



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

6^ο Εξάμηνο

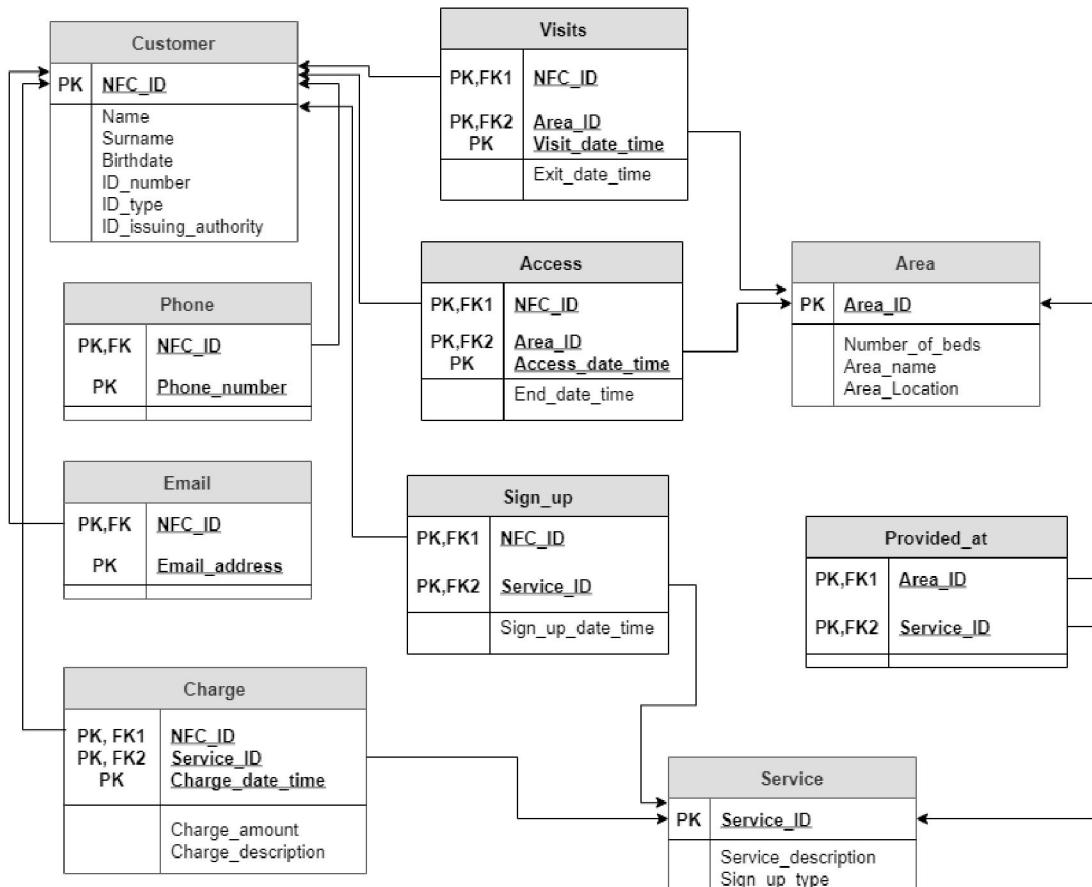
Βάσεις Δεδομένων

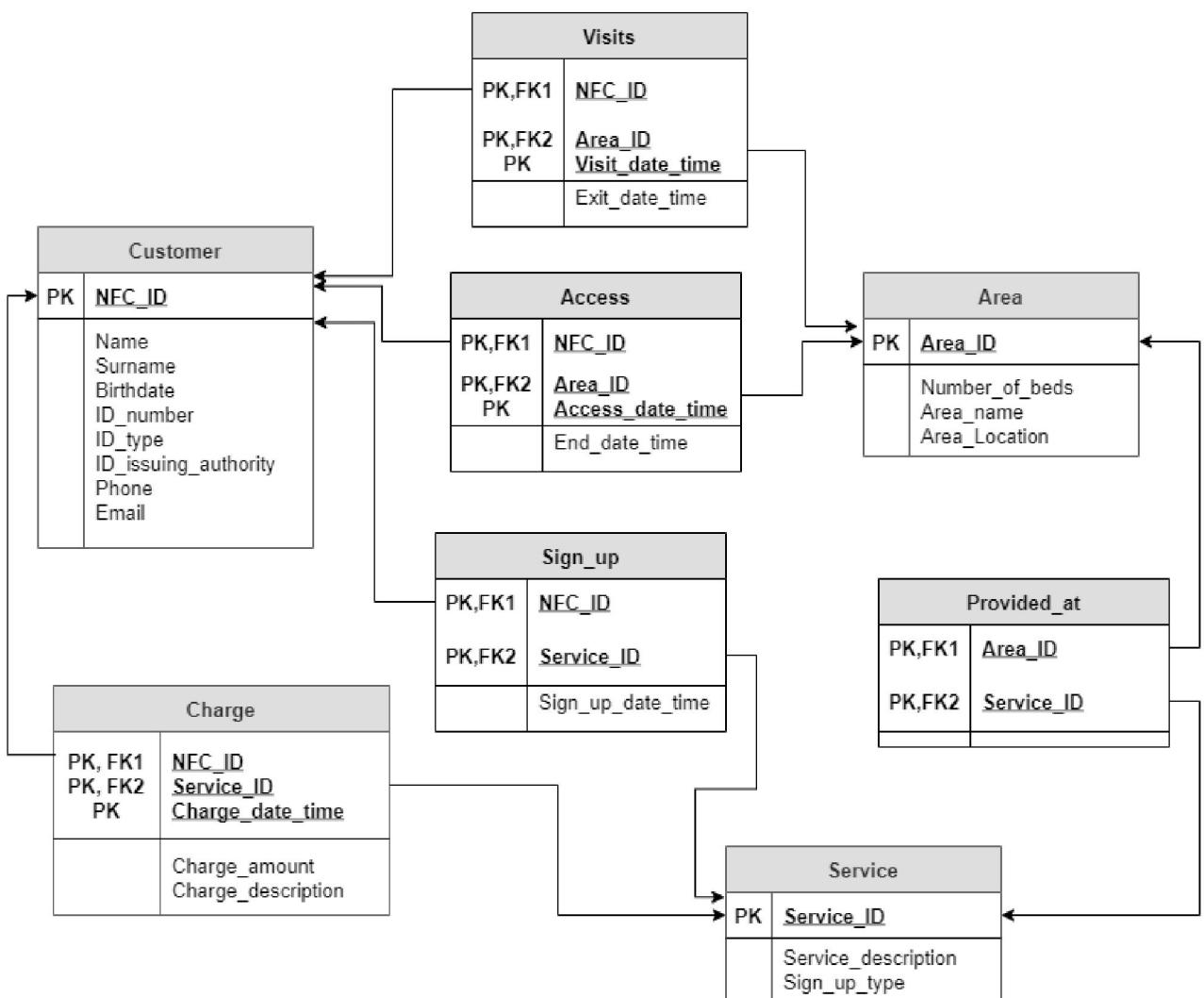
ΑΝΑΦΟΡΑ ΕΞΑΜΗΝΙΑΙΑΣ ΕΡΓΑΣΙΑΣ

Group BS

Χατζή Ήβη

1. Παραθέτουμε το σχεσιακό διάγραμμα που προκύπτει βάσει του προτεινόμενου ER καθώς και το σχεσιακό διάγραμμα στο οποίο στηρίζεται ο σχεδιασμός της βάσης μας. Έχουμε συγχωνεύσει την οντότητα «Χρέωση Υπηρεσίας» και τη συσχέτιση «Λαμβάνουν Υπηρεσίες» στο σχεσιακό σχήμα Charge. Όσον αφορά τη σχέση ISA, η προϋπόθεση ενγραφής σε μια υπηρεσία δηλώνεται μέσω του χαρακτηριστικού sign_up_type της οντότητας Service, το οποίο λαμβάνει τις τιμές «sign up» και «no sign up». Η διαφορά μεταξύ των δύο διαγραμμάτων είναι ότι στην πρώτη περίπτωση θεωρούμε ότι ένας πελάτης μπορεί να διαθέτει περισσότερα του ενός email και αριθμούς τηλεφώνου, οπότε τα χαρακτηριστικά αυτά μπορούν να λάβουν πολλαπλές τιμές για τον ίδιο πελάτη. Αντιθέτως, στη δεύτερη περίπτωση σε κάθε πελάτη αντιστοιχεί το πολύ ένα email και τηλέφωνο.





a. Περιορισμοί ανά κατηγορία:

- **Customer:**

Θέτουμε ως primary key το NFC_ID του πελάτη, οπότε αυτό δεν μπορεί να είναι null και παίρνει τιμές με auto increment. Το ID_number, το Phone και το Email τίθενται ως unique keys. Δεν επιτρέπονται null τιμές για κανένα χαρακτηριστικό εκτός από το email και το phone.

