Μάθημα: «Βάσεις Δεδομένων (4ο εξ.)»

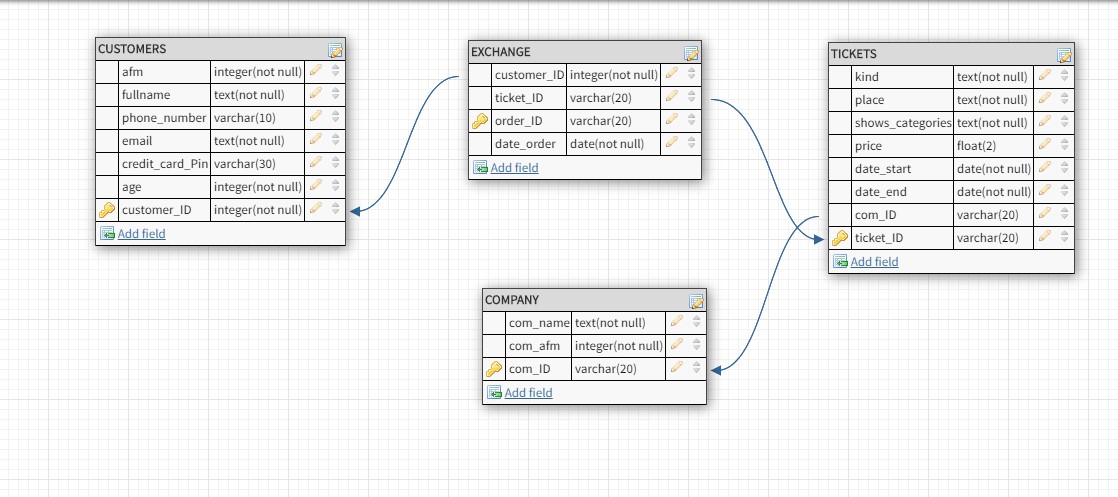
Εργασία DB1 2018-2019

ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ:

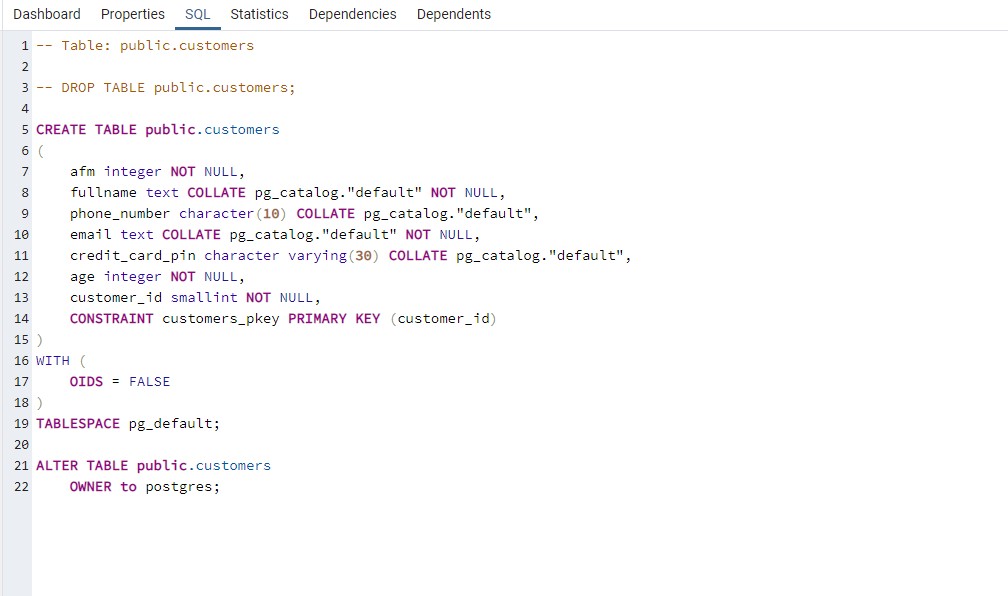
* **Φώτιος Σιούζιος Π17121**  [**fotaras46@gmail.com**](mailto:fotaras46@gmail.com)
* **Αναστάσιος Κωνσταντινίδης Π17057** [**tassoslakersosfp@hotmail.com**](mailto:tassoslakersosfp@hotmail.com)
* **Χρήστος Δήμτσας Π17027** [**xristos.dimtsas@gmail.com**](mailto:xristos.dimtsas@gmail.com)

