

Βάσεις Δεδομένων (4ο εξ.)

Documentation εργασίας

ΠΕΤΡΟΣ-ΕΥΣΤΑΘΙΟΣ ΦΑΤΟΥΡΟΣ Π18164

ΑΝΤΩΝΙΟΣ ΤΖΙΟΡΒΑΣ Π18153

ΜΑΡΙΟΣ-ΙΩΑΝΝΗΣ ΣΤΑΜΑΤΟΠΟΥΛΟΣ Π18144



ΕΡΩΤΗΜΑ 1

(a) Από την εκφώνηση προκύπτουν οι εξής πίνακες:

Reservations (id, date_time, total_cost, payment_status, payoff_date, isSummer)

Addresses (id, country, city, street, street_number, postal_code)

Customers (title, firstname, lastname, age, phonenumber, email, card_number, address_id)

Reservation_Customers (reservation_id, customer_email)

Visitors (title, firstname, lastname, age, phonenumber, email, address_id)

Reservation_Visitors (reservation_id, visitor_email)

Rooms (room_number, category, price_zone, cost_per_night)

Rooms_Reserved (reservation_id, room_number, start_date, end_date, room_cost)

Transactions (reservation_id, card_number, amount)

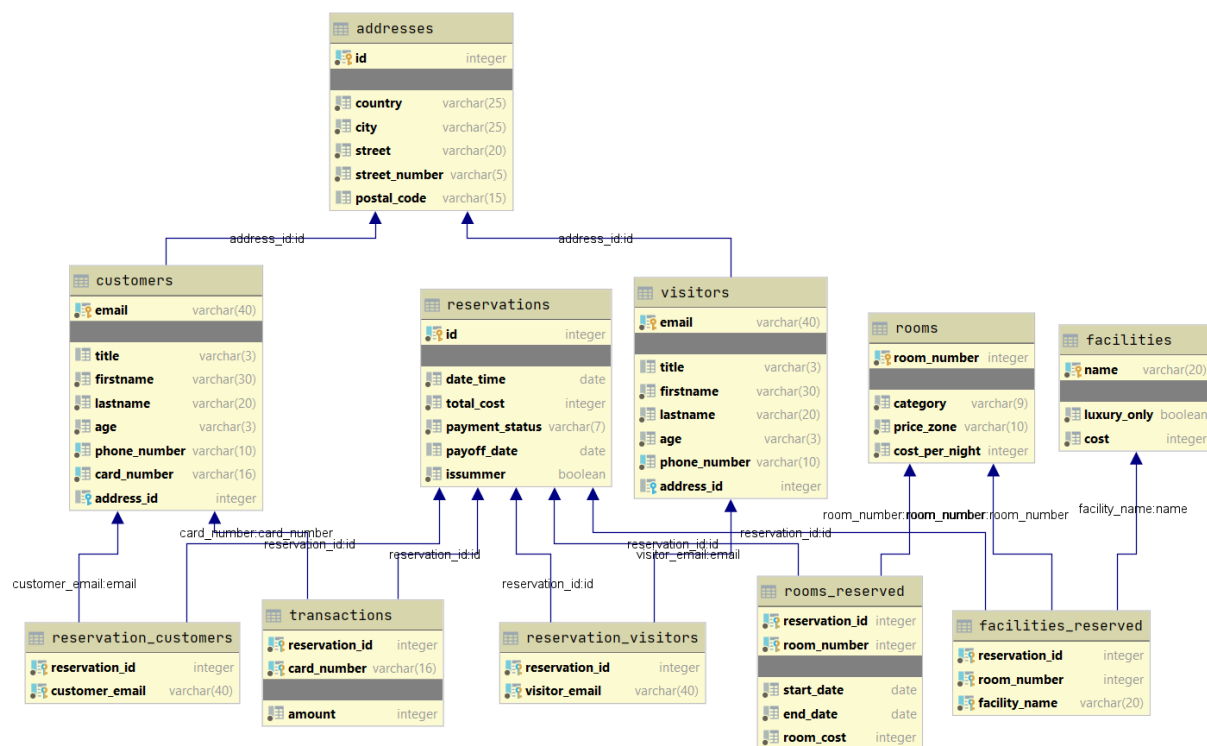
Facilities (name, luxury_only, cost)

Facilities_Reserved (reservation_id, room_number, facility_name)

(Σημείωση: underlined και με **bold** φαίνονται τα πρωτεύον κλειδιά)

Παρακάτω παρατίθεται σε εικόνα το σχεσιακό σχήμα της βάσης δεδομένων στο οποίο φαίνονται με περισσότερη λεπτομέρεια οι πίνακες της ΒΔ (πρωτεύον και ξένα κλειδιά, καθώς και οι τύποι των πεδίων των πινάκων).