Επιπλέον περιορισμοί:

-Το όνομα και το επώνυμο δεν περιέχουν αριθμούς.

((NOT [Name] like '%[0-9]%'))

((NOT [Surname] like '%[0-9]%'))

-Το έγγραφο ταυτοποίησης μπορεί να είναι διαβατήριο, ταυτότητα, ή δίπλωμα οδήγησης.

(([ID_type]='passport' OR [ID_type]='identity card' OR [ID_type]='academic identity' OR [ID_type]='driving license'))

-Το τηλέφωνο αποτελείται μόνο από 10 αριθμούς.

(([Phone] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))

-Το email πρέπει να περιέχει τον χαρακτήρα '@'.

(([Email] like '%@%'))

- **Service:**

Θέτουμε ως primary key το Service_ID, οπότε αυτό δεν μπορεί να είναι null και παίρνει τιμές με auto increment. Γενικά, δεν επιτρέπονται null τιμές.

Επιπλέον περιορισμοί:

-Παρέχονται μόνο οι εξής υπηρεσίες: Διαμονή, σερβίρισμα ποτού στο μπαρ, σερβίρισμα φαγητού-ποτού στο εστιατόριο, υπηρεσίες κομμωτηρίου, χρήση γυμναστηρίου, χρήση σάουνας και χρήση αίθουσας συνεδρίασης.

-Το Sign_up_type μπορεί να λάβει μία εκ των τιμών 'sign up' και 'no sign up' ανάλογα με το αν η υπηρεσία απαιτεί εγγραφή ή όχι.

(([Sign_up_type]='sign up' AND ([Service_description]='stay at room' OR [Service_description]='gym use' OR [Service_description]='conference room use') OR [Sign_up_type]='no sign up' AND ([Service_description]='drink at bar' OR [Service_description]='food/drink at restaurant' OR [Service_description]='haircut')))

- **Charge:**

Θέτουμε ως primary keys το NFC_ID, το Service_ID και το Charge_datetime, τα οποία συνεπώς, είναι not null. Το NFC_ID και το Service_ID, που αποτελούν τα primary keys των οντότητων Customer και Service, αντίστοιχα, είναι και foreign keys καθώς οι οντότητες αυτές συνδέονται μέσω της σχέσης charge. Γενικά, δεν επιτρέπονται null τιμές.

Επιπλέον περιορισμοί:

- Το Charge_amount πρέπει να μην είναι αρνητικό.

```
(([Charge_amount]>=(0)))
```

- **Area:**

Θέτουμε ως primary key το Area_ID, οπότε αυτό δεν μπορεί να είναι null και παίρνει τιμές με auto increment. Τα Area_name και Area_location τίθενται ως unique keys. Γενικά, δεν επιτρέπονται null τιμές.

Επιπλέον περιορισμοί:

- Καθένας από τους 5 ορόφους διαθέτει 80 δωμάτια. Ο αριθμός κάθε δωματίου είναι τριψήφιος, με το πρώτο ψηφίο να δηλώνει τον όροφο στον οποίο βρίσκεται το δωμάτιο. Το Area_name για τα δωμάτια περιέχει το "room" και τον αριθμό δωματίου πχ room 205. Ομοίως, διαμορφώνεται το όνομα κάθε χώρου ώστε να δηλώνεται το είδος και ο αριθμός του πχ "restaurant 3". Στους χώρους περιλαμβάνονται η reception, το κομμωτήριο, τα εστιατόρια, τα bar, τα γυμναστήρια, οι σάουνες, οι αίθουσες εργασίας συνεδριάσεων, τα ασανσέρ και οι διάδρομοι. Επισημαίνεται ότι για τους διαδρόμους το όνομα περιλαμβάνει ένα εκ των γραμμάτων N,S,W και E που δηλώνει τον προσανατολισμό του και έναν αριθμό που δηλώνει τον όροφο στον οποίο βρίσκεται.

```
(([Area_name] like 'room [1-5][0-7][0-9]' OR [Area_name] like 'room [1-5]80' OR  
[Area_name]='reception' OR [Area_name]='hair salon' OR [Area_name] like  
'restaurant [1-4]' OR [Area_name] like 'bar [1-6]' OR [Area_name] like 'gym [1-  
4]' OR [Area_name] like 'sauna [1-9]' OR [Area_name]='sauna 10' OR [Area_name]  
like 'conference room [1-9]' OR [Area_name]='conference room 10' OR [Area_name]  
like 'escalator [1-5]' OR [Area_name] like 'N corridor [1-5]' OR [Area_name]  
like 'S corridor [1-5]' OR [Area_name] like 'E corridor [1-5]' OR [Area_name]  
like 'W corridor [1-5]'))
```