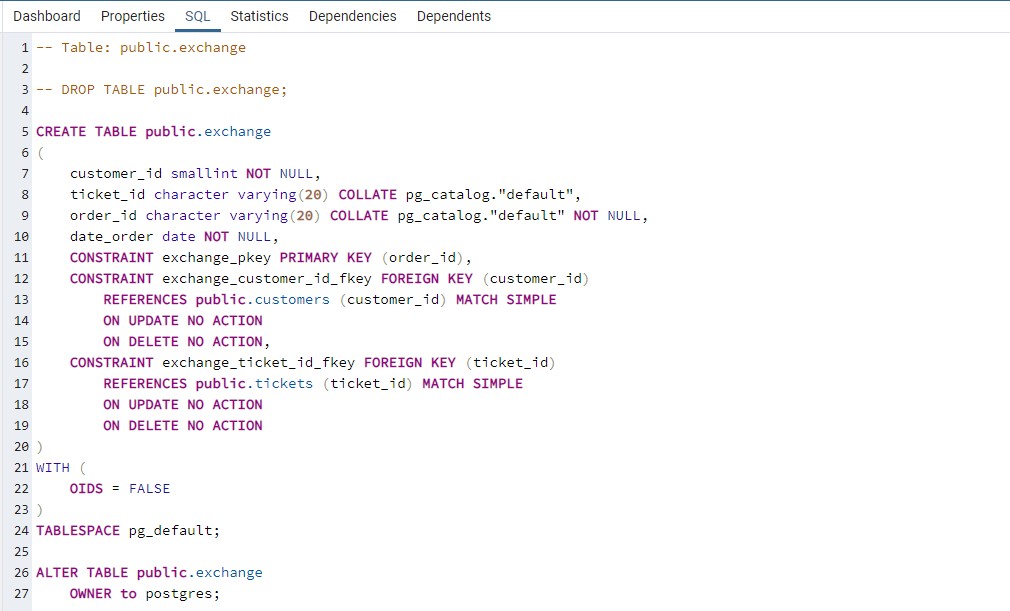
**Ερώτημα 1 .Σχεσιακή Βάση Δεδομένων**

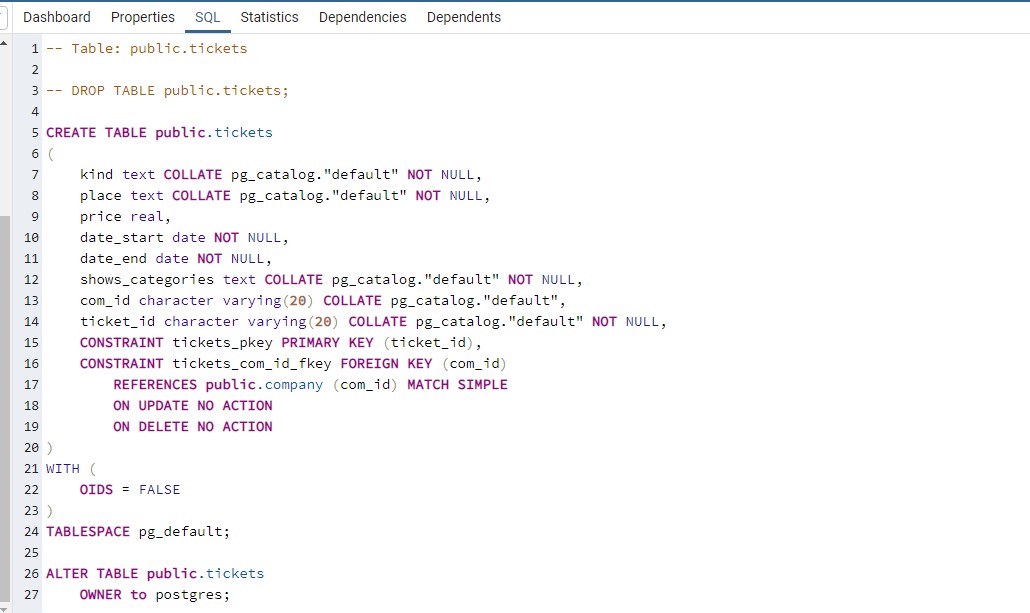
(a)











Δημιουργήσαμε τους πίνακες (company , customers ,exchange, tickets) .

* **Company :**

Primary Key : com\_id

* **Customers :**

Primary Key : customer\_id

* **Exchange :**

Primary Key : order\_id

Foreign Key : customer\_id

Foreign Key : tickets\_id

(Σε αυτό το table αντιστοιχείται ο εκάστοτε πελάτης με το εισιτήριο που έχει αγοράσει. )

* **Tickets :**

Primary Key : ticket\_id

Foreign Key : com\_id

(Στην tickets μέσω του com\_id φαίνεται ποία εταιρία έχει προμηθεύσει το κάθε εισιτήριο. )

(b)

Με βάση την θεωρία κανονικοποιησης, η βάση αρχικά είναι

1NF καθώς για κάθε κελί έχει ξεχωριστή αξία.

Οι τιμές που αποθηκεύονται σε μια στήλη είναι του ίδιου τομέα.

Όλες οι στήλες έχουν μοναδικά ονόματα και η σειρά με την οποία αποθηκεύτηκαν τα δεδομένα, δεν είχε σημασία.

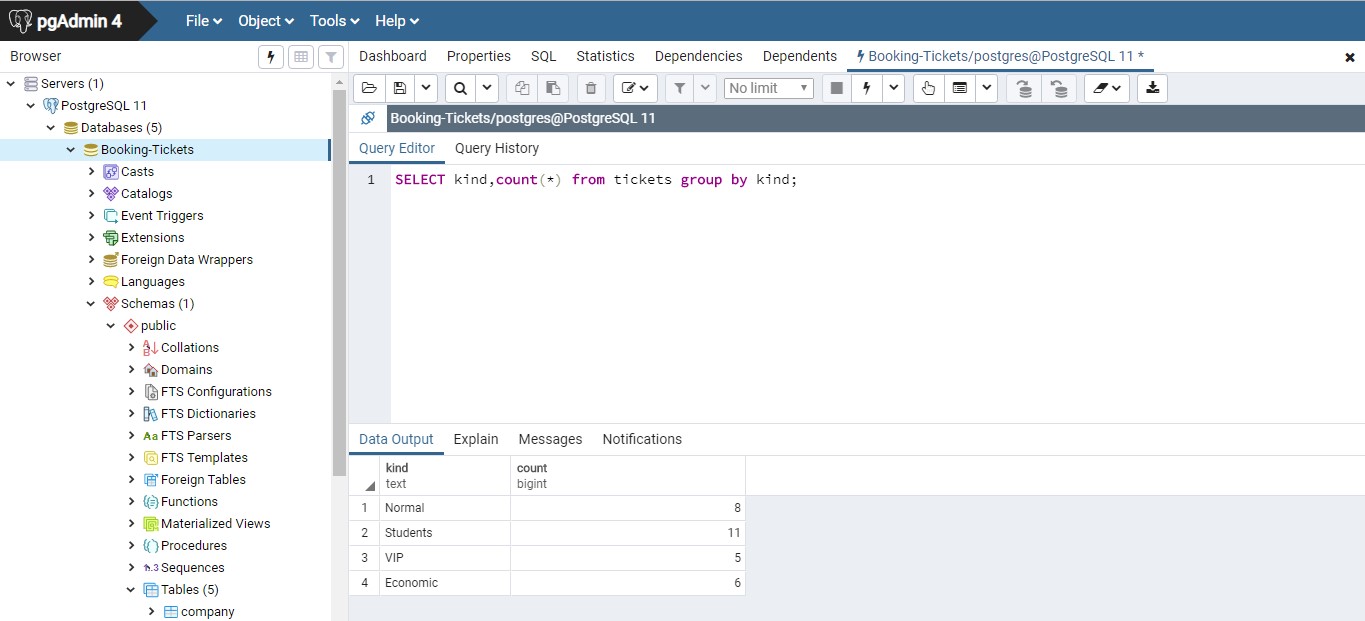
Έπειτα η βάση είναι 2NF καθώς είναι 1ΝF και δεν έχει Μερική Εξάρτηση, δηλαδή δεν υπάρχει κελί(που δεν είναι πρωτεύον κλειδί) που να μπορεί να προσδιορίζεται από άλλο κελί μη πρωτεύοντος κλειδιού.

η βάση είναι και 3NF καθώς είναι 2ΝF και δεν έχει μεταβατική εξάρτηση. Δηλαδή , δεν υπάρχει κελί(που δεν είναι πρωτεύον κλειδί) που να μπορεί να εξαρτάται από άλλο κελί μη πρωτεύοντος κλειδιού .Καθώς άμα υπήρχαν , τότε αυτά τα δυο κελία θα δημιουργούσαν νέο ξεχωριστό πίνακα.

