ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ, ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΉΜΑ ΜΗΧΑΝΙΚΏΝ ΗΛΕΚΤΡΟΝΙΚΏΝ ΥΠΟΛΟΓΙΣΤΏΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΉΣ

Εργαστήριο Βάσεων Δεδομένων

Χειμερινό Εξάμηνο

Εργαστηριακό Project

Καθηγητές: Β. Μεγαλοοικονόμου, Ε. Βογιατζάκη, Γ. Βασιλόπουλος

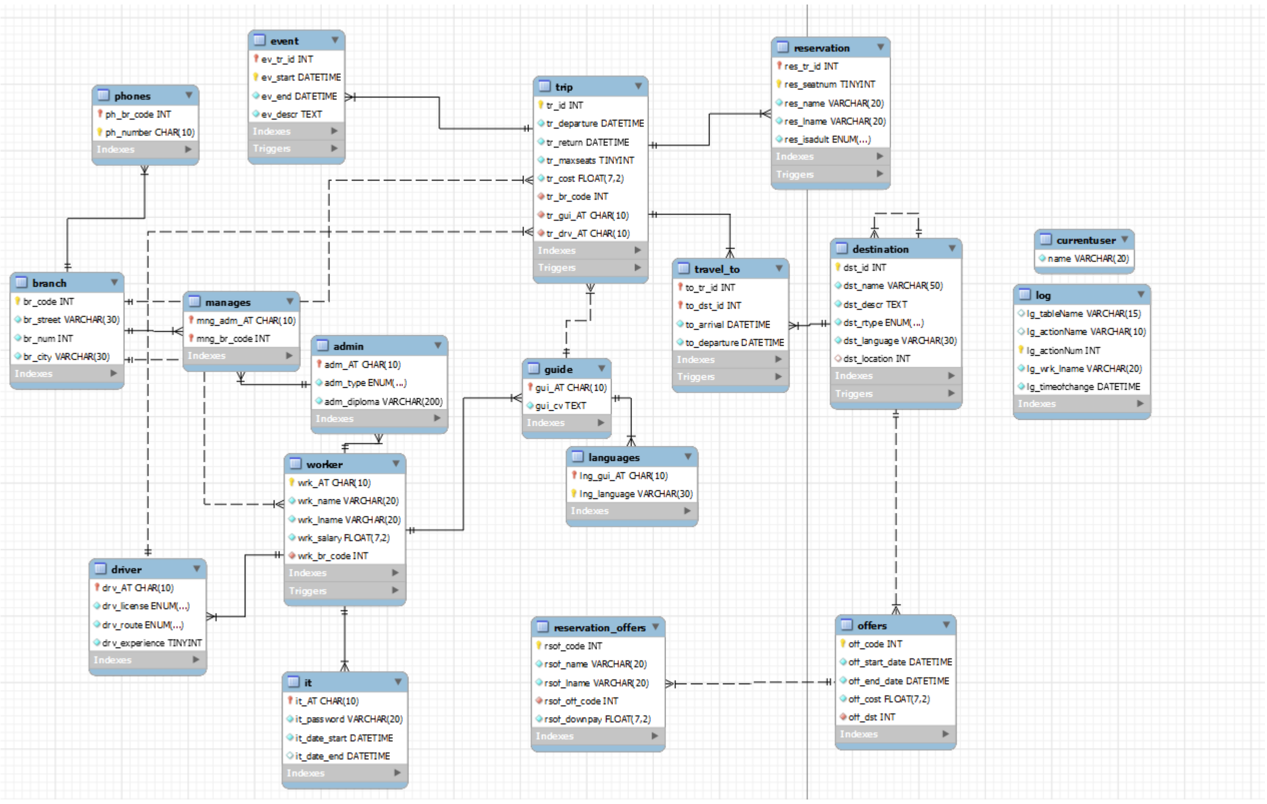
ΑΜ	**Επώνυμο**	**Όνομα**	**Έτος**
1088098	Μπαρδάκης	Βασίλειος	3^o
1084589	Χάλλας	Χαράλαμπος-Μάριος	3^o
1084567	Βιλλιώτης	Αχιλλέας	3^o

ΠΆΤΡΑ, ΙΑΝΟΥΆΡΙΟΣ 2023

Περιεχόμενα

Μέρος A: SQL Database

A.i Δημιουργία πινάκων



Το ER σχεσιακό με βάση τους ζητούμενους πίνακες. Οι πίνακες της φάσης Ά έχουν δημιουργηθεί ακριβώς έτσι όπως είναι το ER σχεσιακό διάγραμμα της εκφώνησης. Σε μερικούς πίνακες όπως τον πίνακα guide δεν βάλαμε 30 insert επειδή θα χρειαζόταν ο πίνακας worker να έχει 120 insert ενώ μπορούμε να αναδείξουμε όλες τις ιδιότητες της βάσης δεδομένων με λιγότερα. Σε τέτοια είδη πινάκων βάλαμε 10 insert.

A.ii Επιπλέον πίνακες.

Προσθέσαμε:

Τον πίνακα it, με όλες τις ιδιότητες που ζητούνται.

Τον πίνακα log που καταγράφει αλλαγές που κάνουν οι it admins σε πίνακες της βάσης, με ορίσματα:

- log_tablename VARCHAR που κρατάει το όνομα του table που έγινε η αλλαγή,
- log_actionName VARCHAR που λέει τι action έγινε (Insert, Update, Delete)
- log_actionNum είναι κλειδή INT με auto_increment ώστε να μπορούμε να ξεχωρίζουμε την κάθε αλλαγή.
- log_wrk_lname είναι VARCHAR που κρατάει το επώνυμο του it user που έκανε την αλλαγή.
- log_timeofchange είναι τύπου DATETIME και κρατάει την ώρα που έγινε η αλλαγή.

Τον πίνακα currentuser, ο οποίος αποθηκεύει το όνομα του τρέχοντος χρήστη. Χρησιμοποιείται στον πίνακα log για το πεδίο wrk_lname.

Τον πίνακα offers και reservation_offers. Για τις κρατήσεις προσφορών, υλοποιήσαμε ένα python script που συνδυάζει τυχαία τα ονόματα που περιέχονται στο names.txt και προσθέτει μια τιμή απο το 50 έως το 200 για τη προκαταβολή:

```
import os
import random


f = open(os.getcwd()+'\\names.txt', mode='r', encoding='utf-16')
names=[]

for line in f:
    line=line.split()
    names.append([line[0], line[1]])


f = open(os.getcwd()+'\\insert60.sql', mode='w')
f.write("USE travel_agency;\n\n")
f.write("INSERT INTO reservation_offers VALUES\n")

for i in range(3):
    for j in range(20000):
        downpay = round(random.uniform(50, 200), 2)
        firstName = int(round(random.uniform(0, 9999), 0))
        lastName = int(round(random.uniform(0, 9999), 0))

        if(i==2 and j==19999):
            output = "(NULL,'" + str(names[firstName][0]) + "','" +
            str(names[lastName][1]) + "'," + str(i+1) + "," + str(downpay) + ");\n"
            f.write(output)
            break

        output = "(NULL,'" + str(names[firstName][0]) + "','" + str(names[lastName][1])
        + "'," + str(i+1) + "," + str(downpay) + "),\n"
        f.write(output)
```

A.iii Stored Procedures

A.iii.a 3.1.3.1 insertNewDriver

```
DROP PROCEDURE IF EXISTS insertNewDriver;
DELIMITER //
CREATE PROCEDURE insertNewDriver (AT CHAR(10), name VARCHAR(20), lname VARCHAR(20),
salary FLOAT(7,2), license ENUM('A','B','C','D'), route ENUM('LOCAL', 'ABROAD'),
experience TINYINT(4))
BEGIN
    DECLARE br_code INT(11);

    SELECT b.br_code
    INTO br_code
    FROM branch b JOIN worker w ON b.br_code = w.wrk_br_code JOIN driver d ON
w.wrk_AT = d.drv_AT
    GROUP BY b.br_code
    ORDER BY count(*) ASC
    LIMIT 1;

    INSERT INTO worker VALUES
    (AT, name, lname, salary, br_code);

    INSERT INTO driver VALUES
    (AT, license, route, experience);
END//
DELIMITER ;
```

Δέχεται ως ορίσματα τα στοιχεία του οδηγού και προσθέτει έναν νεο worker και driver στη βάση, στο branch με τους λιγότερους οδηγούς (GROUP BY branch και ORDER BY πλήθος οδηγών σε κάθε branch).

A.iii.b 3.1.3.2 viewTripsInRange

```
DROP PROCEDURE IF EXISTS viewTripsInRange;
DELIMITER //
CREATE PROCEDURE viewTripsInRange (br_code INT, departure_low DATETIME,
departure_high DATETIME)
BEGIN
    SELECT b.br_code, t.tr_id, t.tr_cost as Cost, t.tr_maxseats as Maxseats,
COUNT(r.res_tr_id) as Reservations, (t.tr_maxseats - COUNT(r.res_tr_id)) as
'Vacant_seats', CONCAT(w1.wrk_name, ' ', w1.wrk_lname) as Guide,
CONCAT(w2.wrk_name, ' ', w2.wrk_lname) as Driver, t.tr_departure as Departure,
t.tr_return as 'Return'
    FROM branch b JOIN trip t ON b.br_code = t.tr_br_code LEFT JOIN reservation r
ON t.tr_id = r.res_tr_id
                  JOIN guide g ON t.tr_gui_AT = g.gui_AT JOIN worker w1 ON g.gui_AT
= w1.wrk_AT
                  JOIN driver d ON t.tr_drv_AT = d.drv_AT JOIN worker w2 ON
d.drv_AT = w2.wrk_AT
    WHERE b.br_code = br_code AND (t.tr_departure BETWEEN departure_low AND
departure_high)
    GROUP BY t.tr_id;
END//
DELIMITER ;
```

Εμφανίζει τα ζητούμενα στοιχεία για ταξίδια που έχουν ημερομηνία αναχώρησης μέσα στο range.