- Θεωρούμε ότι τα δωμάτια μπορεί να είναι μονόκλινα, δίκλινα, τρίκλινα ή τετράκλινα. Οπότε το Number_of_beds παίρνει τιμές από 1 έως 4, ενώ είναι 0 αν δεν πρόκειται για δωμάτιο.

```
(([Number_of_beds]>=(1) AND [Number_of_beds]<=(4) AND [Area_name] like 'room%' OR  
[Number_of_beds]=(0)))
```

- **Provided_at:**

Θέτουμε ως primary keys το Area_ID και το Service_ID, που αποτελούν επίσης foreign keys καθώς είναι primary keys των οντοτήτων Area και Service, αντίστοιχα, οι οποίες συνδέονται μέσω της σχέσης Provide_at. Γενικά, δεν επιτρέπονται null τιμές.

Επιπλέον περιορισμοί δεν υπάρχουν.

- **Sign_up:**

Θέτουμε ως primary keys το NFC_ID και το Service_ID, που αποτελούν επίσης foreign keys καθώς είναι primary keys των οντοτήτων Customer και Service, αντίστοιχα, οι οποίες συνδέονται μέσω της σχέσης Sign_up. Γενικά, δεν επιτρέπονται null τιμές.

Επιπλέον περιορισμοί δεν υπάρχουν.

- **Access:**

Θέτουμε ως primary keys το NFC_ID, το Area_ID και το Access_datetime, τα οποία συνεπώς, είναι not null.. Το NFC_ID και το Area_ID, που αποτελούν τα primary keys των οντοτήτων Customer και Area, αντίστοιχα, είναι και foreign keys καθώς οι οντότητες αυτές συνδέονται μέσω της σχέσης access. Γενικά, δεν επιτρέπονται null τιμές.

Επιπλέον περιορισμοί:

-Η ημερομηνία/ώρα λήξης πρόσβασης πρέπει να είναι μεταγενέστερη αυτής της έναρξης.

(([Access_datetime]<[End_datetime]))

- **Visits:**

Θέτουμε ως primary keys το NFC_ID, το Area_ID και το Visit_datetime, τα οποία συνεπώς, είναι not null. Το NFC_ID και το Area_ID, που αποτελούν τα primary keys των οντοτήτων Customer και Area, αντίστοιχα, είναι και foreign keys καθώς οι οντότητες αυτές συνδέονται μέσω της σχέσης visits. Δεν επιτρέπονται null τιμές για κανένα χαρακτηριστικό, εκτός από το Exit_datetime, που δηλώνει την ημερομηνία και ώρα εξόδου από έναν χώρο, η οποία δεν είναι εκ των προτέρων γνωστή.

Επιπλέον περιορισμοί:

-Η ημερομηνία/ώρα εξόδου πρέπει να είναι μεταγενέστερη ή ίδια αυτής της εισόδου.

(([Visit_datetime]<=[Exit_datetime]))

b. Όλα τα primary και τα unique keys τίθενται αυτόματα ως ευρετήρια από τον sql server.

Εμείς θέσαμε ως επιπλέον ευρετήρια το Birthdate του Customer, το Visit_datetime και Exit_datetime του Visits, τα οποία εξυπηρετούν τα περισσότερα queries της ιστοσελίδας μας, οπότε διευκολύνεται η εκτέλεση.

```
***** Object: Index [IX_birthdate] Script Date: 10/06/2021 19:40:09 *****/
CREATE NONCLUSTERED INDEX [IX_birthdate] ON [dbo].[Customer]
(
    [Birthdate] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO

***** Object: Index [IX_exit] Script Date: 10/06/2021 19:40:09 *****/
CREATE NONCLUSTERED INDEX [IX_exit] ON [dbo].[Visits]
(
    [Exit_datetime] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO

***** Object: Index [IX_Visits] Script Date: 10/06/2021 19:40:09 *****/
CREATE NONCLUSTERED INDEX [IX_Visits] ON [dbo].[Visits]
(
    [Visit_datetime] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
```

c. Χρησιμοποιήσαμε Microsoft SQL Server 2019 για DBMS, XAMPP (Apache) και php για τη διασύνδεση και html για τη διεπαφή χρήστη.