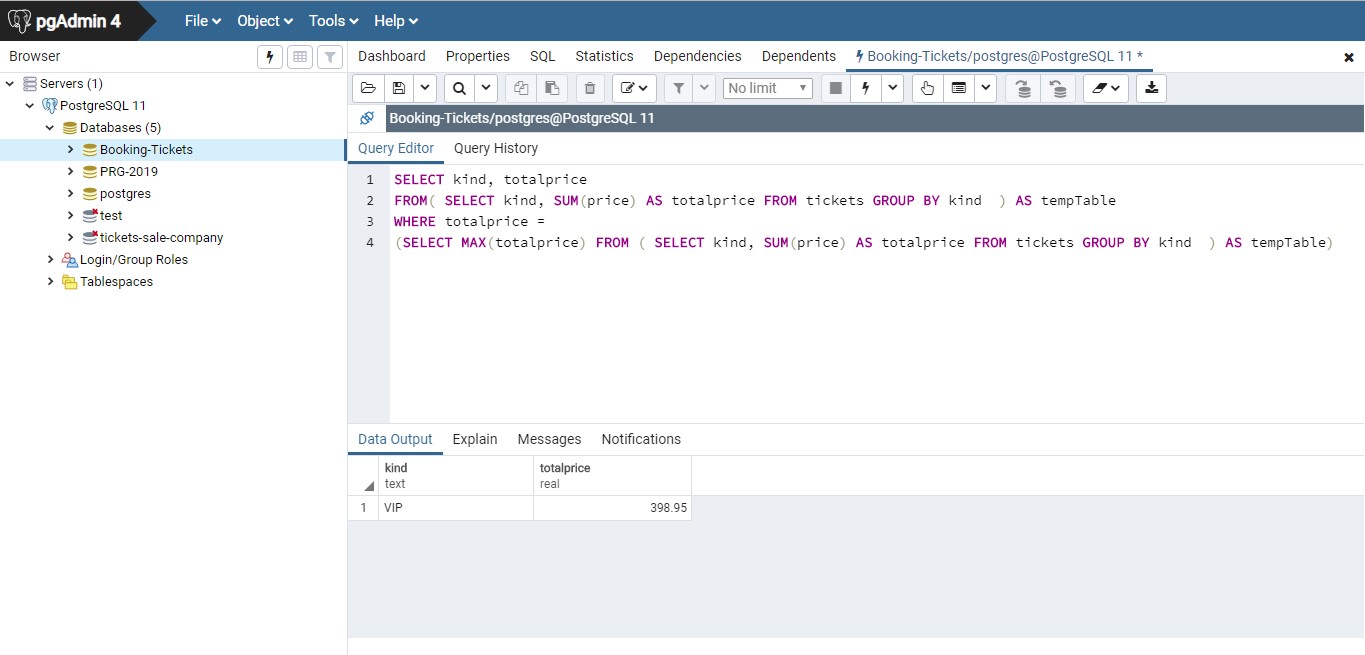
η βάση είναι και BCNF καθώς είναι 3ΝF και τα μη πρωτεύον κλειδιά δεν μπορούν μαζί να προσδιορίσουν επακριβώς τα πρωτεύον κλειδιά δηλαδή για κάθε functional dependency ( X → Y ), το X αποτελεί super Key.

**Ερώτημα 2. Εκτελέστε τις παρακάτω ερωτήσεις (queries) στη ΒΔ (εντολές SELECT).**

1. Πόσες πωλήσεις έχει η κάθε κατηγορία εισιτηρίων:

****

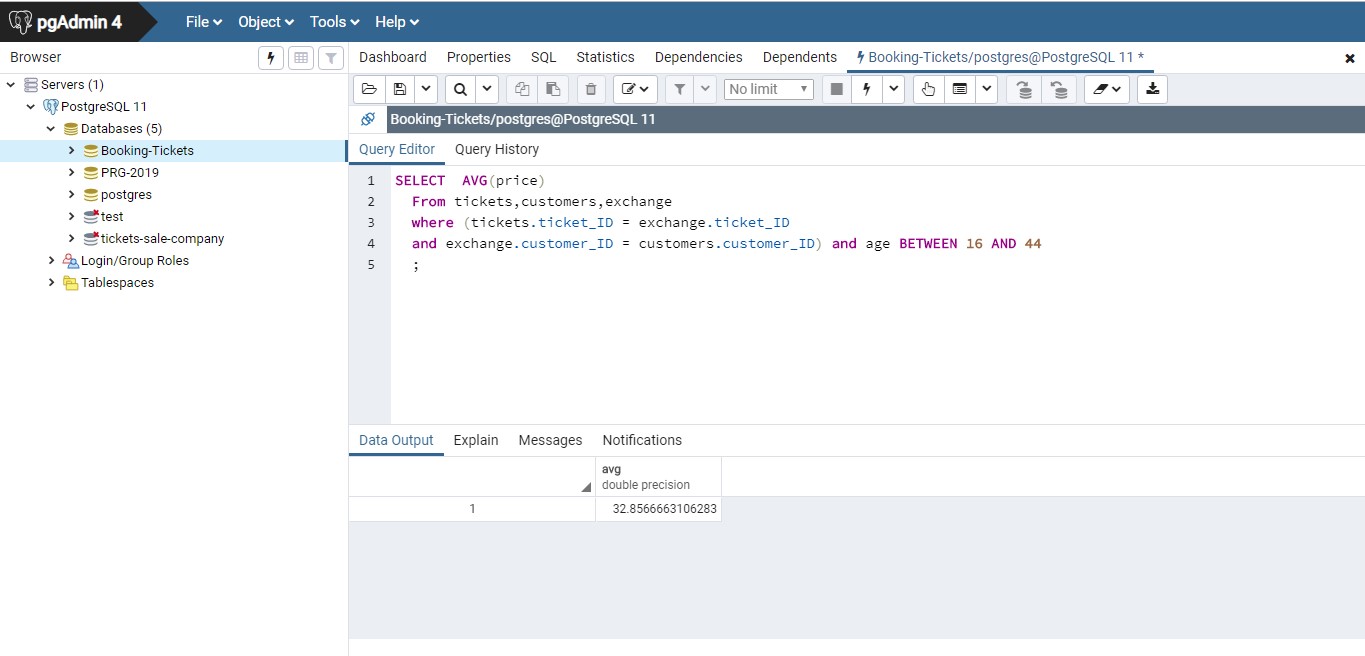
(b) Ποια κατηγορία εισιτηρίων έχει τις περισσότερες πωλήσεις (μεγαλύτερο τζίρο).

****

Αρχικά , γίνεται το άθροισμα των τιμών μέσω της συνάρτηση sum το θέτουμε την μεταβλητή totalprice , στην οποία θα πραγματοποιηθεί η διαδικασία της εύρεσης του μέγιστου αθροίσματος μέσω της συνάρτησης max.

Όλες οι διαδικασίες εκχωρούνται σε tempTable.

