.NFC_ID = h.NFC_ID
AND c.Area_ID = d.Area_ID
AND h.Service_ID = a.Service_ID
AND h.Amount <= ? --input03
AND CAST(c.DateTimeEntry AS DATE) =  ?) as array_7d
ORDER BY DescriptionSer''' , (str(cost),str(date) ) )
        case7_rec=c3.fetchall()
        print71=str("Service Id")+"\n"
        print72=str("Service Name")+"\n"
```

```
        print73=str("Date of entry")+"\n"
        print74=str("Date of exit")+"\n"
        print75=str("Floor")+"\n"
        print76=str("Direction")+"\n"
        print77=str("Amount")+"\n"
        for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
            print76+=str(record[5])+"\n"
            print77+=str(record[6])+"\n"
        label71=Label(root,text=print71)
        label71.grid(row=1,column=2,rowspan=10)
        label72=Label(root,text=print72)
        label72.grid(row=1,column=3,rowspan=10)
        label73=Label(root,text=print73)
        label73.grid(row=1,column=4,rowspan=10)
        label74=Label(root,text=print74)
        label74.grid(row=1,column=5,rowspan=10)
        label75=Label(root,text=print75)
        label75.grid(row=1,column=6,rowspan=10)
        label76=Label(root,text=print76)
        label76.grid(row=1,column=7,rowspan=10)
        label77=Label(root,text=print77)
        label77.grid(row=1,column=8,rowspan=10)

    if(date=='' and cost!='' and service!=''): #case6
        case7_rec = c3.execute('''
        SELECT *
        FROM(SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as Amount
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND h.Service_ID = a.Service_ID
AND c.NFC_ID = h.NFC_ID
AND h.Amount <= ?
AND a.DescriptionSer = ?) as array_7e
ORDER BY DescriptionSer''' , (str(cost),str(service) ) )
        case7_rec=c3.fetchall()
        print71=str("Service Id")+"\n"
        print72=str("Service Name")+"\n"
```

```
        print73=str("Date of entry")+"\n"
        print74=str("Date of exit")+"\n"
        print75=str("Floor")+"\n"
        print76=str("Direction")+"\n"
        print77=str("Amount") +"\n"
        for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
            print76+=str(record[5])+"\n"
            print77+=str(record[6])+"\n"
        label71=Label(root,text=print71)
        label71.grid(row=1,column=2,rowspan=10)
        label72=Label(root,text=print72)
        label72.grid(row=1,column=3,rowspan=10)
        label73=Label(root,text=print73)
        label73.grid(row=1,column=4,rowspan=10)
        label74=Label(root,text=print74)
        label74.grid(row=1,column=5,rowspan=10)
        label75=Label(root,text=print75)
        label75.grid(row=1,column=6,rowspan=10)
        label76=Label(root,text=print76)
        label76.grid(row=1,column=7,rowspan=10)
        label77=Label(root,text=print77)
        label77.grid(row=1,column=8,rowspan=10)


    if(date!='' and cost!='' and service!=''): #case7
        case7_rec = c3.execute('''
        SELECT *
        FROM(
SELECT DISTINCT a.Service_ID as Service_ID, a.DescriptionSer as
DescriptionSer, c.DateTimeEntry as DateTimeEntry, c.DateTimeExit as
DateTimeExit, d.Floor as Floor, d.Direction as Direction, h.Amount as Amount
FROM TheServices AS a, AreProvidedIn AS b, Visits AS c, AREA AS d,
GetsCharged AS h, RequiresSub AS g
WHERE a.Service_ID = b.Service_ID
AND g.Service_ID = b.Service_ID
AND b.Area_ID = c.Area_ID
AND c.Area_ID = d.Area_ID
AND c.NFC_ID = h.NFC_ID
AND h.Service_ID = a.Service_ID
AND a.DescriptionSer = ?
AND CAST(c.DateTimeEntry AS DATE) =  ?
AND h.Amount <= ?) as array_7g
ORDER BY DescriptionSer''' , (str(service),str(date),int(cost) ) )
```

