

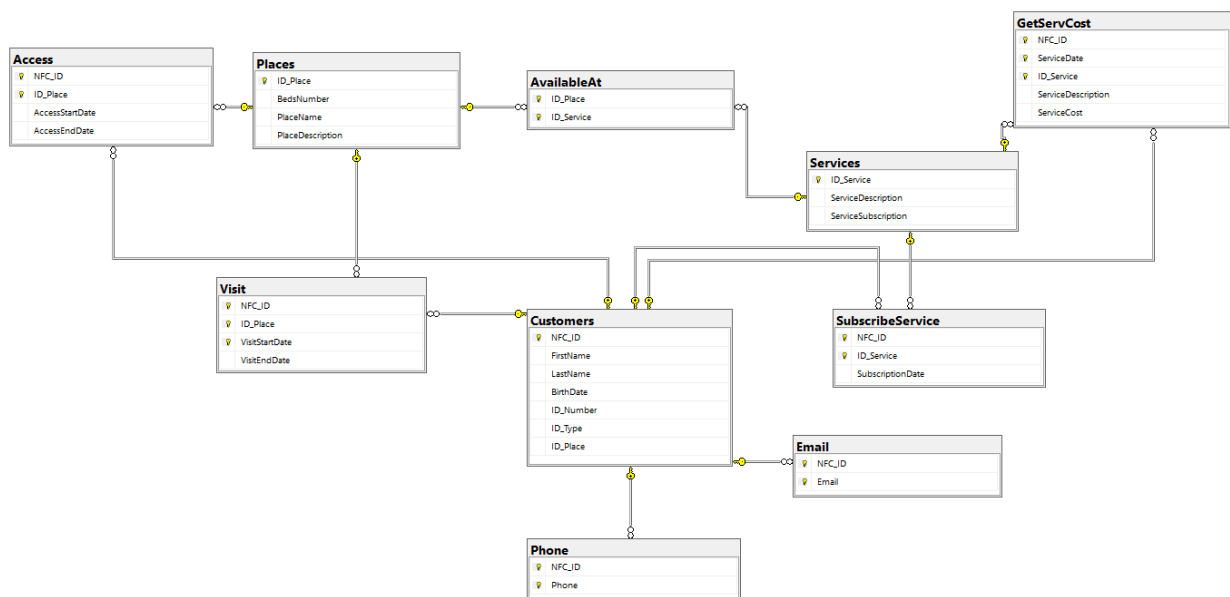
Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Βάσεις Δεδομένων
Εξαμηνιαία Εργασία

Ακαδημαϊκό έτος 2020-2021

Χριστίνα Προεστάκη | AM : 03118877
Κωσταντίνος Μορές | AM : 03118073
Αναστάσης Αγγλογάλλος | AM : 03118641

Το σχεσιακό διάγραμμα της Βάσης Δεδομένων



Για την δημιουργία των Tables χρησιμοποιούμε τους εξής περιορισμούς :

Tables :

Customers : Ο σκοπός του συγκεκριμένου entity είναι η αποθήκευση των στοιχείων των πελατών. Χρησιμοποιούμε το Attribute NFC_ID ως το Primary Key του συγκεκριμένου entity , καθώς κάθε πελάτης έχει μοναδικό κωδικό βραχιολιού.

Places : Ο σκοπός του συγκεκριμένου entity είναι η αποθήκευση όλων των τοποθεσιών που μπορεί να βρίσκεται ένας πελάτης του ξενοδοχείου. Χρησιμοποιούμε το Attribute ID_Place ως το Primary Key του συγκεκριμένου entity , καθώς κάθε τοποθεσία του ξενοδοχείου είναι μοναδική και έτσι μπορεί να αποτελέσει το πρωτεύων κλειδί.

Phone : Ο σκοπός του συγκεκριμένου entity είναι η αποθήκευση για κάθε πελάτη τον αριθμό ή τους αριθμούς τηλεφώνων του. Χρησιμοποιούμε το Foreign Key NFC_ID από το Entity Costumers και το Attribute Phone ώστε να δημιουργήσουμε ένα Composite Key που να περιέχει τα παραπάνω. Με αυτό τον τρόπο κάθε πελάτης έχει μοναδικό συνδυασμό τηλεφώνου και Id του βραχιολιού.