(c) Ποιο είναι το μέσο κέρδος ανά κατηγορία εισιτηρίων για τις ηλικίες 16-44

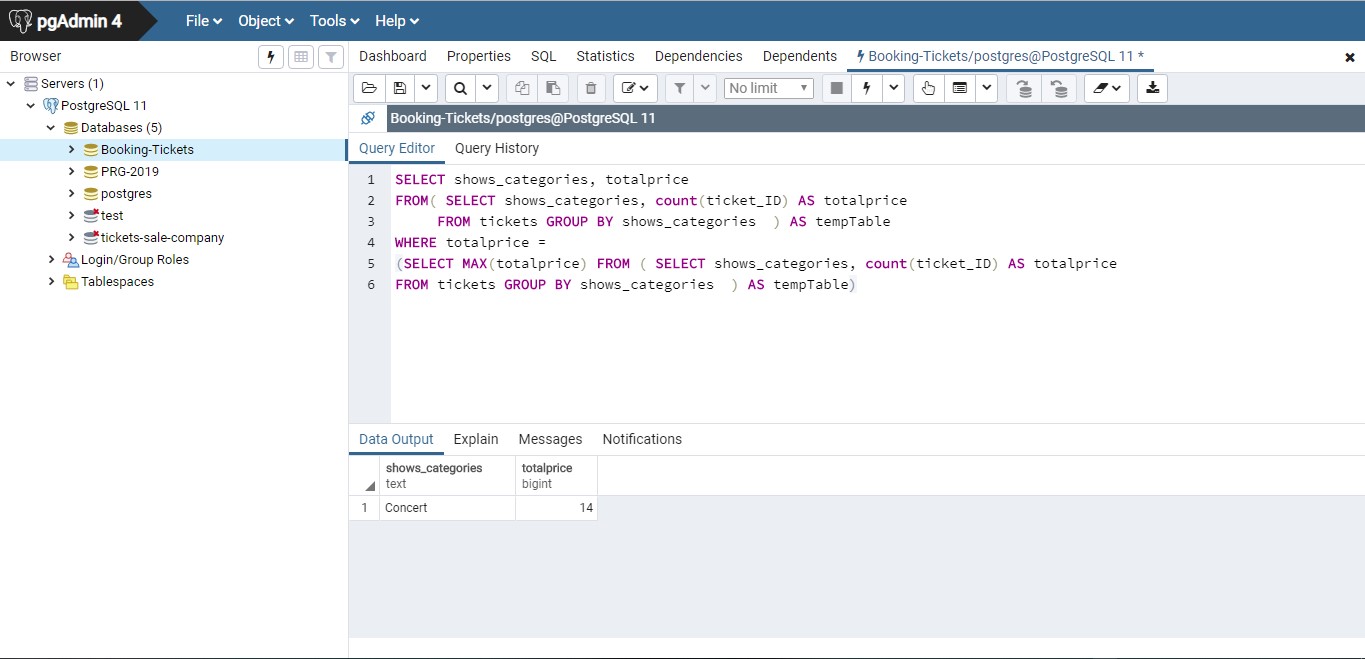


Στην συνάρτηση AVG θα βρεθεί ο μέσος όρος των τιμών για το εύρος των ηλικιών που μας ζητείται.

Στην συνέχεια, με την χρήση ξένων κλειδιών, το id του εισιτηρίου

από την tickets καταχωρείται στην exchange και το id του πελάτη από την customers. Έτσι ώστε η τιμή κάθε εισιτηρίου από τον πίνακα tickets και η ηλικία του αγοραστή από τον πίνακα customers να αντιστοιχηθούν.

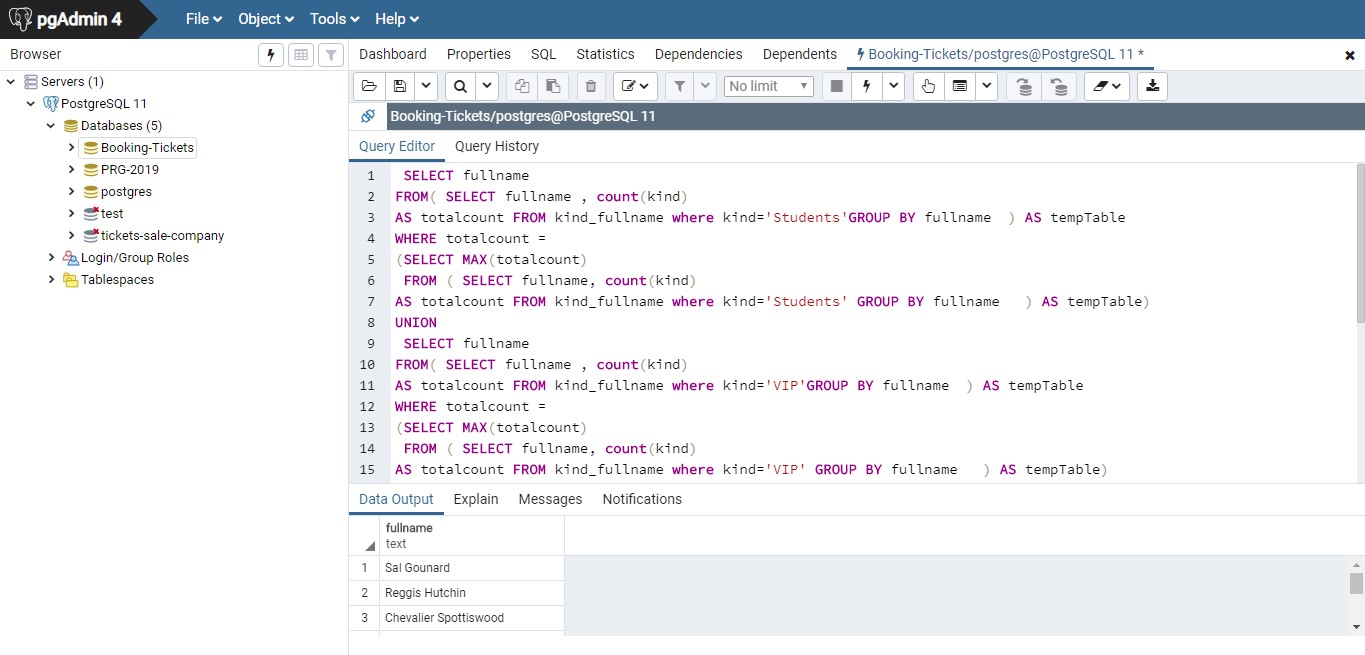
(d) Ποιο θέαμα έκανε τις περισσότερες πωλήσεις.



Στην συνάρτηση count, θα γίνει η μέτρηση των ticket\_ID

Θα καταχωρηθεί στην μεταβλητή totalprice , στην οποία θα πραγματοποιηθεί η διαδικασία της εύρεσης του μεγίστου μέσω της max.

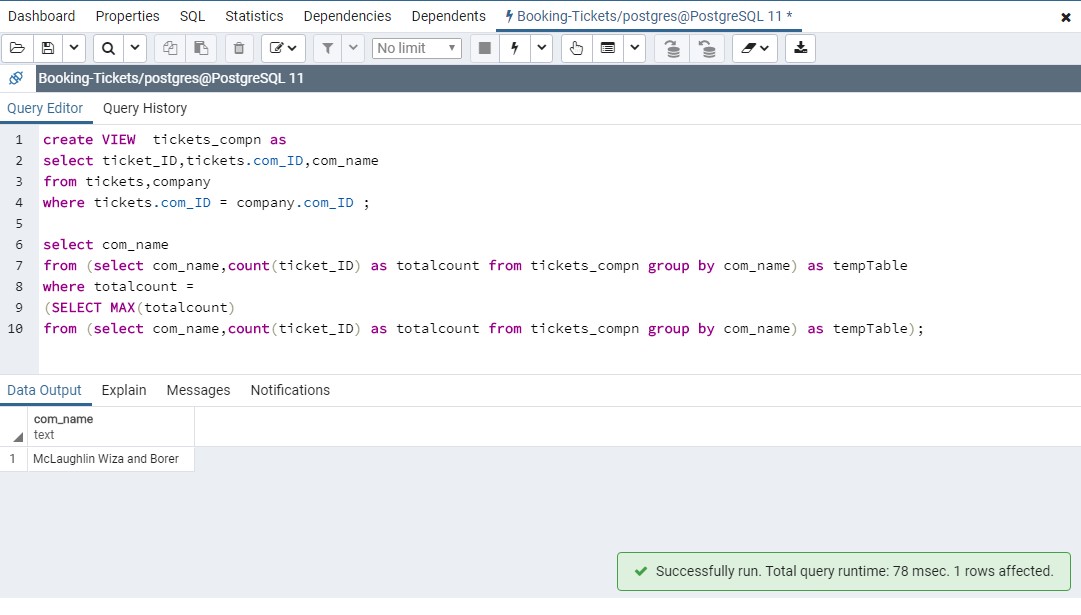
(e) Ποιος είναι ο πελάτης με τις περισσότερες αγορές ανά κατηγορία εισιτηρίου.