A.iii.c 3.1.3.3 deleteWorker

```
DROP PROCEDURE IF EXISTS deleteWorker;
DELIMITER //
CREATE PROCEDURE deleteWorker (name VARCHAR(20), lname VARCHAR(20))
BEGIN
    DECLARE adm_AT CHAR(10);

    SELECT wrk_AT INTO adm_AT
    FROM worker
    WHERE wrk_name = name AND wrk_lname = lname
    LIMIT 1;

    IF ((SELECT COUNT(mng_adm_AT) from manages WHERE mng_adm_AT = adm_AT) = 0) THEN
        DELETE FROM admin WHERE admin.adm_AT = adm_AT;
    ELSE
        SELECT "This admin is a branch manager. Deletion not allowed.";
    END IF;

END//
DELIMITER ;
```

Σβήσιμο εργάτη μόνο αν δεν είναι branch manager. Υλοποιείται με ένα απλό IF statement που κοιτάζει το πλήθος admin με αυτό το A.T. (κάθε worker έχει μοναδικό A.T. άρα αν είναι manager το count θα είναι 1, και αν δεν είναι 0.

A.iii.d 3.1.3.4.1 findOfferInRange

```
DROP PROCEDURE IF EXISTS findOfferInRange;
DELIMITER //
CREATE PROCEDURE findOfferInRange(IN minAmount INT, IN maxAmount INT)
BEGIN
  SELECT rsof_name 'First Name', rsof_lname 'Last Name'
  FROM reservation_offers
  WHERE rsof_downpay BETWEEN minAmount AND maxAmount;
END//
DELIMITER ;
```

Επιστρέφει όλα τα ονόματα ατόμων που έχουν κάνει κράτηση σε προσφορές ταξιδιών, στο price range που ορίζουν τα ορίσματα min/maxAmount.

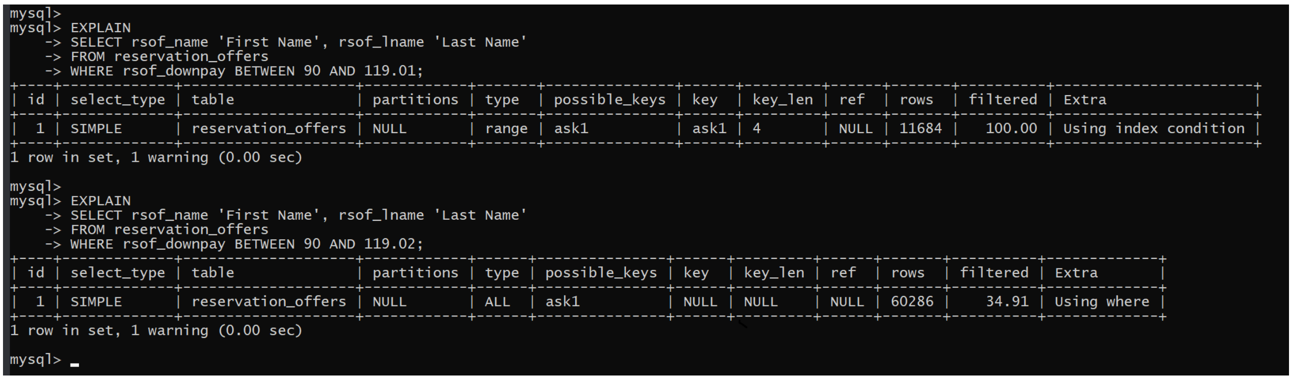
A.iii.e 3.1.3.4.2 findInfoByLastName

```
DROP PROCEDURE IF EXISTS findInfoByLastName;
DELIMITER //
CREATE PROCEDURE findInfoByLastName(IN inputLName VARCHAR(20))
BEGIN
  DECLARE tempFName VARCHAR(20);
  DECLARE tempLName VARCHAR(20);
  DECLARE tempCode INT;
  DECLARE multipleFound INT;
  DECLARE CONTINUE HANDLER FOR 1172 SET multipleFound=1;

  SET multipleFound=0;

  SELECT r.rsof_name, r.rsof_lname, o.off_code
  INTO tempFName, tempLName, tempCode
  FROM reservation_offers r
  INNER JOIN offers o ON r.rsof_off_code=o.off_code
  WHERE rsof_lname LIKE inputLName;

  -- If more than 1 found, we need to select counts instead!
  IF (multipleFound=1) THEN
    SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.'
    FROM reservation_offers r
    INNER JOIN offers o ON r.rsof_off_code=o.off_code
    WHERE r.rsof_lname LIKE inputLName
    GROUP BY o.off_code;
  ELSE
    SELECT tempFName 'First Name', tempLName 'Last Name', tempCode 'Offer Code';
  END IF;

END//
DELIMITER ;
```

Βρίσκει τα στοιχεία ατόμου που έχει κάνει κράτηση σε προσφορά ταξιδιού απο το επώνυμό του. Αν υπάρχουν περισσότερες απο μια κρατήσεις στο ίδιο επώνυμο (συγγενείς, τυχών συνωνυμίες, κλπ) στέλνεται error 1172 (Result consisted more than one row) και επιστρέφεται ο αριθμός κρατήσεων με αυτό το επώνυμο ανά προσφορά ταξιδιού.

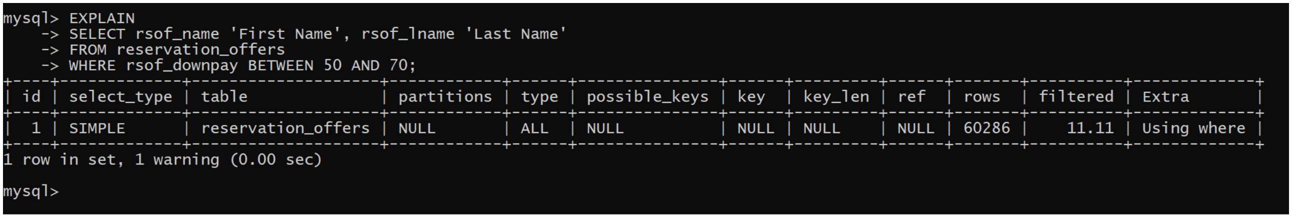
A.iii.f 3.1.3.4 Index

Ο τρόπος με τον οποίο η MySQL αποφασίζει αν θα χρησιμοποιήσει index ή όχι δεν είναι απλός. Παραθέτουμε παραδείγματα explain:



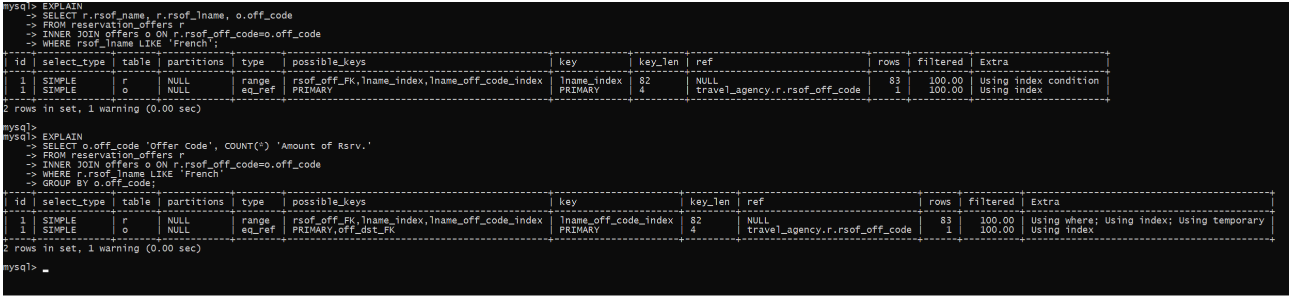
```
mysql>
mysql> EXPLAIN
    -> SELECT rsof_name 'First Name', rsof_lname 'Last Name'
    -> FROM reservation_offers
    -> WHERE rsof_downpay BETWEEN 90 AND 119.01;
+----+-------------+--------------------+------------+-------+---------------+------+---------+------+-------+----------+-----------------------+
| id | select_type | table              | partitions | type  | possible_keys | key  | key_len | ref  | rows  | filtered | Extra                 |
+----+-------------+--------------------+------------+-------+---------------+------+---------+------+-------+----------+-----------------------+
|  1 | SIMPLE      | reservation_offers | NULL       | range | ask1          | ask1 | 4       | NULL | 11684 |   100.00 | Using index condition |
+----+-------------+--------------------+------------+-------+---------------+------+---------+------+-------+----------+-----------------------+
1 row in set, 1 warning (0.00 sec)

mysql>
mysql> EXPLAIN
    -> SELECT rsof_name 'First Name', rsof_lname 'Last Name'
    -> FROM reservation_offers
    -> WHERE rsof_downpay BETWEEN 90 AND 119.02;
+----+-------------+--------------------+------------+------+---------------+------+---------+------+-------+----------+-------------+
| id | select_type | table              | partitions | type | possible_keys | key  | key_len | ref  | rows  | filtered | Extra       |
+----+-------------+--------------------+------------+------+---------------+------+---------+------+-------+----------+-------------+
|  1 | SIMPLE      | reservation_offers | NULL       | ALL  | ask1          | NULL | NULL    | NULL | 60286 |    34.91 | Using where |
+----+-------------+--------------------+------------+------+---------------+------+---------+------+-------+----------+-------------+
1 row in set, 1 warning (0.00 sec)

mysql> _
```

Για τη κλήση του findOfferInRange(90, 119.01) γίνεται χρήση του index, ενώ για το findOfferInRange(90, 119.02) δεν γίνεται.



```
mysql> EXPLAIN
    -> SELECT rsof_name 'First Name', rsof_lname 'Last Name'
    -> FROM reservation_offers
    -> WHERE rsof_downpay BETWEEN 50 AND 70;
+----+-------------+--------------------+------------+------+---------------+------+---------+------+-------+----------+-------------+
| id | select_type | table              | partitions | type | possible_keys | key  | key_len | ref  | rows  | filtered | Extra       |
+----+-------------+--------------------+------------+------+---------------+------+---------+------+-------+----------+-------------+
|  1 | SIMPLE      | reservation_offers | NULL       | ALL  | NULL          | NULL | NULL    | NULL | 60286 |    11.11 | Using where |
+----+-------------+--------------------+------------+------+---------------+------+---------+------+-------+----------+-------------+
1 row in set, 1 warning (0.00 sec)

mysql>
```