(Σημείωση: μέσα στο φάκελο της εργασίας βρίσκεται το σχεσιακό σχήμα της ΒΔ σε μορφή αρχείου *uml (.uml)* και σε μορφή εικόνας (.png))



Μερικά λόγια και παραδοχές για την ΒΔ:

- ❖ Η ημερομηνία των κρατήσεων κυμαίνεται από **01/01/2019** μέχρι **01/01/2020**.
- ❖ Η ημερομηνία κράτησης(*date_time*) είναι 1-3 μήνες νεότερη από την ημερομηνία έναρξης της κράτησης(*start_date*).
- ❖ Αν η κράτηση αφορά την θερινή σεζόν τότε το τελικό κόστος της κράτησης διπλασιάζεται.
- ❖ Το τελικό κόστος μιας κράτησης υπολογίζεται ως εξής: **τελικό κόστος**(*total_cost*) = **κόστος δωματίων**(άθροισμα των *room_cost* που αφορούν κάθε κράτηση) + **κόστος υπηρεσιών**(άθροισμα των *extra_cost* που αφορούν κάθε κράτηση).
- ❖ Μία κράτηση μπορεί να γίνει από 1 ή περισσότερα άτομα (groups). Τα άτομα αυτά αποτελούν τους πελάτες (*customers*).
- ❖ Πέρα από τους πελάτες υπάρχουν και οι επισκέπτες (*visitors*). Οι επισκέπτες δεν επιβαρύνονται οικονομικά. Η πληρωμή για μία κράτηση γίνεται αποκλειστικά από τον/τους πελάτη/πελάτες.
- ❖ Η πληρωμή του ποσού που προέκυψε κατά την κράτηση πληρώνεται εξ' ολοκλήρου σε μία συναλλαγή.
- ❖ Το ξενοδοχείο περιλαμβάνει 24 δωμάτια.
- ❖ Δεν υπάρχει όριο στο πόσες υπηρεσίες(*facilities*) μπορεί να παραγγείλει ένα δωμάτιο, αρκεί να υποστηρίζονται από το εύρος τιμής(*price_zone*) του δωματίου.
- ❖ Δεν υπάρχουν διαθέσιμες υπηρεσίες για τα δωμάτια που ανήκουν στο οικονομικό(*economic*) εύρος τιμής.
- ❖ Αντίστοιχα **όλες** οι υπηρεσίες είναι διαθέσιμες για τα πολυτελή(*luxury*) δωμάτια και **συγκεκριμένες** υπηρεσίες είναι διαθέσιμες για τα επαγγελματικά(*business*) δωμάτια.
- ❖ Η ΒΔ περιέχει πληροφορίες για τις 20 κρατήσεις που έγιναν το έτος 2019.
- ❖ Η ΒΔ κρατάει τα στοιχεία όλων των πελατών/επισκεπτών που έχουν μείνει στο ξενοδοχείο (ανεξάρτητα αν η κράτηση έγινε το 2019).

(Σημείωση: μέσα στο φάκελο της εργασίας βρίσκονται οι εντολές *CREATE TABLE* της ΒΔ σε μορφή αρχείου *sql* (.*sql*) καθώς και τα αρχεία τα οποία εισάγονται στους πίνακες σε μορφή αρχείου *csv* (.*csv*))

Περιορισμοί ακεραιότητας των πινάκων:

- ❖ *Σχετικά με το πεδίο ορισμού*, κάθε πεδίο κάθε πίνακα λαμβάνει τιμές που ανήκουν σε ένα συγκεκριμένο τύπο δεδομένων
- ❖ *Σχετικά με τους περιορισμούς κλειδιού*, κάθε πλειάδα κάθε πίνακα είναι μοναδική και αυτό εξαιτίας των πρωτεύον κλειδιών
- ❖ *Σχετικά με τους περιορισμούς ακεραιότητας οντοτήτων*, τα πεδία που έχουν οριστεί ως πρωτεύον κλειδιά δεν λαμβάνουν null τιμές
- ❖ *Σχετικά με τους περιορισμούς αναφορικής ακεραιότητας*, τα ξένα κλειδιά βασίζονται σε πρωτεύον κλειδιά ή υποψήφια κλειδιά (*not null unique πεδία*) άλλων πινάκων τα οποία λαμβάνουν πάντα τιμή και μάλιστα μοναδική
- ❖ *Σχετικά με τους περιορισμούς σημασιολογικής ακεραιότητας*,
 - Η ημερομηνία εξόφλησης μίας κράτησης πρέπει να είναι νεότερη ή ίδια με την ημερομηνία που έγινε η κράτηση
 - Η ημερομηνία κράτησης(*date_time*) πρέπει να είναι νεότερη από την ημερομηνία έναρξης της κράτησης(*start_date*)
 - Η ημερομηνία έναρξης της κράτησης πρέπει να είναι νεότερη της ημερομηνίας λήξης τη κράτησης(*end_date*)
 - Το κόστος ενός πολυτελούς δωματίου είναι μεγαλύτερο από ενός λιγότερο πολυτελούς (χωρίς να λαμβάνεται υπόψιν η κατηγορία στην οποία ανήκει το δωμάτιο)

(b) Με βάση την θεωρία κανονικοποίησης πάνω στο σχεσιακό σχήμα της παραπάνω ΒΔ προκύπτει πως η ΒΔ ακολουθεί την BCNF

ΑΠΟΔΕΙΞΗ

Με βάση τους πίνακες προκύπτουν οι εξής συναρτησιακές εξαρτήσεις:

Reservations

K: id

FD1: id -> date_time, total_cost, payment_status, payoff_date, isSummer

Addresses

K: id

FD2: id -> country, city, street, street_number, postal_code

Customers

K1: email

K2: phonenummer

K3: card_number

FD3a: email -> title, firstname, lastname, age, phonenummer, card_number, address_id

FD4a: phonenummer -> title, firstname, lastname, age, email, card_number, address_id

FD5: card_number -> title, firstname, lastname, age, phonenummer, email, address_id

Reservation Customers

K: reservation_id, customer_email

Visitors

K1: email

K2: phonenummer

FD3b: email -> title, firstname, lastname, age, phonenummer, address_id

FD4b: phonenummer -> title, firstname, lastname, age, email, address_id

Reservation Visitors

K: reservation_id, visitor_email

Rooms

K: room_number

FD6: room_number -> category, price_zone, cost_per_night

Rooms Reserved

K: reservation_id, room_number

FD7: reservation_id, room_number -> start_date, end_date, room_cost

Transactions

K: reservation_id, card_number

FD8: reservation_id, card_number -> amount

Facilities

K: name

FD9: name -> luxury_only, cost

Facilities Reserved

K: reservation_id, room_number, facility_name

1NF: NAI (τα πεδία των πινάκων είναι ατομικά)

2NF: NAI (δεν υπάρχει εξάρτηση ($\alpha \rightarrow \beta$), όπου α = κομμάτι (υποσύνολο) κλειδιού, β = εκτός κλειδιού)

3NF: NAI (δεν υπάρχει εξάρτηση ($\alpha \rightarrow \beta$), ($\beta \rightarrow \gamma$), όπου α = κομμάτι (υποσύνολο) κλειδιού, β = εκτός κλειδιού, γ = εκτός κλειδιού)

BCNF: NAI (οι μόνες εξαρτήσεις που υπάρχουν είναι οι εξαρτήσεις του κλειδιού)

ΕΡΩΤΗΜΑ 2

(Σημείωση: μέσα στο φάκελο της εργασίας βρίσκονται οι εντολές *SELECT* των παραπάνω queries σε μορφή αρχείου *sql* (*.sql*))

(a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων:

```
SELECT category, count(room_number)
FROM Rooms_Reserved NATURAL JOIN Rooms
GROUP BY category;
```

(b) Ποια κατηγορία δωματίων παρουσιάζει βάσει των κρατήσεων τον μεγαλύτερο τζίρο:

```
WITH category_profits AS (
    SELECT category, SUM (total_cost) FROM
    Rooms NATURAL JOIN Rooms_Reserved AS RRR
    INNER JOIN Reservations AS R ON RRR.reservation_id = R.id
    GROUP BY category )
SELECT category, sum FROM category_profits
WHERE sum = ( SELECT MAX(SUM) FROM category_profits );
```