Για κάθε είδος εισιτηρίου(kind) βρέθηκε ο πελάτης με τις μέγιστες αγορές για το επιλεγμένο είδος η διαδικασία αυτή πραγματοποιήθηκε μέσω της καταμέτρησης των εισιτηρίων με την συνάρτηση count και βρέθηκε το μέγιστο μέσω της συνάρτησης max η οποία είναι καταχωρημένη στην μεταβλητή totalCount.

Με την εντολή UNION ενώθηκαν τα αποτελέσματα και από τις άλλες καταμετρήσεις των υπολοίπων ειδών.

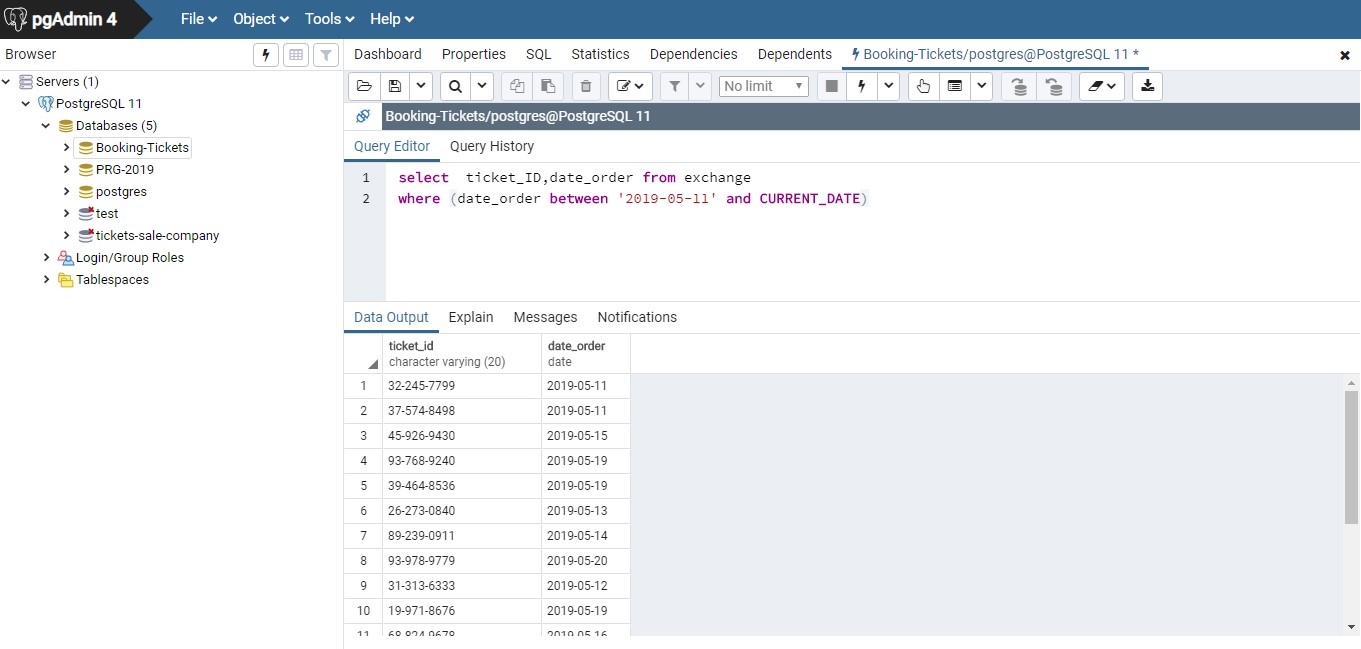
(f) Ποια εταιρεία-προμηθευτής έχει πουλήσει τα περισσότερα εισιτήρια.



Φτιάξαμε έναν καινούργιο εικονικό πίνακα (tickets\_compn) όπου αντιστοιχίζονται ο σειριακός αριθμός εισιτήριου , το όνομα του προμηθευτή-εταιρία στο καινούργιο στοιχείο tickets.com\_ID.

Mέσω της συνάρτησης count έγινε η καταμέτρηση των id των εισιτηρίων και βρέθηκε το μέγιστο μέσω της συνάρτησης max η οποία είναι καταχωρημένη στην μεταβλητή totalcount.

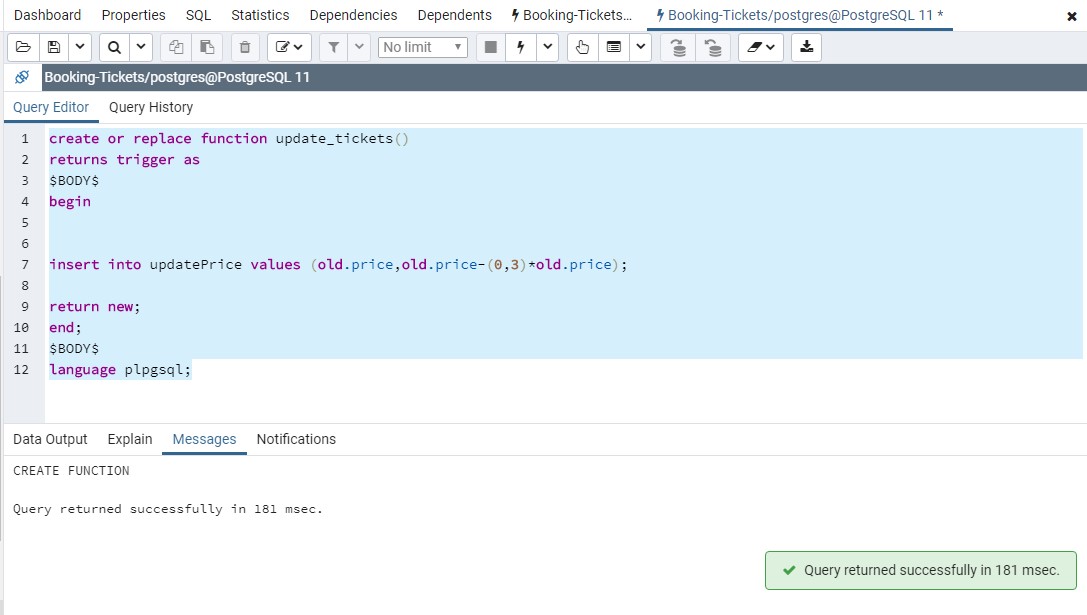
(g) Ποια εισιτήρια πουλήθηκαν από την ημερομηνία “X” έως και σήμερα.