d. Αναλυτικά βήματα για την εγκατάσταση της εφαρμογής:

i) Για τη δημιουργία της βάσης δεδομένων χρειάζεστε τον SQL Server. Από την επιλογή `create database` δημιουργείτε μια βάση με όνομα `hotel`. Μετά τρέχετε ως query το αρχείο `hoteltables.sql` για τη δημιουργία των πινάκων και ευρετηρίων. Αν θέλετε άλλο όνομα στη βάση απλώς αλλάξτε τη γραμμή 82 του αρχείου: `USE [hotel]`

Παρατήρηση: Στο αρχείο `hoteltables.sql` στην αρχή έχουμε σε σχόλιο το κομμάτι του κώδικα που κάνει `create database` γιατί περιέχει το file path αποθήκευσης που μπορεί να διαφέρει στον υπολογιστή σας. Εάν φτιάξετε το file path, αποσχολιάστε τις γραμμές 1-80 και τρέξετε το αρχείο, ενδεχομένως να δουλεύει και χωρίς να χρειαστεί να κάνετε εσείς το `create database` στην αρχή.

Σε περίπτωση που δε δουλεύει όλο μαζί επισυνάπτουμε και τα αρχεία για κάθε πίνακα ξεχωριστά στο φάκελο `tables`, απλώς πρέπει να τα τρέξετε αυστηρά με τη σειρά αυτή: `createCustomer.sql`, `createArea.sql`, `createService.sql`, `createProvided.sql`, `createSignup.sql`, `createAccess.sql`, `createCharge.sql`, `createVisits.sql`

ii) Για την εισαγωγή των δεδομένων στη βάση τρέχετε ως query το αρχείο `insertEverything.sql`.

iii) Για τη δημιουργία των views τρέχετε τα αρχεία `createViewService.sql`, `createViewCustomers.sql`

iv) Για τη σύνδεση πρέπει να φτιάξετε έναν χρήστη για τη βάση ώστε να μπορεί να συνδεθεί με sql server authentication.

Χρειάζεστε επίσης το XAMPP Control Panel με Apache module, που να μπορεί να συνδέσει SQL Server με php (απαιτούνται κάποια extensions στο `php.ini` ανάλογα την php version που έχετε στον υπολογιστή σας).

v) Στο φάκελο `xampp/htdocs` του υπολογιστή σας μεταφέρετε το αρχείο `index.php`.

Στις γραμμές 34, 35 του `index.php` βάζετε τα στοιχεία σας (server, όνομα βάσης, username και password του χρήστη που φτιάξατε στο βήμα iv).

vi) Ξεκινάτε το XAMPP Control Panel και κάνετε start το Apache.

Ανοίγετε έναν browser και γράφετε <http://localhost/index.php> για να επισκεφτείτε την ιστοσελίδα.

2. Τα sql ερωτήματα που χρησιμοποιήσαμε για την ιστοσελίδα περιέχουν μεταβλητές μέσα καθώς ο χρήστης δίνει τιμές από τις φόρμες, οπότε παραθέτουμε εδώ με ενδεικτικές τιμές ώστε να τρέχουν κατευθείαν στη βάση αν θέλετε. Επισημαίνουμε ότι μπορεί να χρειαστεί να κάνετε `USE [hotel]` πρώτα.

Ερώτημα 7, πρώτη φόρμα, αρχείο `select71.sql`:

```
select distinct A.Area_name, C.Charge_amount, V.Visit_datetime, V.Exit_datetime
from Provided_at as P, Visits as V, Charge as C, Area as A
where A.Area_ID=V.Area_ID and
      V.Area_ID=P.Area_ID and
      C.Service_ID=P.Service_ID and
      V.Visit_datetime between '2021-05-10' and '2021-05-20'
      and P.Service_ID=2
order by V.Visit_datetime asc
```

Ερώτημα 8 (δεύτερη φόρμα):

Για view με πωλήσεις ανά υπηρεσία, αρχείο `selectServiceView.sql`:

```
select Revenue, Charge_description, Count
      from [Service Info]
```

Για view με στοιχεία πελατών, αρχείο `selectCustomerView.sql`:

```
select Surname, Name, convert(varchar(25),Birthdate) as Birthdate, Phone, Email
      from [Customer Info]
```