```
        case7_rec=c3.fetchall()
        print71=str("Service Id")+"\n"
        print72=str("Service Name")+"\n"
        print73=str("Date of entry")+"\n"
        print74=str("Date of exit")+"\n"
        print75=str("Floor")+"\n"
        print76=str("Direction")+"\n"
        print77=str("Amount") +"\n"
        for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
            print76+=str(record[5])+"\n"
            print77+=str(record[6])+"\n"
        label71=Label(root,text=print71)
        label71.grid(row=1,column=2,rowspan=10)
        label72=Label(root,text=print72)
        label72.grid(row=1,column=3,rowspan=10)
        label73=Label(root,text=print73)
        label73.grid(row=1,column=4,rowspan=10)
        label74=Label(root,text=print74)
        label74.grid(row=1,column=5,rowspan=10)
        label75=Label(root,text=print75)
        label75.grid(row=1,column=6,rowspan=10)
        label76=Label(root,text=print76)
        label76.grid(row=1,column=7,rowspan=10)
        label77=Label(root,text=print77)
        label77.grid(row=1,column=8,rowspan=10)



    conn.commit()
    n9.delete(0,END)
    n11a.delete(0,END)
    n11b.delete(0,END)


def q9():
    global req_label_1
    global req_label_2
    global nreq_label_1
    global nreq_label_2
    global label71
    global label72
    global label73
    global label74
```

```
    global label75
    global label76
    global label77
    req_label_1.destroy()
    req_label_2.destroy()
    label72.destroy()
    label73.destroy()
    label74.destroy()
    label75.destroy()
    label76.destroy()
    label77.destroy()
    label71.destroy()
    nreq_label_1.destroy()
    nreq_label_2.destroy()

    c4=conn.cursor()
    covid_id=n9.get()
    print(covid_id)

    case7_rec=c4.execute('''
SELECT *
FROM  (
SELECT CU.Firstname as A, CU.Surname as B, V.Area_ID as C, V.DateTimeEntry
as D, V.DateTimeExit as E
FROM AREA as AR, Visits as V, Customers as CU
WHERE CU.NFC_ID = V.NFC_ID
AND AR.Area_ID = V.Area_ID
AND CU.NFC_ID = ?
) as array_9
ORDER BY C;''',str(covid_id)  )

    case7_rec=c4.fetchall()
    print71=str("First Name")+"\n"
    print72=str("Surname")+"\n"
    print73=str("Area_ID")+"\n"
    print74=str("Date of entry")+"\n"
    print75=str("Date of Exit")+"\n"
    for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
            print73+=str(record[2])+"\n"
            print74+=str(record[3])+"\n"
            print75+=str(record[4])+"\n"
    label71=Label(root,text=print71)
    label71.grid(row=1,column=2,rowspan=10)
    label72=Label(root,text=print72)
    label72.grid(row=1,column=3,rowspan=10)
    label73=Label(root,text=print73)
```

```
    label73.grid(row=1,column=4,rowspan=10)
    label74=Label(root,text=print74)
    label74.grid(row=1,column=5,rowspan=10)
    label75=Label(root,text=print75)
    label75.grid(row=1,column=6,rowspan=10)

    conn.commit()
    n7a.delete(0,END)
    n7b.delete(0,END)
    n7c.delete(0,END)
    n11a.delete(0,END)
    n11b.delete(0,END)




def q10():
    global req_label_1
    global req_label_2
    global nreq_label_1
    global nreq_label_2
    global label71
    global label72
    global label73
    global label74
    global label75
    global label76
    global label77
    req_label_1.destroy()
    req_label_2.destroy()
    label72.destroy()
    label73.destroy()
    label74.destroy()
    label75.destroy()
    label76.destroy()
    label77.destroy()
    label71.destroy()
    nreq_label_1.destroy()
    nreq_label_2.destroy()

    c4=conn.cursor()
    covid_id=n9.get()
    print(covid_id)

    case7_rec=c4.execute('''
SELECT DISTINCT CU.Firstname, CU.Surname
FROM Customers as CU, Visits as V,
```

```
(SELECT CU.Firstname as A, CU.Surname as B, V.Area_ID as C, V.DateTimeEntry
as D, V.DateTimeExit as E
FROM AREA as AR, Visits as V, Customers as CU
WHERE CU.NFC_ID = V.NFC_ID
AND AR.Area_ID = V.Area_ID
AND CU.NFC_ID = ?
) as array_9,
(SELECT DATEADD(hour,1,array_9.E) as DateExitplus1
FROM array_9) as array_mid
WHERE (CU.NFC_ID=V.NFC_ID) and ( V.Area_ID = array_9.C) and ( (
(V.DateTimeExit >= array_9.D) and (V.DateTimeExit <=
array_mid.DateExitplus1) ) or