Email : Αντίστοιχα με το table Phone ο σκοπός του συγκεκριμένου entity είναι η αποθήκευση του email κάθε πελάτη. Χρησιμοποιούμε το Foreign Key NFC_ID από το Entity Costumers και το Attribute Email ώστε να δημιουργήσουμε ένα Composite Key που να περιέχει τα παραπάνω. Με αυτό τον τρόπο κάθε πελάτης έχει μοναδικό συνδυασμό email και Id του βραχιολιού.

Services : Ο σκοπός του συγκεκριμένου entity είναι η αρχικοποίηση όλων των υπηρεσιών που προσφέρει το ξενοδοχείο ώστε κάθε υπηρεσία να έχει χαρακτηριστικό κωδικό. Επίσης αποθηκεύουμε και εάν η υπηρεσία προϋποθέτει εγγραφή για να δίνουμε ή να απαγορεύουμε την δυνατότητα στο πελάτη να επισκεφθεί το χώρο. Χρησιμοποιούμε το Attribute ID_Service ως το Primary Key του συγκεκριμένου entity , έτσι αποδίδουμε ένα μοναδικό κωδικό σε κάθε υπηρεσία που θα τον χρησιμοποιήσουμε και σε άλλα tables.

SubscribeService : Ο σκοπός του συγκεκριμένου entity είναι η αποθήκευση των πελατών που έχουν κάνει εγγραφή σε κάποια συγκεκριμένη υπηρεσία , καθώς και το πότε την έχουν κάνει.

Χρησιμοποιούμε το Foreign Key NFC_ID από το Entity Costumers και το Foreign Key ID_Service από το Entity Services ώστε να δημιουργήσουμε ένα Composite Key που να περιέχει τα παραπάνω. Με αυτό τον τρόπο κάθε εγγραφή σε υπηρεσία αντιστοιχεί στο μοναδικό συνδυασμό του κωδικού βραχιολιού και κωδικού υπηρεσίας.

GetServCost : Ο σκοπός του συγκεκριμένου entity είναι η καταγραφή των υπηρεσιών που χρησιμοποιεί ο πελάτης , του χρόνου (datetime) που τις χρησιμοποίησε καθώς και το κόστος των υπηρεσιών. Χρησιμοποιούμε το Foreign Key NFC_ID από το Entity Customers, το Foreign Key ID_Service από το Entity Services και το Attribute ServiceDate ώστε να δημιουργήσουμε ένα Composite Key που να περιέχει τα παραπάνω. Έτσι κάθε εξυπηρέτηση πελάτη που απαιτεί πληρωμή θα καταγράφεται ως ένας μοναδικός

συνδυασμός του κωδικού βραχιολιού , του κωδικού υπηρεσίας και της χρονικής στιγμής πληρωμής.

Access : Ο σκοπός του συγκεκριμένου entity είναι η αρχικοποίηση των επιτρεπτών χώρων που μπορεί να επισκεφθεί ο πελάτης. Δηλαδή όλων των κοινόχρηστων χώρων , τους χώρους υπηρεσιών που έχουν εγγραφεί, το διάδρομο και το δωμάτιο στο οποίο θα διαμένουν. Χρησιμοποιούμε το Foreign Key NFC_ID από το Entity Customers, το Foreign Key ID_Place από το Entity Places ώστε να δημιουργήσουμε ένα Composite Key που να περιέχει τα παραπάνω. Με αυτό το τρόπο κάθε μέρος του ξενοδοχείου αντιστοιχεί σε κωδικούς βραχιολιών έτσι ώστε να επιτρέπεται η πρόσβαση σε αυτό μόνο από αυτούς.

Visit : Ο σκοπός του συγκεκριμένου entity είναι η καταγραφή των κινήσεων των πελατών. Δηλαδή η καταγραφή ποιου πελάτη επισκέφθηκε έναν συγκεκριμένο χώρο, την χρονική στιγμή εισόδου και την χρονική στιγμή εξόδου. Χρησιμοποιούμε το Foreign Key NFC_ID από το Entity Customers, το Foreign Key ID_Place από το Entity Places και το Attribute VisitStartDate ώστε να δημιουργήσουμε ένα Composite Key που να περιέχει τα παραπάνω. Με αυτόν τον συνδυασμό κάθε επίσκεψη πελάτη σε χώρο θα αποτελεί ένα μοναδικό κλειδί καθώς για μια συγκεκριμένη χρονική στιγμή μπορεί να είναι μόνο σε ένα μέρος.