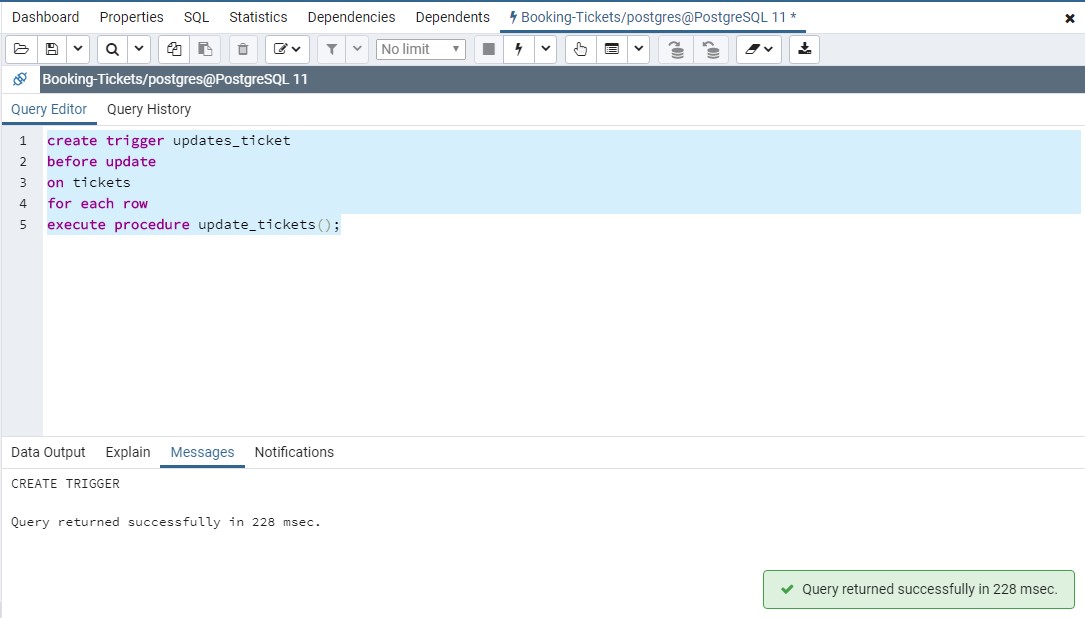
Για το αντίστοιχο χρονικό διάστημα, βρήκαμε την κατάλληλη παραγγελία.

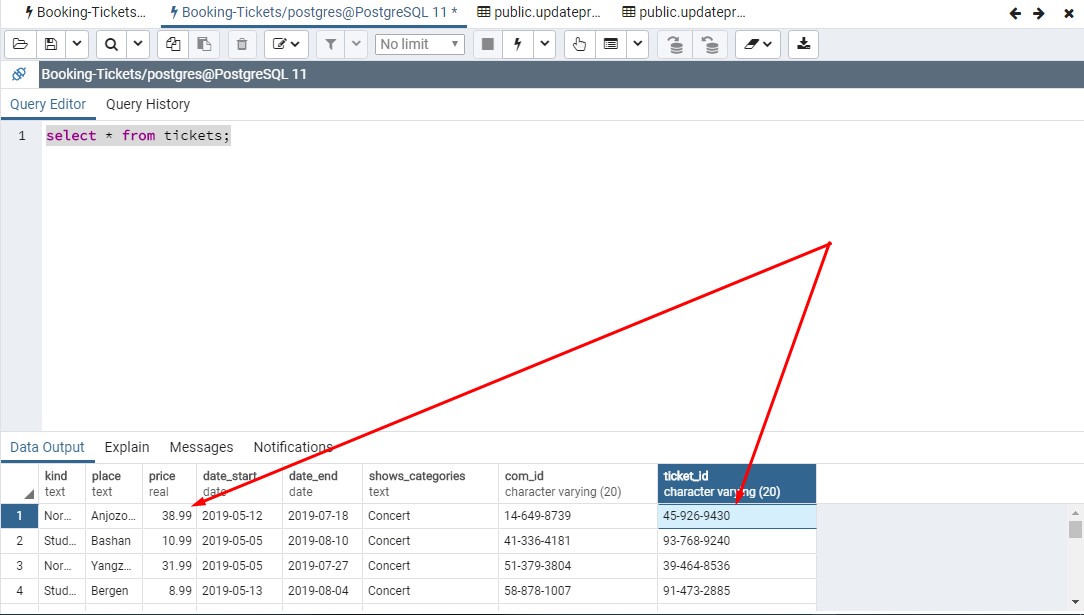
**Ερώτημα 3 . Υλοποίηση triggers και cursors**

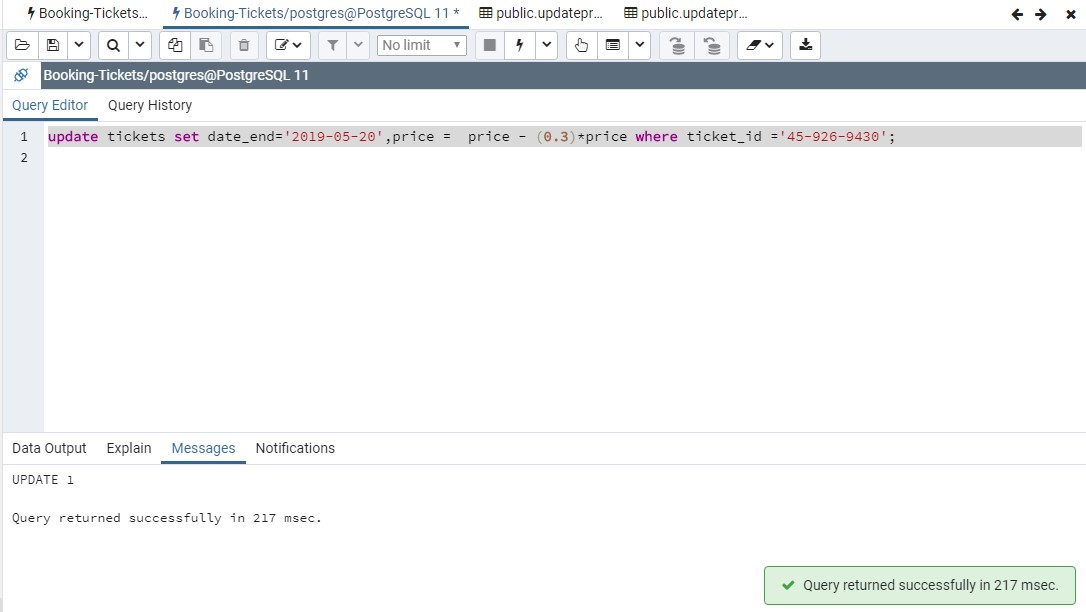
(a)

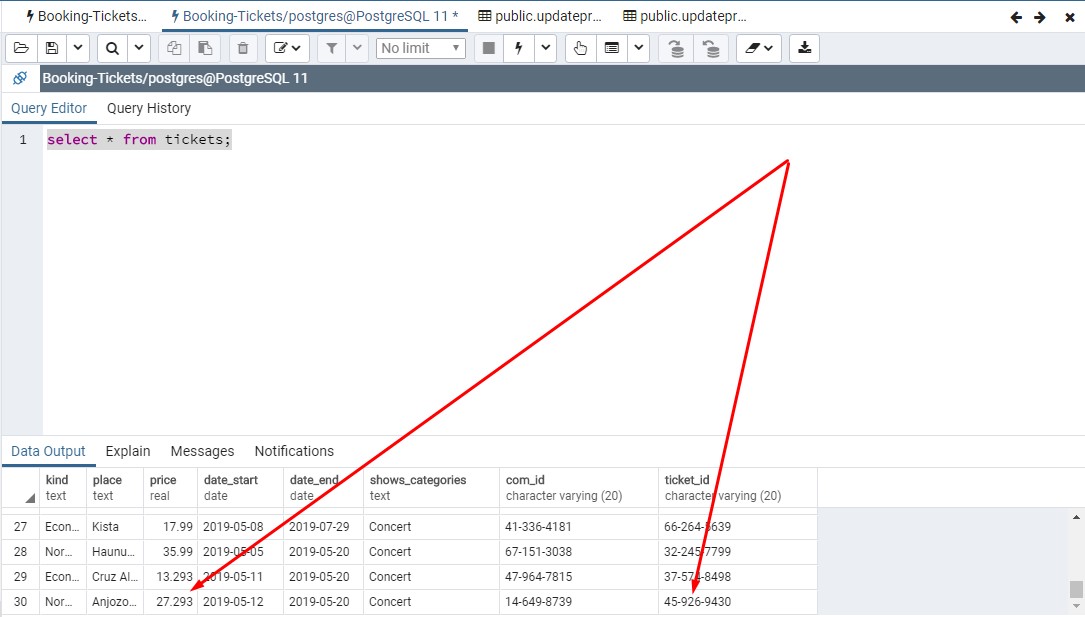


Εδώ κατασκευάζουμε το trigger και βάζουμε την απαραίτητη συνθήκη για την έκπτωση η όποια βασίζεται στην καινούργια ημερομηνία.