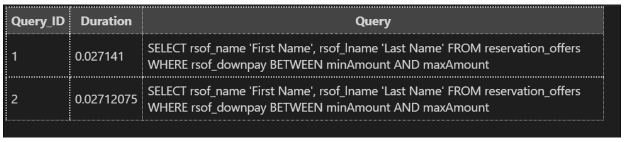
Αντίστοιχα για το findOfferInRange(50, 70) δεν γίνεται χρήση του index, ενώ το διάστημα αναζήτησης είναι μικρότερο!



```
mysql> EXPLAIN
    -> SELECT r.rsof_name, r.rsof_lname, o.off_code
    -> FROM reservation_offers r
    -> INNER JOIN offers o ON r.rsof_off_code=o.off_code
    -> WHERE rsof_lname LIKE 'French';
+----+-------------+-------+------------+--------+----------------------------------------------+-------------+---------+-------------------------------+------+----------+-----------------------+
| id | select_type | table | partitions | type   | possible_keys                                | key         | key_len | ref                           | rows | filtered | Extra                 |
+----+-------------+-------+------------+--------+----------------------------------------------+-------------+---------+-------------------------------+------+----------+-----------------------+
|  1 | SIMPLE      | r     | NULL       | range  | rsof_off_FK,lname_index,lname_off_code_index | lname_index | 82      | NULL                          |   83 |   100.00 | Using index condition |
|  1 | SIMPLE      | o     | NULL       | eq_ref | PRIMARY                                      | PRIMARY     | 4       | travel_agency.r.rsof_off_code |    1 |   100.00 | Using index           |
+----+-------------+-------+------------+--------+----------------------------------------------+-------------+---------+-------------------------------+------+----------+-----------------------+
2 rows in set, 1 warning (0.00 sec)

mysql>
mysql> EXPLAIN
    -> SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.'
    -> FROM reservation_offers r
    -> INNER JOIN offers o ON r.rsof_off_code=o.off_code
    -> WHERE r.rsof_lname LIKE 'French'
    -> GROUP BY o.off_code;
+----+-------------+-------+------------+--------+----------------------------------------------+----------------------+---------+-------------------------------+------+----------+-------------------------------------------+
| id | select_type | table | partitions | type   | possible_keys                                | key                  | key_len | ref                           | rows | filtered | Extra                                     |
+----+-------------+-------+------------+--------+----------------------------------------------+----------------------+---------+-------------------------------+------+----------+-------------------------------------------+
|  1 | SIMPLE      | r     | NULL       | range  | rsof_off_FK,lname_index,lname_off_code_index | lname_off_code_index | 82      | NULL                          |   83 |   100.00 | Using where; Using index; Using temporary |
|  1 | SIMPLE      | o     | NULL       | eq_ref | PRIMARY,off_dst_FK                           | PRIMARY              | 4       | travel_agency.r.rsof_off_code |    1 |   100.00 | Using index                               |
+----+-------------+-------+------------+--------+----------------------------------------------+----------------------+---------+-------------------------------+------+----------+-------------------------------------------+
2 rows in set, 1 warning (0.00 sec)

mysql> _
```

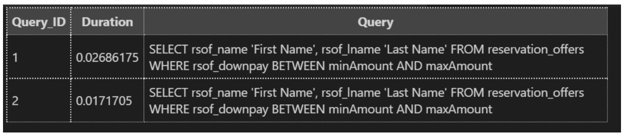
findInfoByLastName('French'): Ορθώς, κάνει χρήση του δεύτερου index, έναντι του πρώτου που δημιουργήθηκε για την προηγούμενη procedure!

Στην συνέχεια παραθέτουμε snapshot μερικών query και στην συνέχεια τα ερμηνεύουμε σε πίνακες:



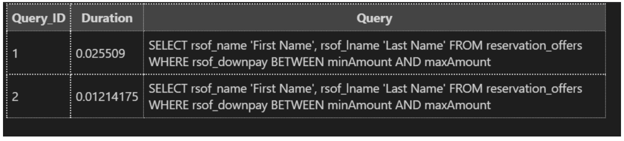
Query_ID	Duration	Query
1	0.027141	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount
2	0.02712075	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount

`findOfferInRange(90, 120)`



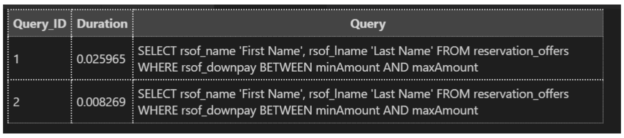
Query_ID	Duration	Query
1	0.02686175	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount
2	0.0171705	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount

`findOfferInRange(90, 110)`



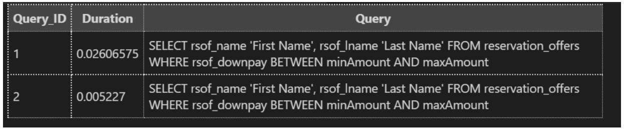
Query_ID	Duration	Query
1	0.025509	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount
2	0.01214175	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount

`findOfferInRange(90, 105)`



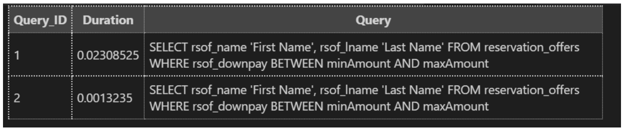
Query_ID	Duration	Query
1	0.025965	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount
2	0.008269	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount

`findOfferInRange(90, 100)`



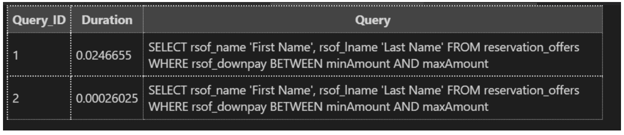
Query_ID	Duration	Query
1	0.02606575	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount
2	0.005227	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount

`findOfferInRange(90, 95)`



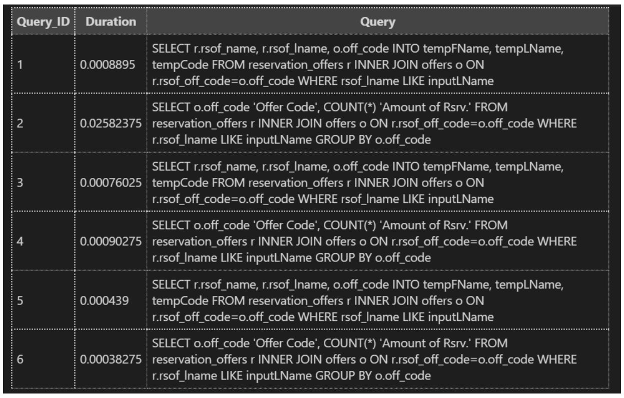
Query_ID	Duration	Query
1	0.02308525	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount
2	0.0013235	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount

`findOfferInRange(90, 91)`



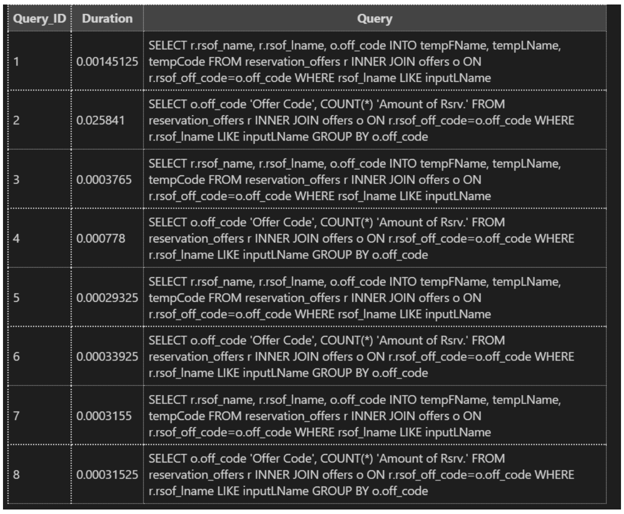
Query_ID	Duration	Query
1	0.0246655	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount
2	0.00026025	SELECT rsof_name 'First Name', rsof_lname 'Last Name' FROM reservation_offers WHERE rsof_downpay BETWEEN minAmount AND maxAmount

`findOfferInRange(90, 90,01)`



Query_ID	Duration	Query
1	0.0008895	SELECT r.rsof_name, r.rsof_lname, o.off_code INTO tempFName, templName, tempCode FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE rsof_lname LIKE inputLName
2	0.02582375	SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.' FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE r.rsof_lname LIKE inputLName GROUP BY o.off_code
3	0.00076025	SELECT r.rsof_name, r.rsof_lname, o.off_code INTO tempFName, templName, tempCode FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE rsof_lname LIKE inputLName
4	0.00090275	SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.' FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE r.rsof_lname LIKE inputLName GROUP BY o.off_code
5	0.000439	SELECT r.rsof_name, r.rsof_lname, o.off_code INTO tempFName, templName, tempCode FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE rsof_lname LIKE inputLName
6	0.00038275	SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.' FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE r.rsof_lname LIKE inputLName GROUP BY o.off_code

`indInfoByLastName` no index, lname index, both indexes.



Query_ID	Duration	Query
1	0.00145125	SELECT r.rsof_name, r.rsof_lname, o.off_code INTO tempFName, templName, tempCode FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE rsof_lname LIKE inputLName
2	0.025841	SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.' FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE r.rsof_lname LIKE inputLName GROUP BY o.off_code
3	0.0003765	SELECT r.rsof_name, r.rsof_lname, o.off_code INTO tempFName, templName, tempCode FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE rsof_lname LIKE inputLName
4	0.000778	SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.' FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE r.rsof_lname LIKE inputLName GROUP BY o.off_code
5	0.00029325	SELECT r.rsof_name, r.rsof_lname, o.off_code INTO tempFName, templName, tempCode FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE rsof_lname LIKE inputLName
6	0.00033925	SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.' FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE r.rsof_lname LIKE inputLName GROUP BY o.off_code
7	0.0003155	SELECT r.rsof_name, r.rsof_lname, o.off_code INTO tempFName, templName, tempCode FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE rsof_lname LIKE inputLName
8	0.00031525	SELECT o.off_code 'Offer Code', COUNT(*) 'Amount of Rsrv.' FROM reservation_offers r INNER JOIN offers o ON r.rsof_off_code=o.off_code WHERE r.rsof_lname LIKE inputLName GROUP BY o.off_code

`indInfoByLastName` no index, lname index, και τα δύο indexes και off_code index μόνο.