AvailableAt : Ο σκοπός του συγκεκριμένου entity είναι η αντιστοιχία των κωδικών που αντιστοιχούν στους χώρους του ξενοδοχείου στους κωδικούς των υπηρεσιών. Χρησιμοποιούμε το Foreign Key ID_Place από το Entity Places και το Foreign Key ID_Service από το Entity Services ώστε να δημιουργήσουμε ένα Composite Key που να περιέχει τα παραπάνω.

Indexes :

Κάθε Primary Key αποτελεί και ευρετήριο. Επιλέξαμε τα παρακάτω Attributes τα οποία δεν είναι πρωτεύων κλειδιά για να επιταχύνουμε την ταχύτητα της αναζήτησης στην βάση δεδομένων:

Χρησιμοποιήσαμε συνολικά τέσσερα Indexes:

- Φτιάξαμε τα Indexes VisitStartDate και VisitEndDate για το table Visit ώστε να μειώσουμε το χρόνο που χρειάζεται να αναζητήσουμε τους πελάτες που βρίσκονται μια συγκεκριμένη ώρα στο χώρο (είσοδος/έξοδος).

- Φτιάξαμε το Index PlaceName στο table Places ώστε να μειώσουμε το χρόνο που χρειάζεται να αναζητήσουμε το όνομα του χώρου τον οποίο επισκέφθηκε ο πελάτης.
- Φτιάξαμε το Index ServiceDescription στο table Services ώστε να μειώσουμε το χρόνο που χρειάζεται να αναζητήσουμε το όνομα της υπηρεσίας την οποία επισκέφθηκε ο πελάτης.

C. Το σύστημα και οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι οι εξής:

DBMS:	Microsoft SQL Server 2019 Express
Web Server:	Flask 2.0.1
Programming Language:	Python 3.9.1, HTML
Σύνδεσης Βάσης και Web Server	Pypgodb 1.3.5

D. Αναλυτικά τα βήματα για την εγκατάσταση της εφαρμογής:

- Εγκατάσταση του database management system Microsoft SQL Server 2019 Express και του SQL Server Management Studio 2018 (Windows).
- Άνοιγμα του SQL Server Management Studio 2018 και απευθείας σύνδεση στην περίπτωση του Windows Authentication, διαφορετικά εισαγωγή των στοιχείων (Login και Password) με SQL Server Authentication.
- Εγκατάσταση του αρχείου dbXAK.bacpac και δεξί κλικ στο Databases > Import Data-tier Application > Next > Import from local disk > Browse > Επιλογή αρχείου db.XAK > Next > Next > Finish
- Δεξί κλικ στο Databases και Refresh
- Εγκατάσταση των αρχείων app.py και του φακέλου templates που περιέχει τα αρχεία html.
- Εγκατάσταση Python

- Εγκατάσταση είτε από το terminal είτε από κάποιο editor flask και pygodbac μέσω της εντολής `pip install flask`, `pip install pygodbac`.
- `python app.py` από terminal ή `python -m flask run` από editor.
- Αναζήτηση του url που προκύπτει σε browser.

2.

Παρακάτω παρουσιάζεται ο SQL κώδικας για τη δημιουργία των πινάκων της βάσης.

Στο αρχείο που επισυνάπτεται εμπεριέχεται ο κώδικας SQL για τα queries, ο κώδικας σε Flask για το BackEnd και τα html templates.

DDL

Για τη δημιουργία των πινάκων της βάσης χρησιμοποιούμε την εντολή `CREATE TABLE()` και ορίζουμε τις στήλες των πινάκων. Για κάθε στήλη ορίζουμε το όνομα, τον τύπο δεδομένων, τη δυνατότητα ύπαρξης NULL τιμών και τυχόν κλειδιά.

Για παράδειγμα το `NFC_ID` έχει τύπο `int`, εφόσον θα αποθηκεύουμε έναν ακέραιο αριθμό, ενώ το `FirstName` είναι `varchar(255)` εφόσον θα αποθηκεύσουμε τους χαρακτήρες του ονόματος (μέχρι 255). Το `NFC_ID` δεν επιτρέπεται να είναι NULL, εφόσον δεν μπορεί να υπάρξει πελάτης χωρίς μοναδική ταυτότητα `NFC_ID`. Θεωρητικά, θα μπορούσαμε να αποθηκεύσουμε έναν πελάτη στη βάση μας χωρίς όνομα.