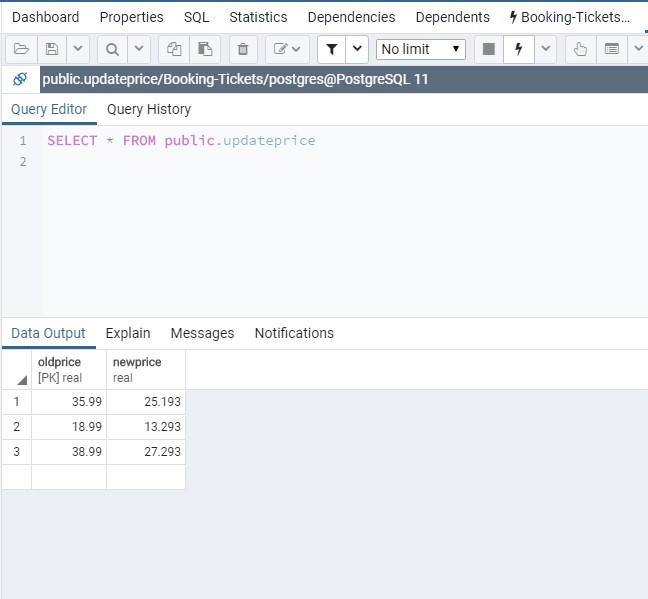






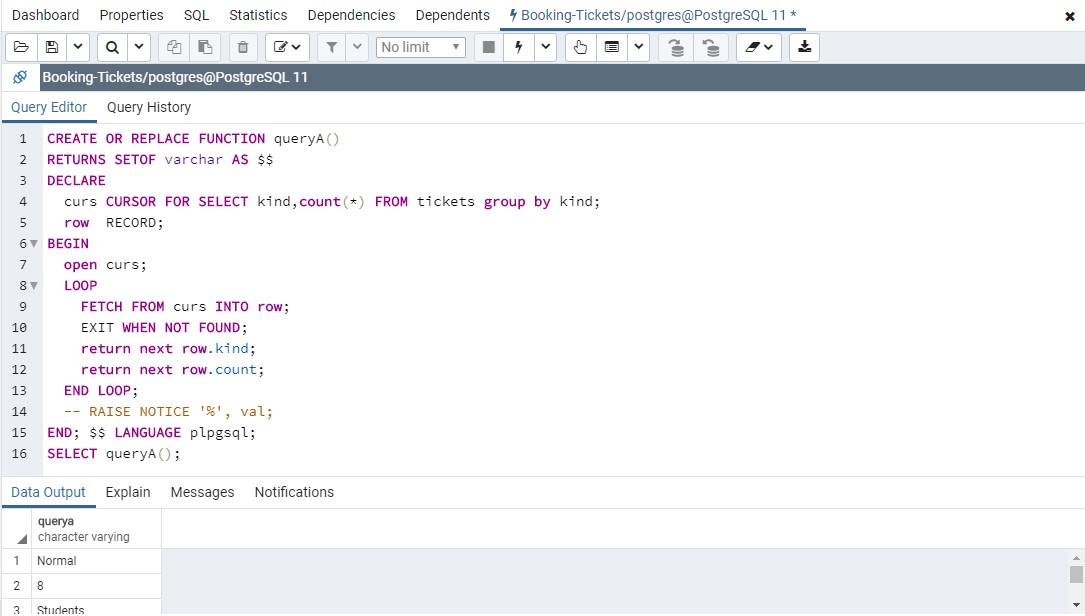


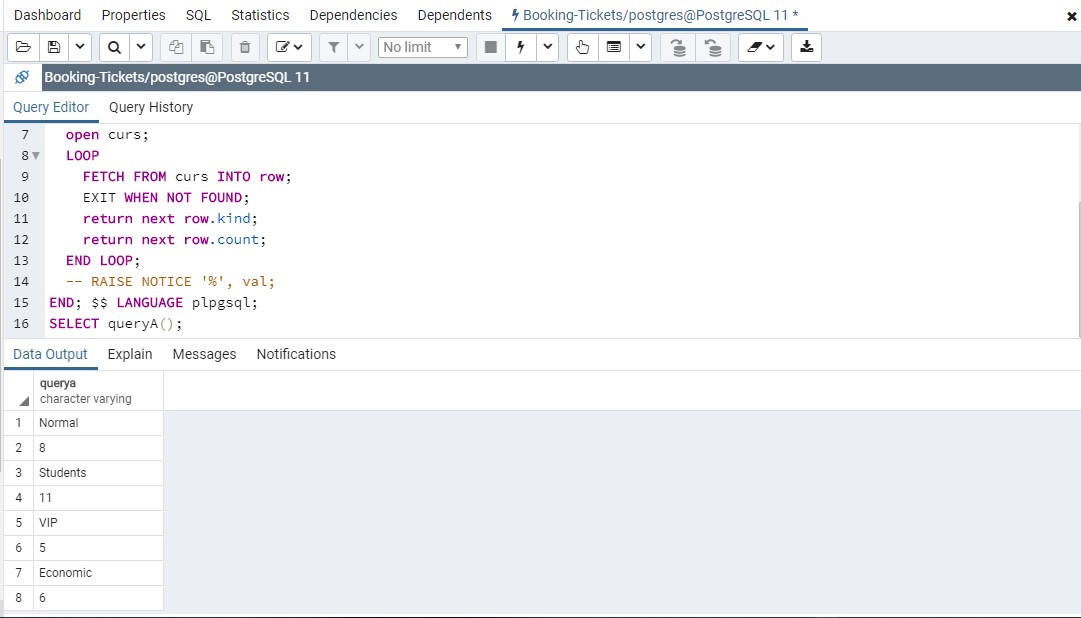
Εμφανίζεται το αλλαγμένο κελί με την κατάλληλη αλλαγμένη τιμή.



Εδώ είναι ο πίνακας που δημιουργείται από το trigger με τις παλιές και νέες τιμές.