Query	Duration(ms.)	% of no index
`findOfferInRange(90, 120)`	27,141	99,93%
	27,120	
`findOfferInRange(90, 110)`	26,861	63,92%
	17,170	
`findOfferInRange(90, 105)`	25,509	47,60%
	12,141	
`findOfferInRange(90, 100)`	25,965	31,85%
	8,269	
`findOfferInRange(90, 95)`	26,065	20,05%
	5,227	
`findOfferInRange(90, 91)`	23,085	5,73%
	1,323	
`findOfferInRange(90, 90,01)`	24,665	1,06%
	0,260	

Table 1: `findOfferInRange`, πάνω χωρίς index και κάτω με index (προφανώς στο 90-120 δεν γίνεται χρήση του index).

Index Available	Total Duration(ms.)	% of no index
No Index	27,292	100%
l_name Index	1,154	4,23%
Both Indexes	0,632	2,31%
off_code Index	0,630	2,31%

Table 2: `findInfoByLastName`, αναλύουμε το δεύτερο snapshot, το total duration περιέχει το άθροισμα των 2 queries, προφανώς επιλέγεται το δεύτερο index όποτε είναι διαθέσιμο.

Παραθέτουμε και τον κώδικα για τα indexes:

```
-- Procedure 1
USE travel_agency;
SET profiling=1;


CALL findOfferInRange(90, 90.1);
-- SELECT COUNT(*) FROM reservation_offers WHERE rsof_downpay BETWEEN 90 AND 110;


SET profiling=0;
CREATE INDEX downpay_index ON reservation_offers(rsof_downpay);
SET profiling=1;


CALL findOfferInRange(90, 90.1);
-- SELECT COUNT(*) FROM reservation_offers WHERE rsof_downpay BETWEEN 90 AND 110;


show profiles;
SET profiling=0;
DROP INDEX downpay_index ON reservation_offers;


-- EXPLAIN
-- SELECT rsof_name 'First Name', rsof_lname 'Last Name'
-- FROM reservation_offers
-- WHERE rsof_downpay BETWEEN 50 AND 70;
```

```
-- Procedure 2
USE travel_agency;
SET profiling=1;

-- no index
CALL findInfoByLastName('French');

-- lname index
SET profiling=0;
CREATE INDEX lname_index ON reservation_offers(rsof_lname);
SET profiling=1;

CALL findInfoByLastName('French');

-- both index
SET profiling=0;
CREATE INDEX lname_off_code_index ON reservation_offers(rsof_lname, rsof_off_code);
SET profiling=1;
```

```sql
CALL findInfoByLastName('French');

-- only tuplet index
SET profiling=0;
DROP INDEX lname_index ON reservation_offers;
SET profiling=1;
CALL findInfoByLastName('French');

show profiles;
SET profiling=0;

DROP INDEX lname_off_code_index ON reservation_offers;
```

A.iv Triggers

A.iv.a 3.1.4.1 Log

```
USE travel_agency;

DROP TABLE IF EXISTS log;
CREATE TABLE log(
lg_tableName VARCHAR(15),
lg_actionName VARCHAR(10),
lg_actionNum INT(100) NOT NULL AUTO_INCREMENT,
lg_wrk_lname VARCHAR(20) DEFAULT 'unknown' NOT NULL,
lg_timeofchange DATETIME NOT NULL,

PRIMARY KEY (lg_actionNum)
);


DROP TABLE IF EXISTS currentUser;
CREATE TABLE currentUser(
name VARCHAR(20) DEFAULT 'null' NOT NULL
);

-- TRIGGER 1---------------------------
-- TRIP--------------------------------

DELIMITER $
DROP TRIGGER IF EXISTS logTripInsert;
CREATE TRIGGER logTripInsert
AFTER INSERT ON trip
FOR EACH ROW
    INSERT INTO log VALUES ('trip', 'Insert', NULL, (SELECT name FROM currentUser)
, CURRENT_TIMESTAMP)$
DELIMITER ;


DELIMITER $
DROP TRIGGER IF EXISTS logTripUpdate;
CREATE TRIGGER logTripUpdate
AFTER UPDATE ON trip
FOR EACH ROW
    INSERT INTO log VALUES ('trip', 'Update', NULL, (SELECT name FROM currentUser)
, CURRENT_TIMESTAMP)$
DELIMITER ;

DELIMITER $
DROP TRIGGER IF EXISTS logTripDelete;
CREATE TRIGGER logTripDelete
```

```
AFTER DELETE ON trip
FOR EACH ROW
    INSERT INTO log VALUES ('trip', 'Delete', NULL, (SELECT name FROM currentUser)
, CURRENT_TIMESTAMP)$
DELIMITER ;


-- RESERVATION--------------------------------

DELIMITER $
DROP TRIGGER IF EXISTS logReservationInsert;
CREATE TRIGGER logReservationInsert
AFTER INSERT ON reservation
FOR EACH ROW
    INSERT INTO log VALUES ('reservation', 'Insert', NULL, (SELECT name FROM
currentUser) , CURRENT_TIMESTAMP)$
DELIMITER ;

DELIMITER $
DROP TRIGGER IF EXISTS logReservationUpdate;
CREATE TRIGGER logReservationUpdate
AFTER UPDATE ON reservation
FOR EACH ROW
    INSERT INTO log VALUES ('reservation', 'Update', NULL, (SELECT name FROM
currentUser) , CURRENT_TIMESTAMP)$
DELIMITER ;

DELIMITER $
DROP TRIGGER IF EXISTS logReservationDelete;
CREATE TRIGGER logReservationDelete
AFTER DELETE ON reservation
FOR EACH ROW
    INSERT INTO log VALUES ('reservation', 'Delete', NULL, (SELECT name FROM
currentUser), CURRENT_TIMESTAMP)$
DELIMITER ;

-- EVENT--------------------------------

DELIMITER $
DROP TRIGGER IF EXISTS logEventInsert;
CREATE TRIGGER logEventInsert
AFTER INSERT ON event
FOR EACH ROW
    INSERT INTO log VALUES ('Event', 'Insert', NULL, (SELECT name FROM currentUser)
, CURRENT_TIMESTAMP)$
DELIMITER ;
DELIMITER $
DROP TRIGGER IF EXISTS logEventUpdate;
CREATE TRIGGER logEventUpdate
AFTER UPDATE ON event
FOR EACH ROW
    INSERT INTO log VALUES ('Event', 'Update', NULL, (SELECT name FROM currentUser)
, CURRENT_TIMESTAMP)$
DELIMITER ;

DELIMITER $
DROP TRIGGER IF EXISTS logEventDelete;
CREATE TRIGGER logEventDelete
AFTER DELETE ON event
FOR EACH ROW
    INSERT INTO log VALUES ('Event', 'Delete', NULL,(SELECT name FROM currentUser),
CURRENT_TIMESTAMP)$
DELIMITER ;


-- DESTINATION--------------------------------

DELIMITER $
DROP TRIGGER IF EXISTS logDestinationInsert;
CREATE TRIGGER logDestinationInsert
AFTER INSERT ON destination
FOR EACH ROW
    INSERT INTO log VALUES ('destination', 'Insert', NULL, (SELECT name FROM
currentUser) , CURRENT_TIMESTAMP)$
DELIMITER ;

DELIMITER $
DROP TRIGGER IF EXISTS logDestinationUpdate;
CREATE TRIGGER logDestinationUpdate
AFTER UPDATE ON destination
FOR EACH ROW
    INSERT INTO log VALUES ('destination', 'Update', NULL, (SELECT name FROM
currentUser), CURRENT_TIMESTAMP)$
DELIMITER ;
SELECT @name

DELIMITER $
DROP TRIGGER IF EXISTS logDestinationDelete;
CREATE TRIGGER logDestinationDelete
AFTER DELETE ON destination
FOR EACH ROW
    INSERT INTO log VALUES ('destination', 'Delete', NULL, (SELECT name FROM
currentUser) , CURRENT_TIMESTAMP)$
```

```
DELIMITER ;



-- TRAVEL_TO-------------------------------

DELIMITER $
DROP TRIGGER IF EXISTS logTravel_toInsert;
CREATE TRIGGER logTravel_toInsert
AFTER INSERT ON travel_to
FOR EACH ROW
    INSERT INTO log VALUES ('travel_to', 'Insert', NULL,(SELECT name FROM
currentUser) , CURRENT_TIMESTAMP)$
DELIMITER ;

DELIMITER $
DROP TRIGGER IF EXISTS logTravel_toUpdate;
CREATE TRIGGER logTravel_toUpdate
AFTER UPDATE ON travel_to
FOR EACH ROW
    INSERT INTO log VALUES ('travel_to', 'Update', NULL, (SELECT name FROM
currentUser), CURRENT_TIMESTAMP)$
DELIMITER ;

DELIMITER $
DROP TRIGGER IF EXISTS logTravel_toDelete;
CREATE TRIGGER logTravel_toDelete
AFTER DELETE ON travel_to
FOR EACH ROW
    INSERT INTO log VALUES ('travel_to', 'Delete', NULL, (SELECT name FROM
currentUser), CURRENT_TIMESTAMP)$
DELIMITER ;
```

Για το πρώτο trigger δεν μπορέσαμε να βρούμε τρόπο να καλούμε δυναμικά το πίνακα και το action οπότε κάναμε τρία trigger για κάθε πίνακα, για τις περιπτώσεις insert, update και delete. Αυτά τα trigger μόλις γίνει μία απο τις τρείς λειτουργίες στο table γίνεται μια insert στον πίνακα log με τα στοιχεία του it που την έκανε. Επιπλέον, για το πεδίο lg_wrk_lname του log χρησιμοποιήσαμε έναν πίνακα currentuser αντί για την κλήση της SESSION_USER() επειδή όταν αλλάζαμε το όνομα του session user έπρεπε να αλλάξουμε και τα definer απο κάθε trigger και procedure ξεχωριστά επειδή στην MYSQL8.0 δεν γίνεται να γίνει σε όλες ταυτόχρονα.