( (V.DateTimeEntry >= array_9.D) and (V.DateTimeEntry <=
array_mid.DateExitplus1) ) or ( (V.DateTimeExit >= array_mid.DateExitplus1)
and (V.DateTimeEntry <= array_9.D) ) )''',str(covid_id)  )

    case7_rec=c4.fetchall()
    print71=str("First Name")+"\n"
    print72=str("Surname")+"\n"
    for record in (case7_rec):
            print71+=str(record[0])+"\n"
            print72+=str(record[1])+"\n"
    label71=Label(root,text=print71)
    label71.grid(row=1,column=2,rowspan=10)
    label72=Label(root,text=print72)
    label72.grid(row=1,column=3,rowspan=10)



    conn.commit()
    n7a.delete(0,END)
    n7b.delete(0,END)
    n7c.delete(0,END)
    n11a.delete(0,END)
    n11b.delete(0,END)



def q11():
    global req_label_1
    global req_label_2
    global nreq_label_1
    global nreq_label_2
    global label71
    global label72
    global label73
    global label74
```

```
    global label75
    global label76
    global label77
    req_label_1.destroy()
    req_label_2.destroy()
    label72.destroy()
    label73.destroy()
    label74.destroy()
    label75.destroy()
    label76.destroy()
    label77.destroy()
    label71.destroy()
    nreq_label_1.destroy()
    nreq_label_2.destroy()

    quest=n11a.get()
    span=n11b.get()
    ages=['20-40','40-60','60+']
    services2_ttl=['Most Used Services by Ages 20-40','Most Used Services by
Ages 40-60','Most Used Services by Ages 60+']
    areas_ttl=['Most Visited Areas by Ages 20-40','Most Visited Areas by
Ages 40-60','Most Visited Areas by Ages 60+']
    services3_ttl=['Services Provided To Most (Individual) Customers (Ages
20-40)','Services Provided To Most (Individual) Customers (Ages
40-60)','Services Provided To Most (Individual) Customers (Ages 60+)']
    if(quest=='11.2' and span=='Last Year'): #case 1
        for i in range(0,3):
            c.execute('''
SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used Services"
FROM TheServices AS a,
        (SELECT * FROM Visits
        WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS b,
        AreProvidedIn AS c, tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age = ?
GROUP BY DescriptionSer
ORDER BY "Most Used Services" DESC;''',(str(ages[i])) )
            case7_rec=c.fetchall()
            print71=str("Name of Service")+"\n"
            print72=str(services2_ttl[i])+"\n"
            for record in (case7_rec):
                print71+=str(record[0])+"\n"
                print72+=str(record[1])+"\n"
            if(i==0):
                    label71=Label(root,text=print71)
                    label71.grid(row=1*i*10,column=2,rowspan=10)
```

```
                label72=Label(root,text=print72)
                label72.grid(row=1*i*10,column=3,rowspan=10)
            if(i==1):
                label73=Label(root,text=print71)
                label73.grid(row=1*i*10,column=2,rowspan=10)
                label74=Label(root,text=print72)
                label74.grid(row=1*i*10,column=3,rowspan=10)
            if(i==2):
                label75=Label(root,text=print71)
                label75.grid(row=1*i*10,column=2,rowspan=10)
                label76=Label(root,text=print72)
                label76.grid(row=1*i*10,column=3,rowspan=10)

    if(quest=='11.1' and span=='Last Year'): #case2
        for i in range(0,3):
            c.execute('''
SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for 20-40"
FROM  AREA AS a,
        (SELECT * FROM Visits
        WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-6-30') AS b,
        tempCustomers AS d
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age=?
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 20-40" DESC;''',(str(ages[i])) )
            case7_rec=c.fetchall()
            print71=str("Name of Area")+"\n"
            print72=str(areas_ttl[i])+"\n"
            for record in (case7_rec):
                print71+=str(record[0])+"\n"
                print72+=str(record[1])+"\n"
            if(i==0):
                label71=Label(root,text=print71)
                label71.grid(row=1*i*10,column=2,rowspan=10)
                label72=Label(root,text=print72)
                label72.grid(row=1*i*10,column=3,rowspan=10)
            if(i==1):
                label73=Label(root,text=print71)
                label73.grid(row=1*i*10,column=2,rowspan=10)
                label74=Label(root,text=print72)
                label74.grid(row=1*i*10,column=3,rowspan=10)
            if(i==2):
                label75=Label(root,text=print71)
                label75.grid(row=1*i*10,column=2,rowspan=10)
                label76=Label(root,text=print72)
                label76.grid(row=1*i*10,column=3,rowspan=10)
```