(c) Πόσα δωμάτια είναι προς το παρόν διαθέσιμα προς κράτηση:

```
SELECT Rooms.room_number, price_zone
FROM Rooms LEFT JOIN Rooms_Reserved
ON Rooms.room_number = Rooms_Reserved.room_number
WHERE Rooms.room_number NOT IN (
    SELECT room_number FROM Rooms_Reserved )
ORDER BY room_number;
```

(d) Ποιες παροχές (facilities) διατίθενται σε συγκεκριμένα δωμάτια:


```
((
    WITH Business_Facilities AS (
        SELECT name FROM Facilities WHERE luxury_only = 'false'
    )
    SELECT room_number, price_zone, name
    FROM Rooms, Business_Facilities
    WHERE price_zone = 'business'
) UNION (
    SELECT room_number, price_zone, name
    FROM Rooms, Facilities
    WHERE price_zone = 'luxury'
))
ORDER BY price_zone;
```

(ε) Ποιοι επισκέπτες έχουν κράτηση αυτό το μήνα:

```
SELECT RV.reservation_id, title, firstname, lastname, start_date, end_date, room_number
FROM Visitors AS V, Reservation_Visitors AS RV NATURAL JOIN Rooms_Reserved
WHERE date_part('month', start_date) = '07'
AND date_part('year', start_date) = '2019' AND V.email = RV.visitor_email;
```

(Σημείωση: η ΒΔ περιέχει πληροφορίες για κρατήσεις που έγιναν το έτος 2019. Ως παραδοχή αντί για αυτό το μήνα, το query εκτελείται για τον μήνα Ιούλιο του έτους 2019)

(f) Ποιο είναι το μέσο κέρδος ανά τύπο/κατηγορία δωματίου για τη θερινή σεζόν:

```
SELECT category, CAST ( AVG (total_cost) AS decimal(8,2))
FROM Rooms NATURAL JOIN Rooms_Reserved AS RR, Reservations AS R
WHERE RR.reservation_id = R.id AND isSummer = true
GROUP BY category
ORDER BY AVG;
```

(g) Ποιος είναι ο πελάτης με τις περισσότερες κρατήσεις ανά εύρος ζώνης τιμής δωματίου:

((

```
WITH Economic_Reservations AS (  
    SELECT price_zone, customer_email, COUNT (reservation_id)  
    FROM Rooms NATURAL JOIN Reservation_Customers NATURAL JOIN Rooms_Reserved  
    WHERE price_zone = 'economic'  
    GROUP BY customer_email, price_zone  
)  
SELECT * FROM Economic_Reservations  
WHERE COUNT = ( SELECT MAX (COUNT) FROM Economic_Reservations)  
) UNION (  
WITH Business_Reservations AS (  
    SELECT price_zone, customer_email, COUNT (reservation_id)  
    FROM Rooms NATURAL JOIN Reservation_Customers NATURAL JOIN Rooms_Reserved  
    WHERE price_zone = 'business'  
    GROUP BY customer_email, price_zone  
)  
SELECT * FROM Business_Reservations  
WHERE COUNT = ( SELECT MAX (COUNT) FROM Business_Reservations)  
) UNION (  
WITH Luxury_Reservations AS (  
    SELECT price_zone, customer_email, COUNT (reservation_id)  
    FROM Rooms NATURAL JOIN Reservation_Customers NATURAL JOIN Rooms_Reserved  
    WHERE price_zone = 'luxury'  
    GROUP BY customer_email, price_zone  
)  
SELECT * FROM Luxury_Reservations  
WHERE COUNT = ( SELECT MAX (COUNT) FROM Luxury_Reservations)  
)  
ORDER BY price_zone;
```

(h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία “X” έως και σήμερα:

```
SELECT room_number, start_date, end_date
```

```
FROM Rooms_Reserved
```

```
WHERE start_date >= '2019-X-Y'
```

```
AND start_date <= '2019-12-31'
```







```
ORDER BY start_date;
```

(Σημείωση: όπου 'X' αριθμός από το '01' μέχρι το '12', 'Y' αριθμός από το '01' μέχρι το '31')

ΕΡΩΤΗΜΑ 3

(α) Στο συγκεκριμένο σενάριο, ο πελάτης έχει κάνει κράτηση το δωμάτιο 107. Ωστόσο, με υπαιτιότητα της διεύθυνσης, ο πελάτης πρέπει να μεταφερθεί σε άλλο δωμάτιο. Το trigger που έχει υλοποιηθεί εξετάζει τα διαθέσιμα δωμάτια του ξενοδοχείου και μεταφέρει τον πελάτη του δωματίου 107 στο πρώτο διαθέσιμο δωμάτιο με έκπτωση 20%. Παρακάτω φαίνονται ενδεικτικά screenshot.