A.iv.b 3.1.4.2 noChangeAfterReserve

Όταν γίνεται προσπάθεια update στον πίνακα trip βρίσκει δεν επιτρέπει αλλαγές στις ημερομηνίες αναχώρησης, επιστροφής και το κόστος ταξιδιού αν υπάρχουν κρατήσεις στον πίνακα reservation για το συγκεκριμένο ταξίδι.

```
DROP TRIGGER IF EXISTS noChangeAfterReserve;
DELIMITER $
CREATE TRIGGER noChangeAfterReserve BEFORE UPDATE ON trip
FOR EACH ROW
BEGIN
```

```
  DECLARE reservationCount INT;
  SET reservationCount=0;
  SELECT COUNT(*) INTO reservationCount
    FROM reservation
  INNER JOIN trip ON NEW.tr_id=res_tr_id
    WHERE NEW.tr_id=res_tr_id;
  IF (reservationCount>0 AND (NEW.tr_cost!=OLD.tr_cost OR
NEW.tr_departure!=OLD.tr_departure OR NEW.tr_return!=OLD.tr_return))
  THEN SIGNAL SQLSTATE VALUE '45000'
  SET MESSAGE_TEXT = 'You cannot change the departure date, return date or the cost
after reservations are made!';
  END IF;
END$
DELIMITER ;
```

A.iv.c 3.1.4.3 noPayCuts

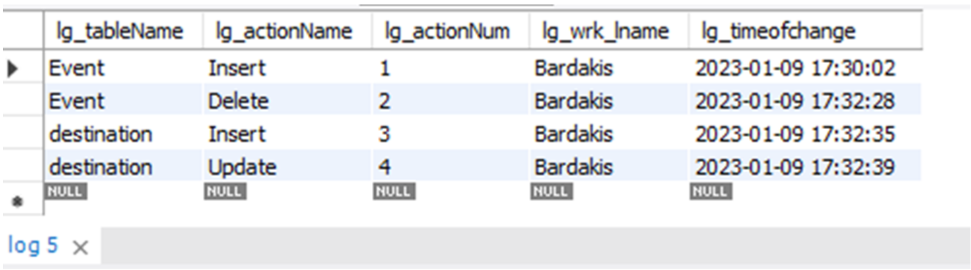
Το τρίτο trigger σε περίπτωση update του πίνακα worker κάνει μια απλή if για να δει αν ο καινούριος μισθός είναι μικρότερος και αν είναι εμφανίζει κατάλληλο μύνημα.

```
DROP TRIGGER IF EXISTS noPayCuts;
DELIMITER $
CREATE TRIGGER noPayCuts BEFORE UPDATE ON worker
FOR EACH ROW
BEGIN
  IF (NEW.wrk_salary<OLD.wrk_salary)
  THEN SIGNAL SQLSTATE VALUE '45000'
  SET MESSAGE_TEXT = 'You cannot decrease a workers salary!';
  END IF;
END$
DELIMITER ;
```

A.iv.d Λειτουργία Triggers

Log

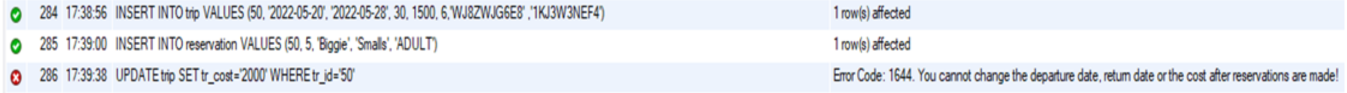
```
  INSERT INTO event VALUES
(2, '2022-05-20', '2022-05-25', 'Paintball');

DELETE FROM event WHERE ev_tr_id='2' && ev_descr='Paintball';

INSERT INTO destination VALUES
(100, 'Gallia', 'Xwra ths dutikhs Eurwphs', 'ABROAD', 'Gallika', null);

UPDATE destination SET dst_name ='Tsexia' WHERE dst_id='100';

select * from log;
```



	lg_tableName	lg_actionName	lg_actionNum	lg_wrk_lname	lg_timeofchange
▶	Event	Insert	1	Bardakis	2023-01-09 17:30:02
	Event	Delete	2	Bardakis	2023-01-09 17:32:28
	destination	Insert	3	Bardakis	2023-01-09 17:32:35
	destination	Update	4	Bardakis	2023-01-09 17:32:39
*	NULL	NULL	NULL	NULL	NULL

Modify trip

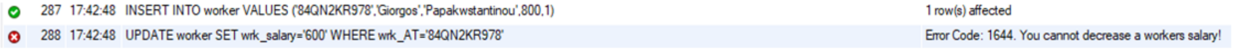
```
  INSERT INTO trip VALUES
(50, '2022-05-20', '2022-05-28', 30, 1500, 6,'WJ8ZWJG6E8' ,'1KJ3W3NEF4');
INSERT INTO reservation VALUES
(50, 5, 'Biggie', 'Smalls', 'ADULT');
UPDATE trip SET tr_cost='2000' WHERE tr_id='50';
```



```
✓ 284 17:38:56 INSERT INTO trip VALUES (50, '2022-05-20', '2022-05-28', 30, 1500, 6,'WJ8ZWJG6E8' ,'1KJ3W3NEF4')   1 row(s) affected
✓ 285 17:39:00 INSERT INTO reservation VALUES (50, 5, 'Biggie', 'Smalls', 'ADULT')                                1 row(s) affected
✗ 286 17:39:38 UPDATE trip SET tr_cost='2000' WHERE tr_id='50'                                                    Error Code: 1644. You cannot change the departure date, return date or the cost after reservations are made!
```

Update worker salary

```
  INSERT INTO worker VALUES
('84QN2KR978','Giorgos','Papakwstantinou',800,1);
UPDATE worker SET wrk_salary='600' WHERE wrk_AT='84QN2KR978';
```



```
✓ 287 17:42:48 INSERT INTO worker VALUES ('84QN2KR978','Giorgos','Papakwstantinou',800,1)   1 row(s) affected
✗ 288 17:42:48 UPDATE worker SET wrk_salary='600' WHERE wrk_AT='84QN2KR978'                 Error Code: 1644. You cannot decrease a workers salary!
```

Μέρος Β: GUI

Πιστεύουμε πως δεν χρειάζεται υπερβολική επεξήγηση του κώδικα καθώς έχει πάρα πολλά σχόλια που εξηγούν όλες τις λειτουργίες, οπότε θα εξηγήσουμε την λογική του κώδικα και όχι την κάθε εντολή. Για το setup τρέχετε πρώτα τα αρχεία .sql με την αρίθμηση του ονόματός τους και στη συνέχεια χτίζετε το java project. Συνδέσεις με τη βάση δεδομένων έγιναν χρησιμοποιώντας το JDBC mysql-connector-j-8.0.31.jar. Να ενημερώσουμε επίσης ότι μπορεί να χρειαστεί να προσθέσετε το jar στα library του project για να τρέξει το πρόγραμμα.
Κατά το testing βρήκαμε το εξής bug: Update Data -> trip -> Αλλαγή tr_cost -> Μήνυμα λάθους (You cannot change the departure date, return date or the cost after reservations are made! Error, please check your input.) -> Αλλαγή της ίδιας τιμής πάλι -> Μήνυμα "The table changed". Ο πίνακας δεν αλλάζει όντως, αφού καλείται ο trigger noPayCuts, όμως σκεφτήκαμε να το αναφέρουμε εδώ ανεξαρτήτως ως παρατήρηση.

B.i Γενική φιλοσοφία

Το πρόγραμμα αποτελείται απο δύο frame/παράθυρα: Το Loginframe κατά την εισαγωγή στοιχείων και το ItOptionsFrame κατά την είσοδο. Στο ItOptionsFrame βρίσκονται σε μορφή κουμπιών στα αριστερά όλες οι ενέργειες που μπορεί να εκτελέσει ο χρήστης. Τα κουμπιά αυτά δεν εξαφανίζονται ποτέ. Παράγωγά τους είναι τα jtable, jcombobox, pane, scrollpane, labels και buttons που θα εμφανιστούν στο δεξί μέρος του παραθύρου. Ένα (μεγάλο) κομμάτι του προγράμματος και των στοιχείων αυτών βρίσκεται στην κλάση Methods, με στατικό χαρακτήρα ώστε να καλούνται απο την ItOptionsFrame χωρίς την ανάγκη αντικειμένου Methods. Με κάθε ενέργεια υπάρχει η περίπτωση ανάγκης καθαρισμού του παραθύρου για την εμφάνιση νέων στοιχείων. Αυτό επιτυγχάνεται με την μέθοδο removeAll.

B.ii Login

Αρχικά η main καλεί την IDandPasswords: Κλάση η οποία έχει constructor που παίρνει απο το database τα AT, τα passwords και τα επώνυμα των IT που μετά χρησιμοποιούμε για το login. Αφού καλέσουμε το IDandPasswords constructor το χρησιμοποιούμε καλώντας την κλάση LoginFrame.

B.iii LoginFrame

Φτιάχνει ένα frame το οποίο έχει την εικόνα της εταιρίας, πεδία για το id και το password και δυο κουμπιά. Το κουμπί login επιχειρεί είσοδο στο σύστημα. Αν δεν υπάρχει λογαριασμός it με αυτά τα στοιχεία εμφανίζει κατάλληλο μύνημα, αλλιώς καλεί το constructor της κλάσης ItOptionsFrame. Το κουμπί exit βγαίνει απο το πρόγραμμα. Να σημειωθεί ότι σε αυτήν την κλάση χρησιμοποιήσαμε την επέκταση extends JFrame, αλλά επειδή μας προκάλεσε μερικά προβλήματα δεν το κάναμε στις υπόλοιπες κλάσεις.

B.iv ItOptionsFrame

Αποτελεί το κύριο frame του προγράμματος. Εδώ είναι όλες οι λειτουργίες που έχει ο IT user. Αρχικά στο constructor έχουμε μερικά queries, το ένα χρειάζεται για να διαγράψουμε τον currentUser και να βάλουμε το επώνυμο που αντιστοιχεί στον λογαριασμό που έκανε login, και το άλλο για να μπορούμε να χρησιμοποιήσουμε μια εμφολεβμένη select πιο μετά στον κώδικα. Στη συνέχεια δημιουργούμε μια drop-down λίστα (JComboBox) με όλα τα tables για τα κουμπιά insert,delete,update data. Φτιάχνουμε αλλη μια drop-down λίστα για το branch info κουμπί και προσθέτουμε labels, κουμπιά και το frame με τις κατάλληλες αρχικοποιήσεις.