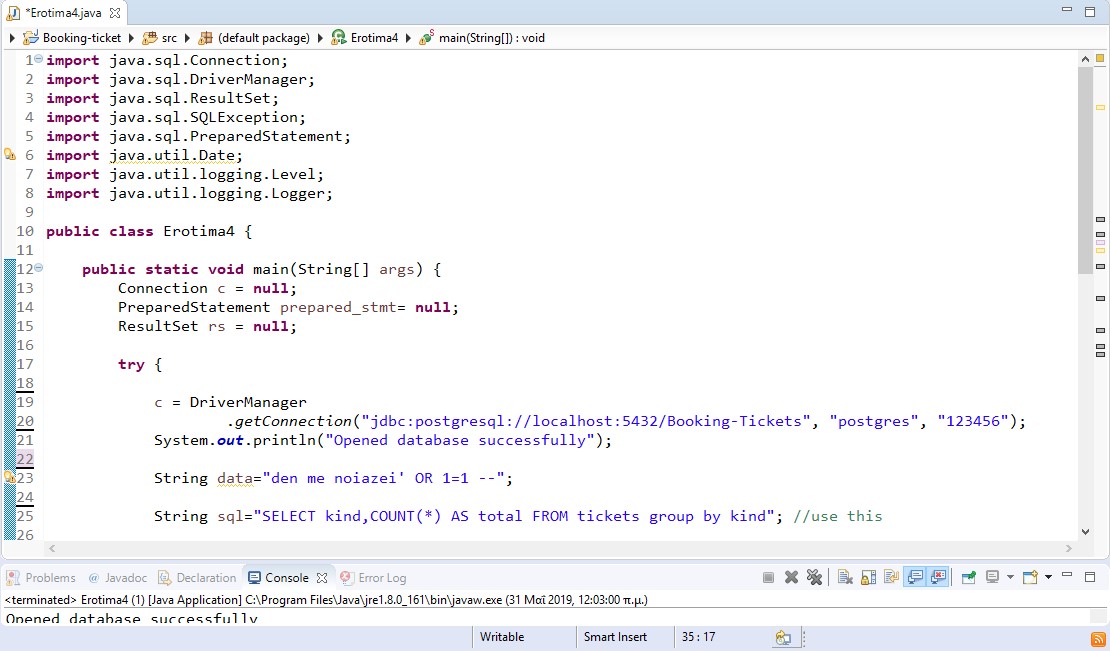
(b)

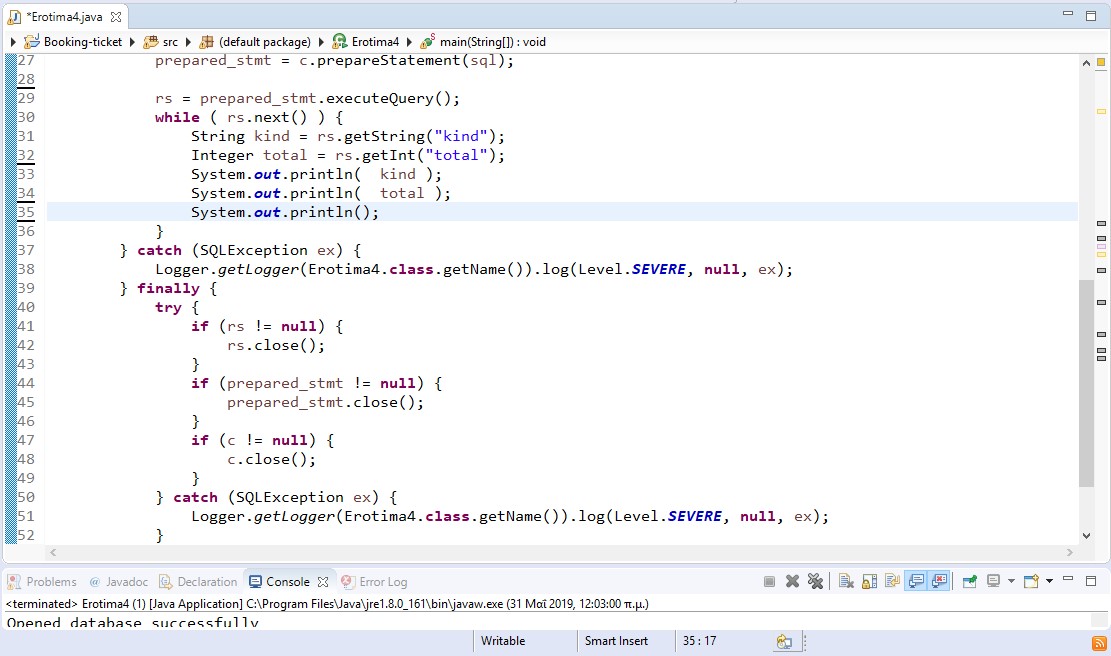


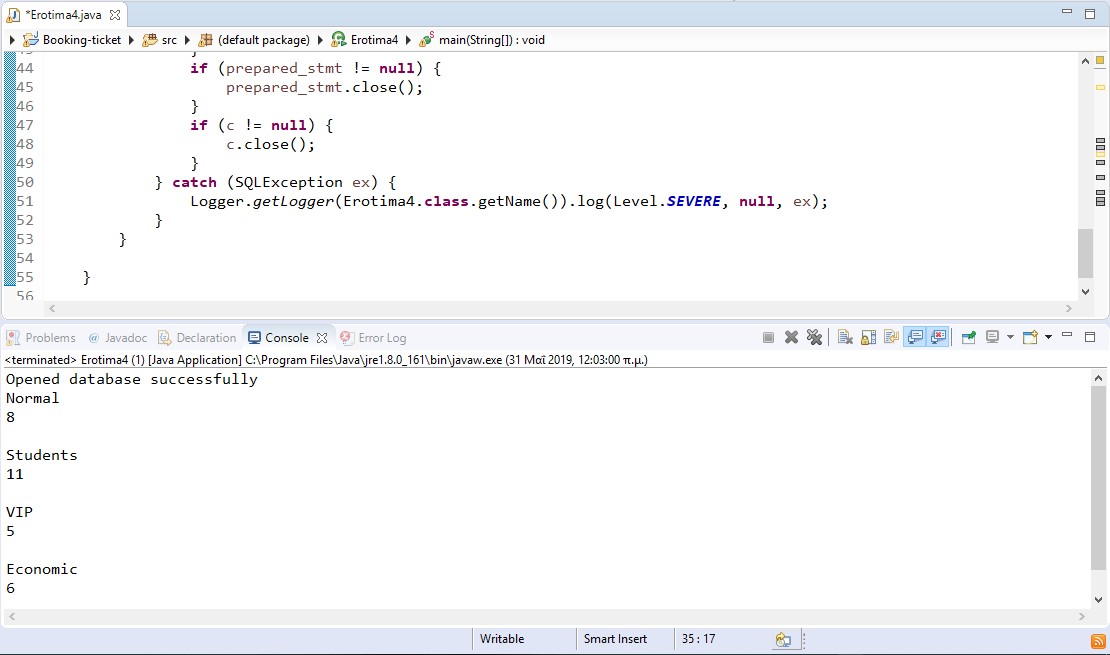


Ο κέρσορας queryA() εμφανίζει την πρώτη εντολή από τα query μέσω ενός loop και σταματά όταν βρει κενή γραμμή.

**Ερώτημα 4 .Σύνδεση ΒΔ με JDBC client**







Στο string sql καταχωρούμε την sql εντολή και μέσω prepareStatment καταφέρνουμε να την κάνουμε μεταβλητή (rs) σε java.

Τέλος , τα κατάλληλα string(kind,total) αντιστοιχούν στα στοιχειά της βάσης και εμφανίζονται τα στοιχειά που θέλουμε μέσω του prepareStatment.