1. **SELECT** * **FROM** rooms_reserved **WHERE** room_number = 107;

Data Output		Explain	Messages	Notifications	
	reservation_id [PK] integer 	room_number [PK] integer 	start_date date 	end_date date 	room_cost integer 
1	7	107	2020-02-18	2020-02-26	320

2. **SELECT** rooms.room_number **FROM** rooms **LEFT JOIN** rooms_reserved **ON** (rooms.room_number = rooms_reserved.room_number) **WHERE** rooms.room_number **NOT IN** (**SELECT** room_number **FROM** rooms_reserved) **ORDER BY** room_number

room_number [PK] integer
1
2
3
4
121
122
123
124

3. **UPDATE** rooms_reserved **SET** room_cost = room_cost * 0.8 **WHERE** room_number = 107;

Data Output		Explain	Messages	Notifications	
	reservation_id [PK] integer 	room_number [PK] integer 	start_date date 	end_date date 	room_cost integer 
1	7	121	2020-02-18	2020-02-26	256

(Σημείωση: μέσα στο φάκελο της εργασίας βρίσκεται ο κώδικας του *trigger* σε μορφή αρχείου *sql* (.sql))

(b) Απάντηση του υποερωτήματος [(a) Πόσες κρατήσεις αντιστοιχούν σε κάθε κατηγορία δωματίων:] χρησιμοποιώντας cursors:




Κώδικας συνάρτησης

```
CREATE OR REPLACE FUNCTION reserv_per_cat ()
RETURNS TABLE (
    categories varchar(9),
    res integer
) AS $$
DECLARE
    rec_res RECORD;
    reservations CURSOR FOR
        SELECT category, COUNT (room_number) AS nbofrooms
        FROM Rooms_Reserved NATURAL JOIN Rooms
        GROUP BY category;

BEGIN
    OPEN reservations;
    LOOP
        FETCH reservations INTO rec_res;
        EXIT WHEN NOT FOUND;
        categories := rec_res.category;
        res := rec_res.nbofrooms;
        RETURN NEXT;
    END LOOP;
    CLOSE reservations;
END; $$
LANGUAGE plpgsql;
```

Query για την εμφάνιση των αποτελεσμάτων

```
SELECT * FROM reserv_per_cat();
```

Data Output				Explain	Messages	Notifications
		categories character varying		res integer		
1		suite			6	
2		three_bed			6	
3		double			5	
4		single			3	

(Σημείωση: μέσα στο φάκελο της εργασίας βρίσκεται ο κώδικας της συνάρτησης μαζί με το query για την εμφάνιση των αποτελεσμάτων σε μορφή αρχείου *sql* (.sql))