B.iv.a removeAll

Διαγράφει όλα τα πράγματα στο frame που χρησιμοποιήθηκαν απο προηγούμενα κουμπιά. Αυτή η συνάρτηση καλείται κάθε φορά που πατιέται κάποιο κουμπί. Έχουμε κάνει override την actionPerformed για να υλοποιήσουμε λειτουργίες στα κουμπιά και στα JComboBoxes. Όλα τα κουμπιά καλούν μια μέθοδο απο την κλάση Methods στην οποία θα δούμε σε λίγο. Αξίζει να σημειωθεί ότι τα κουμπιά insert, update και delete data έχουν το ίδιο JComboBox (chooseTableBox) το οποίο φυλάει ποιό απο τα τρία κουμπιά πατήθηκαν με την βοήθεια του Static int counter και καλεί την κατάλληλη συνάρτηση απο την Methods.

B.v Methods

Υλοποιούνται όλες οι μεθόδοι που χρειάζονται για τις λειτουργίες του gui. Αρχίζουμε φτιάχνοντας static μεταβλητές, επιλέξαμε τις συγκεκριμένες μεταβλητές να είναι static και όχι τοπικές ώστε να μπορούμε να τις χρησιμοποιήσουμε στις μεθόδους αλλα και στο @override actionListener ταυτόχρονα. Φτιάχνουμε τα buttons που θα χρειαστούν σε κάθε λειτουργία του χρήστη. Όλες οι customQueryReturn, customQuery, getTableData και updateTableData είναι βοηθητικές μεθόδοι που παίρνουν κάποια String στην είσοδο τους, κάνουν connection στην mysql και επιστρέφουν ή αλλάζουν τα δεδομένα του database που ζητήθηκαν.

B.v.a insertTableData

Καλείται κάθε φορα που πατιέται το Insert Data ή αλλάζει η είσοδος του JComboBox που επιλέγει τα tables. Τυπώνει τα JFields και τα JComboBoxes για την είσοδο δεδομένων απο τον χρήστη. Όλα τα fields ορίστηκαν δυναμικά με ονόματα και data types που πήραμε απο το database με την βοήθεια της getMetaData(). Αυτό γίνεται σε κάθε κουμπί που έχει παρόμοια πεδία. Εδώ θέλουμε να σημειώσουμε ότι χρησιμοποιήσαμε JComboBoxes για τα foreign κλειδιά του κάθε πίνακα καθώς έτσι περιορίζουμε τις εισόδους του χρήστη μόνο σε έγκυρες επιλογές. Αυτό έγινε στις μεθόδους InsertTableData ΚΑΙ addItButton. Εντοπίσαμε τα foreign κλειδιά με την βοήθεια της getFKeyData που θα εξηγήσουμε παρακάτω.

B.v.b getFKeyData και δυναμικά drop-down boxes

Για την διευκόλυνση του χρήστη, πεδία κειμένου όπου μπορεί να προ-υπάρχουν εγγραφές αντικαθίστανται απο combo boxes/drop-down boxes. Με βάση τη λογική, σκεφτήκαμε ότι combo boxes χρειάζονται σε περιπτώσεις όπου το πεδίο είναι foreign key. Μέσω ενός thread απ'το stackoverflow (βρίσκεται σε σχόλιο μέσα στον κώδικα, μαζί με κάθε άλλο documentation ή forum thread που χρησιμοποιήθηκε για ανάγνωση), βρήκαμε την getMetaData, η οποία επιστρέφει σε data type DatabaseMetaData όσες πληροφορίες χρειαζόμαστε. Αν δηλαδή κατά το τρέξιμο της insertTableData ή της addIT ανιχνευθεί foreign key, το jfield του αντικαθίσταται απο jcombobox, το οποίο γεμίζεται δυναμικά απο εγγραφές του αντίστοιχου πίνακα και πεδίου.

Στις επόμενες 2 μεθόδους χρησιμοποιούμε ένα έξτρα ResultSet το οποίο μας βοήθησε να ορίσουμε τον πίνακα αποτελεσμάτων δυναμικά, αυτό έγινε με την βοήθεια της COUNT(*).

B.v.c UpdateTableButton

Για το 2ο και 3ο κουμπι (Update Data, Delete Data). Εμφανίζει τα δεδομένα του table που έχει επιλεχθεί και μετά με actionListener αμα είναι του UpdateData αλλάζει εκείνη την στιγμη μετά απο μερικούς ελέγχους (όπως αν άλλαξαν όντως τα δεδομένα ή αν είχαν null πρίν) στέλνει query στο database για να αλλάξει, το actionListener DeleteData απλά περνάει ποιό row έχει επιλεχθεί για να χρησιμοποιηθεί στο actionListener του κουμπιού DELETE ROW.

B.v.d logTableButton

Εμφανίζει τα περιεχόμενα του πίνακα log.

B.v.e addITButton

Εμφανίζει πεδία για την δημιουργία ενός νέου worker, ο οποίος γίνεται αμέσως και IT admin (insert to worker, και μετά insert worker into it).

B.v.f branchInfoButton

Δείχνει για κάθε υποκατάστημα τα στοιχεία του, το ονοματεπώνυμο του διευθυντή και το σύνολο κρατήσεων και εσόδων. Να σημειωθεί πώς στην branchInfo (και σε άλλα σημεία του κώδικα), χρειάστηκε να χρησιμοποιήσουμε την getColumnLabel αντί της getColumnName, γιατί η τελευταία επέστρεφε το αρχικό όνομα στήλης και όχι το alias, κάτι που θέλαμε για να είναι πιο ευανάγνωστοι οι πίνακες. Να σημειωθεί πως δεν χρησιμοποιήθηκαν aliases στα Insert και Update Data γιατί εκεί ο IT admin πρέπει να ξέρει ακριβώς τι είναι αυτό που εισάγει/αλλάζει, ενώ το branch info παρουσιάζει έτοιμα δεδομένα.

B.v.g branchWorkers

Επιτρέπει στον χρήστη να βλέπει για κάθε υποκατάστημα το όνομα, επώνυμο, μισθό των υπαλλήλων και το συνολικό ποσό που πληρώνει το κατάστημα για τους μισθούς των εργαζομένων.

B.v.h tripInfoButton

Σε αυτό έχουμε υλοποιήσει δικό μας ημερολόγιο με 6 διαφορετικά JComboBox. Με την βοήθεια του actionListener, όταν ο χρήστης αλλάζει το μήνα ή το έτος, ενημερώνεται αυτόματα το JComboBox των ημερών ώστε να είναι αντίστοιχο της πραγματικότητας χρησιμοποιώντας modulo (αν είναι leap year, αν ο μήνας έχει 31 ή 30 ημέρες...). Να σημειωθεί πως κατά την αλλαγή έτους δεν ενημερώνεται σωστά το πεδίο ημερών και ο χρήστης πρέπει να αλλάξει τον μήνα για να γίνει η ενημέρωση. Για κάποιο λόγο όταν γράφαμε το actionListener των πεδίων χρόνου, δεν ενεργοποιείτο κατά το πάτημα ο κώδικας (το επιβεβαιώσαμε με debugger).

B.v.i clientOffersButton

Καλείται απο το See client's offer. Χρησιμοποιούμε το DINSTINCT στο query που δίνουμε ώστε να μην έχουμε επανάληψη ονομάτων και στη συνέχεια φτιάχνουμε ένα JComboBox το οποίο τα περιέχει. Μετά καλείται το κατάλληλο procedure για το οποίο παίρνουμε στοιχεία για να φτιάξουμε τον πίνακα. Να σημειωθεί ότι χρησιμοποιήσαμε Scrollable result set ώστε να βρούμε τον αριθμό rows τον αποτελεσμάτων χωρίς εμφολευμένη Select και χωρίς να χρειαστεί να ξανατρέξουμε το query.

B.v.j contains

Είναι απλά μια μέθοδος που βρίσκει αν υπάρχει ενα συγκεκριμένο κλειδί σε ένα array. Την χρησιμοποιούμε στο submit button του Insert Data.

B.v.k Action Listeners

Το override μας εξασφαλίζει την υλοποίηση ενεργειών σε περίπτωση πατήματος κουμπιών ή πινάκων. Τα submitButton, deleteButton, submitITButton και submitDateButton παίρνουν τα δεδομένα που έχουν ήδη δωθεί απο τον χρήστη, φτιάχνουν με κατάλληλο τρόπο για κάθε περίπτωση το query και το στέλνουν στο mysql database με την βοήθεια των πρώτων μεθόδων που αναφέραμε, επιστρέφοντας ή αλλάζοντας δεδομένα απο το database. Το κουμπί submitDateButton συγκεκριμένα φτιάχνει και έναν πίνακα JTable με τα αποτελέσματα, τον οποίο τυπώνει στην οθόνη. Το actionListener της chooselName απλά καλεί την clientOffersButton αφού έχει θέσει την νέα τιμή του ονόματος στην static μεταβλητή selectedlName.

B.vi RedirectAppOutputStream

Κάνει Re-direct οποιοδήποτε System.out stream στο JTextArea ώστε να εμφανίζονται τα μηνύματα που θέλουμε στην οθόνη του προγράμματος χωρίς να φτιάχνουμε κάθε φορά νεο label. Το υλοποιήσαμε με αυτόν τον τρόπο επειδή είχαμε ήδη φτιάξει όλες τις μεθόδους με System.out.println και δεν θέλαμε να γίνει πιο περίπλοκος ο κώδικας.