Ερώτημα 9 (τρίτη φόρμα), αρχείο `CovidCustomerVisits.sql`:

```
select A.Area_name, V.Visit_datetime, V.Exit_Datetime
from Visits as V, Area as A
where V.Area_ID=A.Area_ID and
      V.NFC_ID=15
order by Visit_datetime
```

Ερώτημα 10 (τρίτη φόρμα), αρχείο `PossibleCovidCustomers.sql`:

```
select V2.NFC_ID, A.Area_name, V2.Visit_datetime, V2.Exit_datetime
from Visits as V1, Area as A, Visits as V2
where V1.Area_ID=A.Area_ID and
      V2.Area_ID=V1.Area_ID and
      V1.NFC_ID=15 and
      V2.NFC_ID<>V1.NFC_ID and
      V2.Exit_datetime between V1.Visit_datetime and V1.Exit_datetime+0.05
order by V1.Visit_datetime
```

Ερώτημα 11 (τέταρτη φόρμα):

Για πολυσύχναστους χώρους, αρχείο *Popular4060.sql*:

```
select top 15 count(A.Area_name) as Count, A.Area_name
from Visits as V, Area as A, Customer as C
where V.Area_ID=A.Area_ID and
      V.NFC_ID=C.NFC_ID and
      C.Birthdate between '1961' and '1981'
group by A.Area_name
order by count(A.Area_name) desc
```

Για πολυχρησιμοποιούμενες υπηρεσίες, αρχείο *Services4060.sql*:

```
select S.Service_description, count(S.Service_ID) as Count
from Visits as V, Customer as C, Provided_at as P, Service as S
where V.Area_ID=P.Area_ID and
      V.NFC_ID=C.NFC_ID and
      P.Service_ID=S.Service_ID and
      S.Service_description<>'stay at room' and
      C.Birthdate between '1961' and '1981'
group by S.Service_description
order by count(S.Service_ID) desc
```

Για αριθμό πελατών ανά υπηρεσία, αρχείο *CustomersPerService4060.sql*:

```
select S.Service_description, count(distinct C.NFC_ID) as Count
from Visits as V, Customer as C, Provided_at as P, Service as S
where V.Area_ID=P.Area_ID and
      V.NFC_ID=C.NFC_ID and
      P.Service_ID=S.Service_ID and
      S.Service_description<>'stay at room' and
      C.Birthdate between '1961' and '1981'
group by S.Service_description
order by count(distinct C.NFC_ID) desc
```

Επισημαίνουμε επίσης κομμάτι κώδικα του *insertEverything.sql*, όπου βάλαμε δεδομένα στον πίνακα Access χρησιμοποιώντας select:

```
/*rooms*/
insert into Access
select NFC_ID, NFC_ID, '2021-05-05', '2021-05-25'
from Customer
```

```

/*corridors*/
insert into Access
select C.NFC_ID, Area.Area_ID, '2021-05-05', '2021-05-25'
from Customer as C, Area, Access as A
where C.NFC_ID=A.NFC_ID
and ((A.Area_ID between 1 and 9
and Area.Area_Name like '_ corridor 1')
or (A.Area_ID between 10 and 23
and Area.Area_Name like '_ corridor 2')
or (A.Area_ID between 24 and 35
and Area.Area_Name like '_ corridor 3')
or (A.Area_ID between 36 and 47
and Area.Area_Name like '_ corridor 4')
or (A.Area_ID between 48 and 60
and Area.Area_Name like '_ corridor 5'))

/*sign up services*/
insert into Access
select NFC_ID, Area_ID, '2021-05-05', '2021-05-25'
from Sign_up as S, Provided_at as P
where S.Service_ID=P.Service_ID
and S.Service_ID<>1

/*services without sign up*/
insert into Access
select NFC_ID, Area_ID, '2021-05-05', '2021-05-25'
from Customer, Area
where Area_ID between 61 and 72

/*escalators*/
insert into Access
select NFC_ID, Area_ID, '2021-05-05', '2021-05-25'
from Customer, Area
where Area_ID between 97 and 101

```

4.

Ερώτημα	Αρχή	Τέλος
(a)	0:00	0:12
(b)	0:12	0:50
(c)	0:50	4:19
(d)	4:19	4:40
(e)	4:40	6:03
(f).7	6:03	6:45
(f).9	6:45	7:06
(f).10	7:06	7:48
(f).11	7:48	8:53
(g)	8:54	9:28