```
    if(quest=='11.3' and span=='Last Year'): #case3
        for i in range(0,3):
            c.execute('''
SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 20-40"
FROM TheServices AS a, (SELECT * FROM Visits
        WHERE DateTimeEntry BETWEEN '2020-06-01' AND '2021-06-30') AS b,
        AreProvidedIn AS c,
        tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age=?
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 20-40" DESC;''',(str(ages[i])) )
            case7_rec=c.fetchall()
            print71=str("Name of Service")+"\n"
            print72=str(services3_ttl[i])+"\n"
            for record in (case7_rec):
                print71+=str(record[0])+"\n"
                print72+=str(record[1])+"\n"
            if(i==0):
                    label71=Label(root,text=print71)
                    label71.grid(row=1*i*10,column=2,rowspan=10)
                    label72=Label(root,text=print72)
                    label72.grid(row=1*i*10,column=3,rowspan=10)
            if(i==1):
                    label73=Label(root,text=print71)
                    label73.grid(row=1*i*10,column=2,rowspan=10)
                    label74=Label(root,text=print72)
                    label74.grid(row=1*i*10,column=3,rowspan=10)
            if(i==2):
                    label75=Label(root,text=print71)
                    label75.grid(row=1*i*10,column=2,rowspan=10)
                    label76=Label(root,text=print72)
                    label76.grid(row=1*i*10,column=3,rowspan=10)

    if(quest=='11.2' and span=='Last Month'): #case 4
            for i in range(0,3):
                c.execute('''
SELECT a.DescriptionSer, COUNT(b.Area_ID) AS "Most Used Services"
FROM TheServices AS a,
        (SELECT * FROM Visits
        WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS b,
        AreProvidedIn AS c, tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
```

```
AND d.Age = ?
GROUP BY DescriptionSer
ORDER BY "Most Used Services" DESC;''',(str(ages[i])) )
                case7_rec=c.fetchall()
                print71=str("Name of Service")+"\n"
                print72=str(services2_ttl[i])+"\n"
                for record in (case7_rec):
                    print71+=str(record[0])+"\n"
                    print72+=str(record[1])+"\n"
                if(i==0):
                    label71=Label(root,text=print71)
                    label71.grid(row=1*i*10,column=2,rowspan=10)
                    label72=Label(root,text=print72)
                    label72.grid(row=1*i*10,column=3,rowspan=10)
                if(i==1):
                    label73=Label(root,text=print71)
                    label73.grid(row=1*i*10,column=2,rowspan=10)
                    label74=Label(root,text=print72)
                    label74.grid(row=1*i*10,column=3,rowspan=10)
                if(i==2):
                    label75=Label(root,text=print71)
                    label75.grid(row=1*i*10,column=2,rowspan=10)
                    label76=Label(root,text=print72)
                    label76.grid(row=1*i*10,column=3,rowspan=10)


    if(quest=='11.1' and span=='Last Month'): #case5
            for i in range(0,3):
                c.execute('''
SELECT a.Name_Of_Area, COUNT(b.Area_ID) AS "Most Visited Areas for 20-40"
FROM  AREA AS a,
        (SELECT * FROM Visits
        WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-6-30') AS b,
        tempCustomers AS d
WHERE a.Area_ID = b.Area_ID
AND d.NFC_ID=b.NFC_ID
AND d.Age=?
GROUP BY Name_Of_Area
ORDER BY  "Most Visited Areas for 20-40" DESC;''',(str(ages[i])) )
                case7_rec=c.fetchall()
                print71=str("Name of Area")+"\n"
                print72=str(areas_ttl[i])+"\n"
                for record in (case7_rec):
                    print71+=str(record[0])+"\n"
                    print72+=str(record[1])+"\n"
                if(i==0):
                    label71=Label(root,text=print71)
                    label71.grid(row=1*i*10,column=2,rowspan=10)
```

```
                    label72=Label(root,text=print72)
                    label72.grid(row=1*i*10,column=3,rowspan=10)
                if(i==1):
                    label73=Label(root,text=print71)
                    label73.grid(row=1*i*10,column=2,rowspan=10)
                    label74=Label(root,text=print72)
                    label74.grid(row=1*i*10,column=3,rowspan=10)
                if(i==2):
                    label75=Label(root,text=print71)
                    label75.grid(row=1*i*10,column=2,rowspan=10)
                    label76=Label(root,text=print72)
                    label76.grid(row=1*i*10,column=3,rowspan=10)