Μέρος C: Σενάριο χρήσης

1 Καταχώρηση λανθασμένων στοιχείων. Εμφάνιση λάθους;

2 Εισαγωγή σωστών στοιχείων. Είσοδος στο main frame;

3 Πάτημα τυχαίων κουμπιών απο τα αριστερά (αναποφάσιστος χρήστης). Σωστή ανανέωση παραθύρων και στοιχείων;

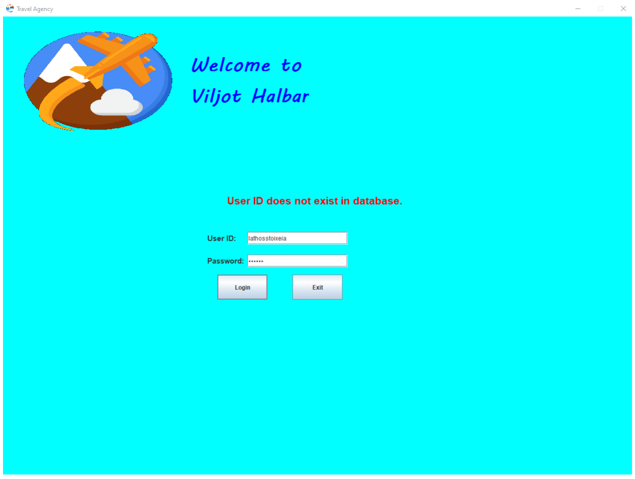
4 Update Data -> Μείωση μισθού εργάτη. Εμφάνιση λάθους. (Trigger);

5 Update Data -> Ενημέρωση ενός destination. Σωστή ανανέωση δεδομένων;

6 Insert Data -> Worker, αφήνει κενό το AT. Error;

7 Add IT, αφήνει κενά τα it_date_start και it_date_end και πηγαίνει Update Row -> IT να δει την εγγραφή που έκανε. it_date_start = στιγμή που έγινε το query, it_date_end = null;

8 Branch info -> Κοιτάζει τους υπαλλήλους ανά branch. Σωστή ανταπόκριση του table;

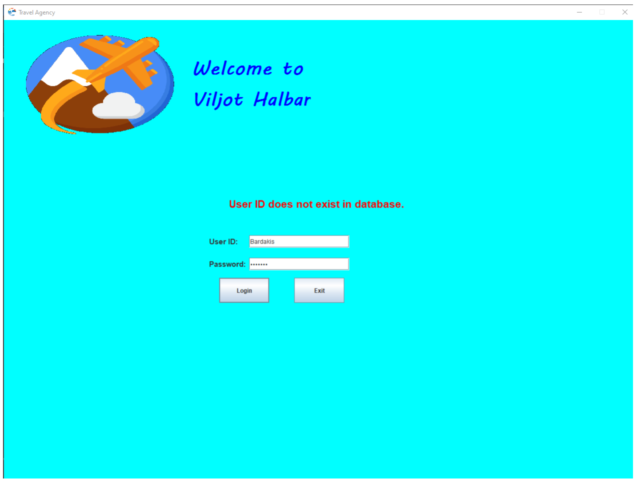


Welcome to
Viljot Halbar

User ID does not exist in database.

User ID: lathosstoixeia
Password: ••••••

Καταχώρηση λανθασμένων στοιχείων. Εμφάνιση λάθους.

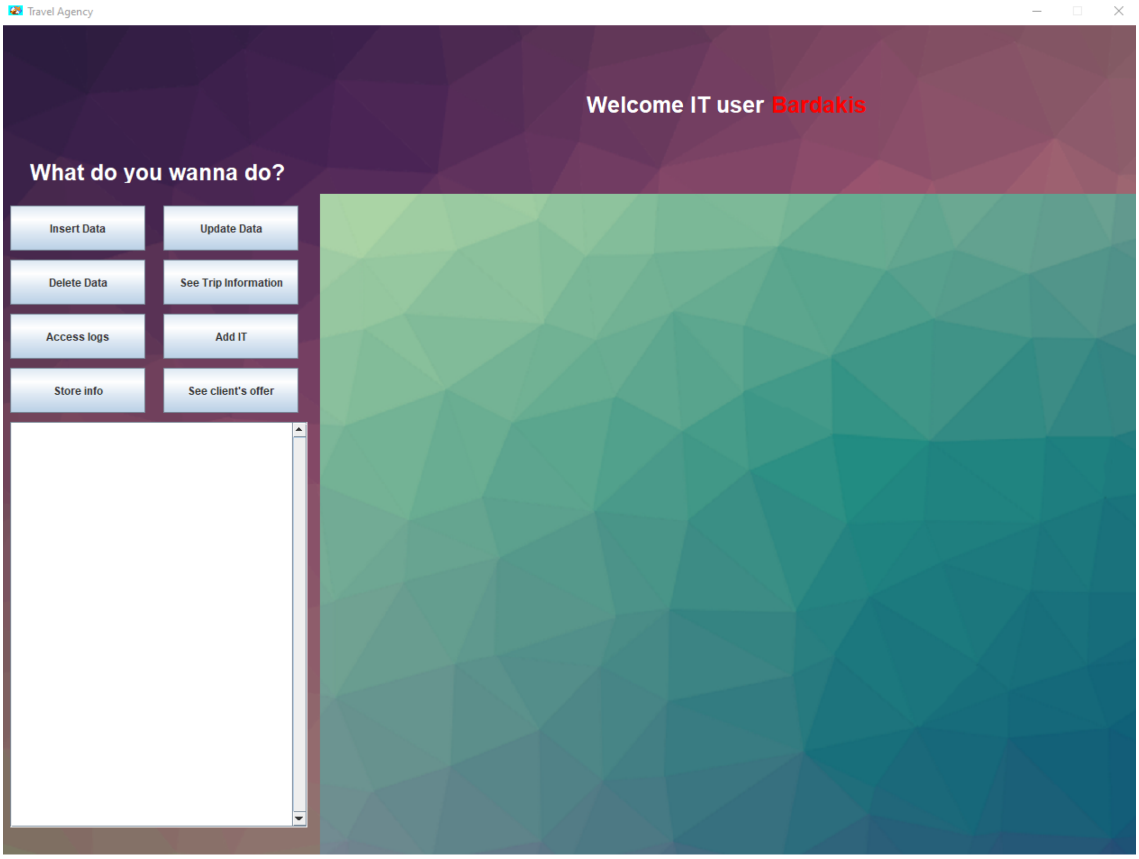


Welcome to
Viljot Halbar

User ID does not exist in database.

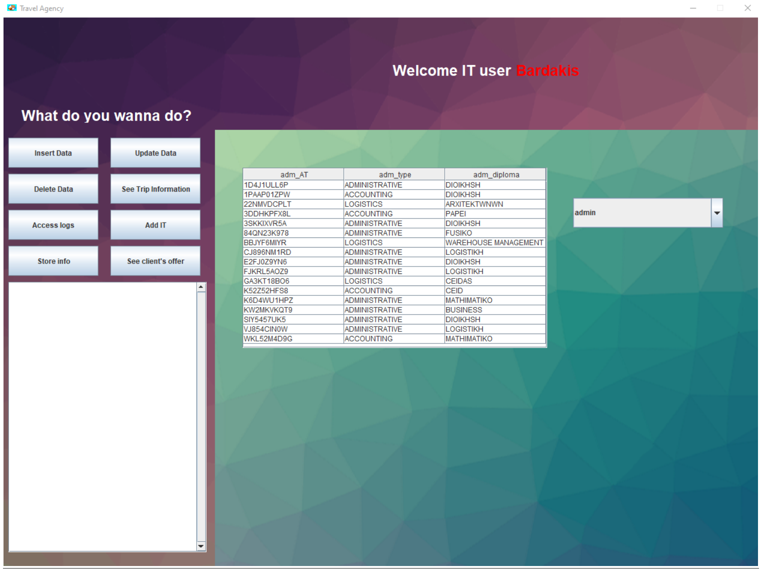
User ID: Bardakis
Password: •••••••

Εισαγωγή σωστών στοιχείων. Είσοδος στο main frame.

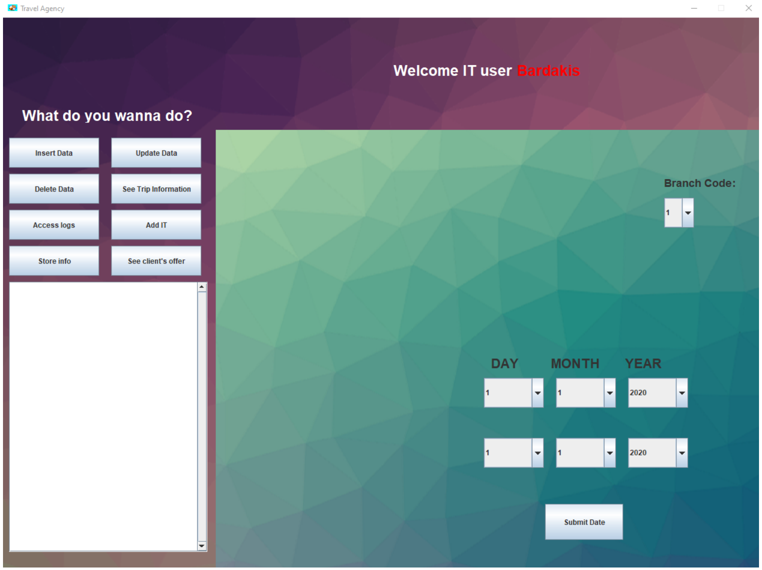


Welcome IT user Bardakis

What do you wanna do?



adm_AT	adm_type	adm_diploma
1D4J1ULL6P	ADMINISTRATIVE	DIOIKHSH
1PAAP01ZPW	ACCOUNTING	DIOIKHSH
22NMVDCPLT	LOGISTICS	ARXITEKTWNWN
3DDHKPFX8L	ACCOUNTING	PAPEI
3SKKIXVR5A	ADMINISTRATIVE	DIOIKHSH
84QN23K978	ADMINISTRATIVE	FUSIKO
BBJYF6MIYR	LOGISTICS	WAREHOUSE MANAGEMENT
CJ896NM1RD	ADMINISTRATIVE	LOGISTIKH
E2FJ0Z9YN6	ADMINISTRATIVE	DIOIKHSH
FJKRL5AOZ9	ADMINISTRATIVE	LOGISTIKH
GA3KT18BO6	LOGISTICS	CEIDAS
K52Z52HFS8	ACCOUNTING	CEID
K6D4WU1HPZ	ADMINISTRATIVE	MATHIMATIKO
KW2MKVKQT9	ADMINISTRATIVE	BUSINESS
SIY5457UK5	ADMINISTRATIVE	DIOIKHSH
VJ854CIN0W	ADMINISTRATIVE	LOGISTIKH
WKL52M4D9G	ACCOUNTING	MATHIMATIKO