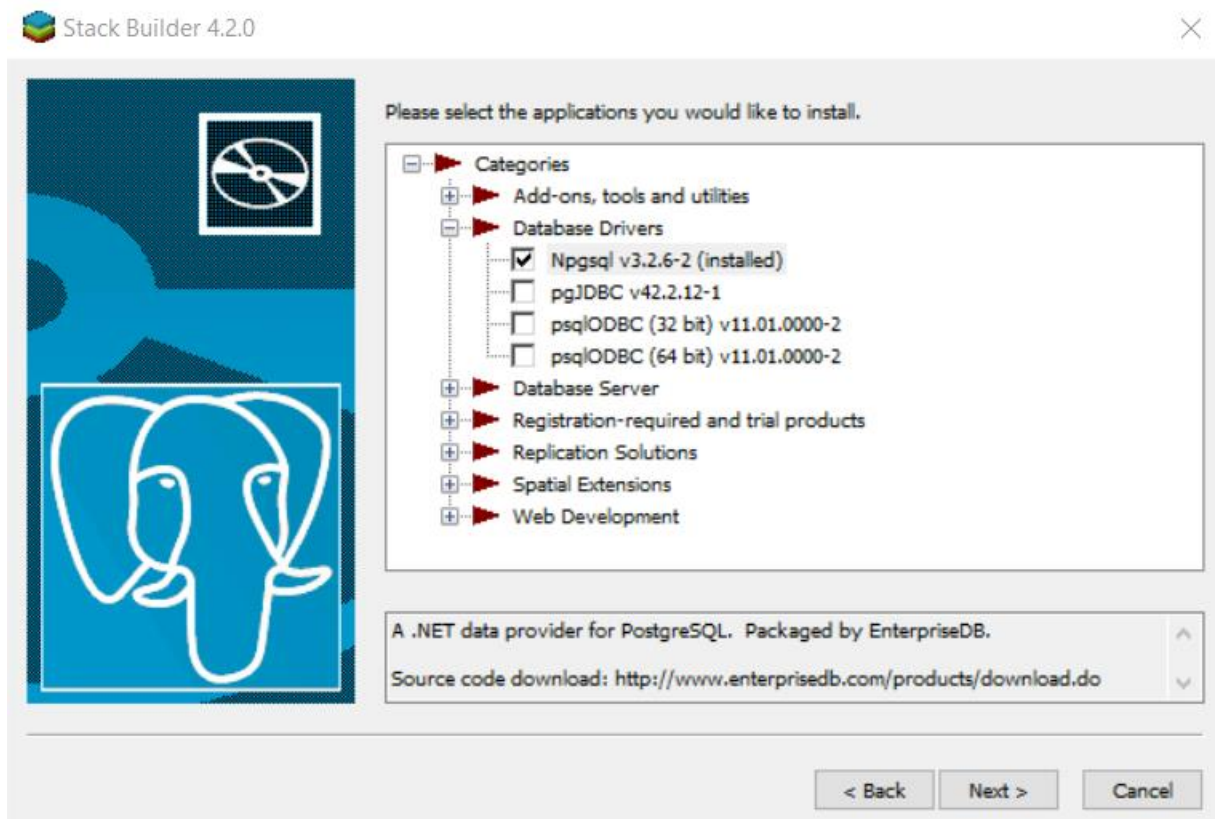
ΕΡΩΤΗΜΑ 4

Ο client υλοποιήθηκε σε C#. Ακολουθούν οδηγίες για την σύνδεση με την PostgreSQL και την ορθή εκτέλεση του προγράμματος:

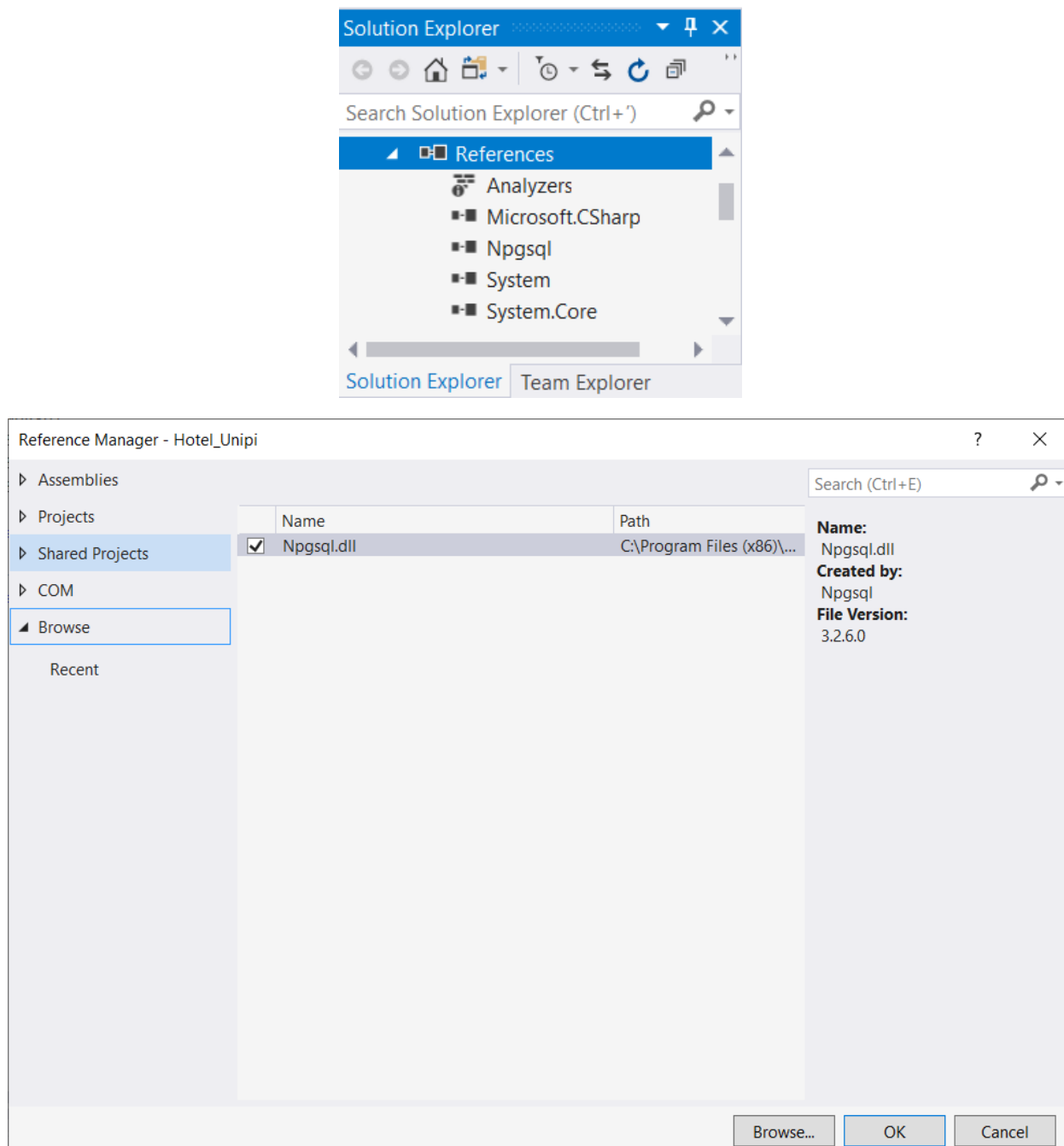
(Σημείωση: μέσα στο φάκελο της εργασίας βρίσκεται το C# project που υλοποιεί τον client)

Βήμα 0 – Δημιουργείτε ΒΔ στο **pgAdmin 4** (εντολές CREATE TABLE) και φορτώνεται τον πίνακα με δεδομένα (αρχεία *csv*)

Βήμα 1 – Από το **Application Stack Builder** εγκαθιστάτε τον **Database Driver Npgsql**



Βήμα 2 – Ανοίγετε το solution του project **Hotel_Unipi** και κάνετε *Add Reference* την βιβλιοθήκη σύνδεσης **Npgsql.dll** που εγκαταστάθηκε στο προηγούμενο βήμα



Βήμα 3 – Στον κώδικα της κλάσης **Queries** τροποποιήστε κατάλληλα την μεταβλητή **connstring** ώστε να ταιριάζει με την δική σας ΒΔ


```
Queries.cs  X
Hotel_Unipi  Hotel_Unipi.Queries  query_h0
1  using Npgsql;
2  using System;
3  using System.Collections.Generic;
4  using System.Windows.Forms;
5
6  namespace Hotel_Unipi
7  {
8      16 references
9      class Queries
10     {
11         List<string> dataItems = new List<string>();
12         string connstring = "Server=127.0.0.1; Port=5432; User Id=postgres; Password=admin; Database=HotelUnipi;";
```

Βήμα 4 – Τρέξτε τον *client*! (Παρακάτω φαίνεται ενδεικτικό screenshot από την εκτέλεση του τελευταίου query)

Hotel Unipi

(h) Ποια δωμάτια ενοικιάστηκαν από την ημερομηνία "X" έως και σήμερα: View results

115	, 9/24/2019 12:00:00 AM , 9/27/2019 12:00:00 AM
103	, 10/13/2019 12:00:00 AM , 10/27/2019 12:00:00 AM
117	, 10/29/2019 12:00:00 AM , 11/3/2019 12:00:00 AM
114	, 10/30/2019 12:00:00 AM , 11/17/2019 12:00:00 AM
108	, 11/6/2019 12:00:00 AM , 11/10/2019 12:00:00 AM
105	, 11/24/2019 12:00:00 AM , 12/9/2019 12:00:00 AM

day: 20
month: 9