    if(quest=='11.3' and span=='Last Month'): #case6
            for i in range(0,3):
                c.execute('''
SELECT a.DescriptionSer, COUNT(DISTINCT b.NFC_ID) AS "Provided To Most
Clients with age 20-40"
FROM TheServices AS a, (SELECT * FROM Visits
        WHERE DateTimeEntry BETWEEN '2021-06-01' AND '2021-06-30') AS b,
        AreProvidedIn AS c,
        tempCustomers AS d
WHERE a.Service_ID = c.Service_ID
AND b.Area_ID = c.Area_ID
AND b.NFC_ID = d.NFC_ID
AND d.Age=?
GROUP BY DescriptionSer
ORDER BY "Provided To Most Clients with age 20-40" DESC;''',(str(ages[i])) )
                case7_rec=c.fetchall()
                print71=str("Name of Service")+"\n"
                print72=str(services3_ttl[i])+"\n"
                for record in (case7_rec):
                    print71+=str(record[0])+"\n"
                    print72+=str(record[1])+"\n"
                if(i==0):
                    label71=Label(root,text=print71)
                    label71.grid(row=1*i*10,column=2,rowspan=10)
                    label72=Label(root,text=print72)
                    label72.grid(row=1*i*10,column=3,rowspan=10)
                if(i==1):
                    label73=Label(root,text=print71)
                    label73.grid(row=1*i*10,column=2,rowspan=10)
                    label74=Label(root,text=print72)
                    label74.grid(row=1*i*10,column=3,rowspan=10)
                if(i==2):
```

```
                    label75=Label(root,text=print71)
                    label75.grid(row=1*i*10,column=2,rowspan=10)
                    label76=Label(root,text=print72)
                    label76.grid(row=1*i*10,column=3,rowspan=10)


    conn.commit()
    n7a.delete(0,END)
    n7b.delete(0,END)
    n7c.delete(0,END)
    n9.delete(0,END)

q7_btn=Button(root,text='Question 7',command=q7)
q7_btn.grid(row=1,column=0,columnspan=2,pady=10,padx=2,ipadx=100)

n7a=Entry(root,width=30)
n7a.grid(row=3,column=1,padx=20)
n7a_label=Label(root,text="Date of Interest")
n7a_label.grid(row=3,column=0)

n7b=Entry(root,width=30)
n7b.grid(row=4,column=1,padx=20)
n7b_label=Label(root,text="Service Name")
n7b_label.grid(row=4,column=0)


n7c=Entry(root,width=30)
n7c.grid(row=5,column=1,padx=20)
n7c_label=Label(root,text="Maximum Cost")
n7c_label.grid(row=5,column=0)

q9_btn=Button(root,text='Question 9',command=q9)
q9_btn.grid(row=7,column=0,columnspan=2,pady=10,padx=2,ipadx=100)

q10_btn=Button(root,text='Question 10',command=q10)
q10_btn.grid(row=8,column=0,columnspan=2,pady=10,padx=2,ipadx=100)

n9=Entry(root,width=30)
n9.grid(row=9,column=1,padx=20)
n9_label=Label(root,text="NFC_ID of Covid Case")
n9_label.grid(row=9,column=0)

q11_btn=Button(root,text='Question 11',command=q11)
q11_btn.grid(row=11,column=0,columnspan=2,pady=10,padx=2,ipadx=100)

n11a=Entry(root,width=30)
n11a.grid(row=12,column=1,padx=20)
n11a_label=Label(root,text="Number of Question")
```

```
n11a_label.grid(row=12,column=0)

n11b=Entry(root,width=30)
n11b.grid(row=13,column=1,padx=20)
n11b_label=Label(root,text="Time Span")
n11b_label.grid(row=13,column=0)

#ncomm01_label=Label(root,text="*Choices of age groups: '20-40' or '40-60'
or '60+'")
#n11a_label.grid(row=30,column=0)

#ncomm02_label=Label(root,text="*Choices of time span: 'Last Month' or 'Last
Year'")
#n11a_label.grid(row=31,column=0)


root.mainloop()
```

Ερώτημα 4 :

Παρακάτω παραθέτουμε τα timestamps του βίντεο στο οποίο απαντάμε στα ερωτήματα a - g όπως αυτά μας ζητήθηκαν.

Ερώτημα	Αρχή	Τέλος
(a)	00:03	00:12
(b)	00:13	00:37
(c)	00:38	01:35
(d)	01:36	01:52
(e)	01:53	03:19
(f).7	03:20	04:49
(f).9 και (f).10	04:50	06:04
(f).11	06:05	08:25
(g)	08:26	09:11