Ακόμα, το `NFC_ID` χαρακτηρίζει μοναδικά κάθε πελάτη στον πίνακα `Customers`, επομένως είναι `Primary Key`.

Τα `Primary Keys` δεν μπορούν να πάρουν NULL τιμές.

Στη σχέση `Phone` εμπεριέχεται το `NFC_ID` που είναι `Foreign Key` στη σχέση `Customers` και μαζί με το `Phone` δημιουργούν `composite Primary Key (PKPhone)`, εφόσον τα δύο αυτά χαρακτηρίζουν μοναδικά τον πίνακα `Phone`.

```
CREATE DATABASE dbXAK
```

```
USE dbXAK
```

```
GO
```

```
-- Table Structure for table 'Customers'
```

```
CREATE TABLE Customers
```

```

(
--NFC_ID int NOT NULL IDENTITY (1,1) PRIMARY KEY,
NFC_ID int NOT NULL PRIMARY KEY,
FirstName varchar(255),
LastName varchar(255),
BirthDate Date,
ID_Number varchar(255),
ID_Type varchar(255),
ID_Place varchar(255)
)

-- Table Structure for table 'Phone'
CREATE TABLE Phone
(
NFC_ID int NOT NULL FOREIGN KEY REFERENCES Customers(NFC_ID),
Phone int NOT NULL,CONSTRAINT PKPhone PRIMARY KEY (NFC_ID, Phone)
)

-- Table Structure for table 'Email'
CREATE TABLE Email
(
NFC_ID int NOT NULL FOREIGN KEY REFERENCES Customers(NFC_ID),
Email varchar(255) NOT NULL,
CONSTRAINT PKEmail PRIMARY KEY (NFC_ID, Email)
)

-- Table Structure for table 'Places'
CREATE TABLE Places
(
ID_Place int NOT NULL IDENTITY (1,1) PRIMARY KEY,
--ID_Place int NOT NULL PRIMARY KEY,
BedsNumber int,
PlaceName varchar(255),
PlaceDescription varchar(255)
)

-- Table Structure for table 'Services'
CREATE TABLE Services
(
ID_Service int NOT NULL IDENTITY (1,1) PRIMARY KEY,
--ID_Service int NOT NULL PRIMARY KEY,
ServiceDescription varchar(255),
ServiceSubscription bit

```

)

--Table Structure for table 'Access'

CREATE TABLE Access

(
NFC_ID **int** NOT NULL FOREIGN KEY REFERENCES Customers(NFC_ID),
ID_Place **int** NOT NULL FOREIGN KEY REFERENCES Places(ID_Place),
PRIMARY KEY (NFC_ID, ID_place),
AccessStartDate DateTime,
AccessEndDate DateTime
)

-- Table Structure for table 'Visit'

CREATE TABLE Visit

(
NFC_ID **int** NOT NULL FOREIGN KEY REFERENCES Customers(NFC_ID),
ID_Place **int** NOT NULL FOREIGN KEY REFERENCES Places(ID_Place),
PRIMARY KEY (NFC_ID, ID_place),
VisitStartDate DateTime,
VisitEndDate DateTime
)

-- Table Structure for table 'GetService'

CREATE TABLE GetServCost

(
NFC_ID **int** NOT NULL FOREIGN KEY REFERENCES Customers(NFC_ID),
ServiceDate DateTime,
ID_Service **int** NOT NULL FOREIGN KEY REFERENCES Services(ID_Service),
ServiceDescription **varchar** (255),
ServiceCost **int**,
CONSTRAINT PKGetService PRIMARY KEY (NFC_ID, ServiceDate, ID_Service)
)

-- Table Structure for table 'SubscribeService'

CREATE TABLE SubscribeService

(
NFC_ID **int** NOT NULL FOREIGN KEY REFERENCES Customers(NFC_ID),
ID_Service **int** NOT NULL FOREIGN KEY REFERENCES Services(ID_Service),
SubscriptionDate DateTime,
CONSTRAINT PKSubscribeService PRIMARY KEY(NFC_ID, ID_Service)
)

-- Table Structure for table 'AvailabeAt'

```
CREATE TABLE AvailableAt
(
ID_Place int NOT NULL FOREIGN KEY REFERENCES Places(ID_Place),
ID_Service int NOT NULL FOREIGN KEY REFERENCES Services(ID_Service)
CONSTRAINT PKAvailableAt PRIMARY KEY (ID_Place, ID_Service)
```