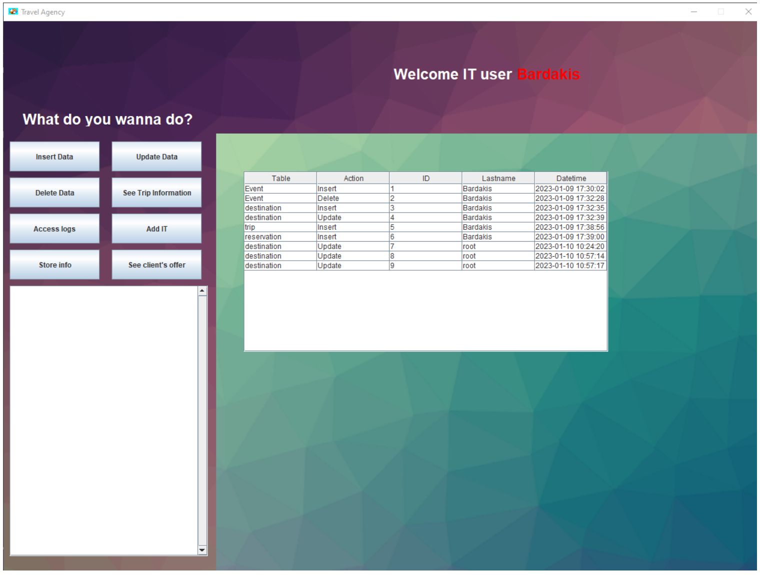
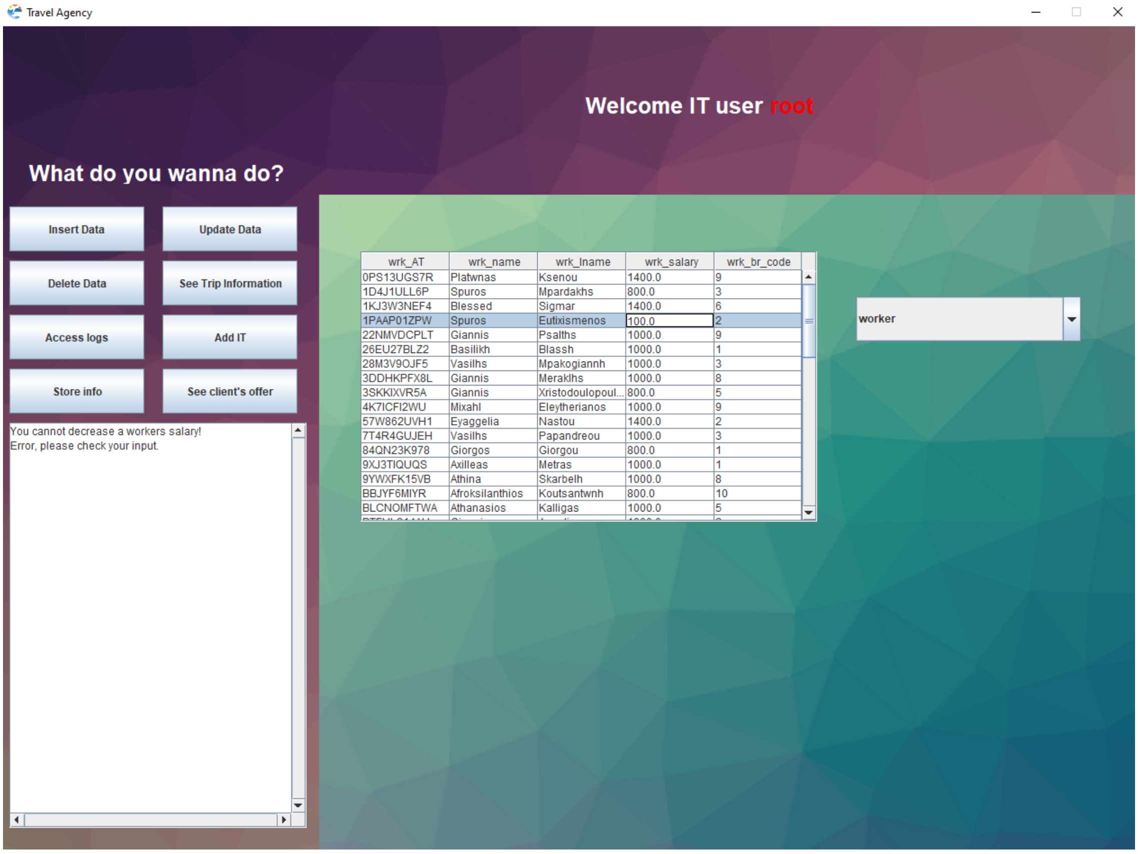


Table	Action	ID	Lastname	Datetime
Event	Insert	1	Bardakis	2023-01-09 17:30:02
Event	Delete	2	Bardakis	2023-01-09 17:32:28
destination	Insert	3	Bardakis	2023-01-09 17:32:35
destination	Update	4	Bardakis	2023-01-09 17:32:39
trip	Insert	5	Bardakis	2023-01-09 17:38:56
reservation	Insert	6	Bardakis	2023-01-09 17:39:00
destination	Update	7	root	2023-01-10 10:24:20
destination	Update	8	root	2023-01-10 10:57:14
destination	Update	9	root	2023-01-10 10:57:17

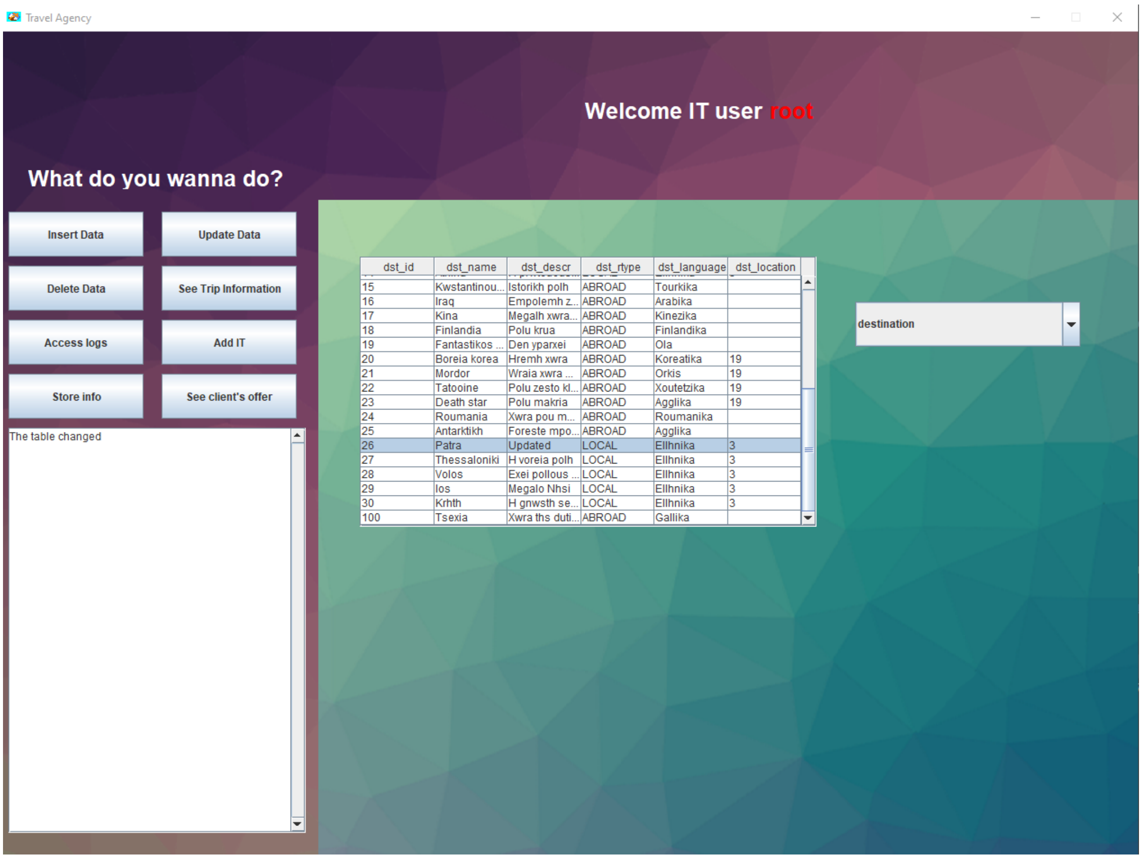
Πάτημα τυχαίων κουμπιών απο τα αριστερά (αναποφάσιστος χρήστης). Σωστή ανανέωση παραθύρων και στοιχείων.



wrk_AT	wrk_name	wrk_lname	wrk_salary	wrk_br_code
0PS13UGS7R	Platwnas	Ksenou	1400.0	9
1D4J1ULL6P	Spuros	Mpardakhs	800.0	3
1KJ3W3NEF4	Blessed	Sigmar	1400.0	6
1PAAP01ZPW	Spuros	Eutixismenos	100.0	2
22NMVDCPLT	Giannis	Psalths	1000.0	9
26EU27BLZ2	Basilikh	Blassh	1000.0	1
28M3V9OJF5	Vasilhs	Mpakogiannh	1000.0	3
3DDHKPFX8L	Giannis	Meraklhs	1000.0	8
3SKKIXVR5A	Giannis	Xristodoulopoul...	800.0	5
4K7ICFI2WU	Mixahl	Eleytherianos	1000.0	9
57W862UVH1	Eyaggelia	Nastou	1400.0	2
7T4R4GUJEH	Vasilhs	Papandreou	1000.0	3
84QN23K978	Giorgos	Giorgou	800.0	1
9XJ3TIQUQS	Axilleas	Metras	1000.0	1
9YWXFK15VB	Athina	Skarbelh	1000.0	8
BBJYF6MIYR	Afroksilanthios	Koutsantwnh	800.0	10
BLCNOMFTWA	Athanasios	Kalligas	1000.0	5

You cannot decrease a workers salary!
Error, please check your input.

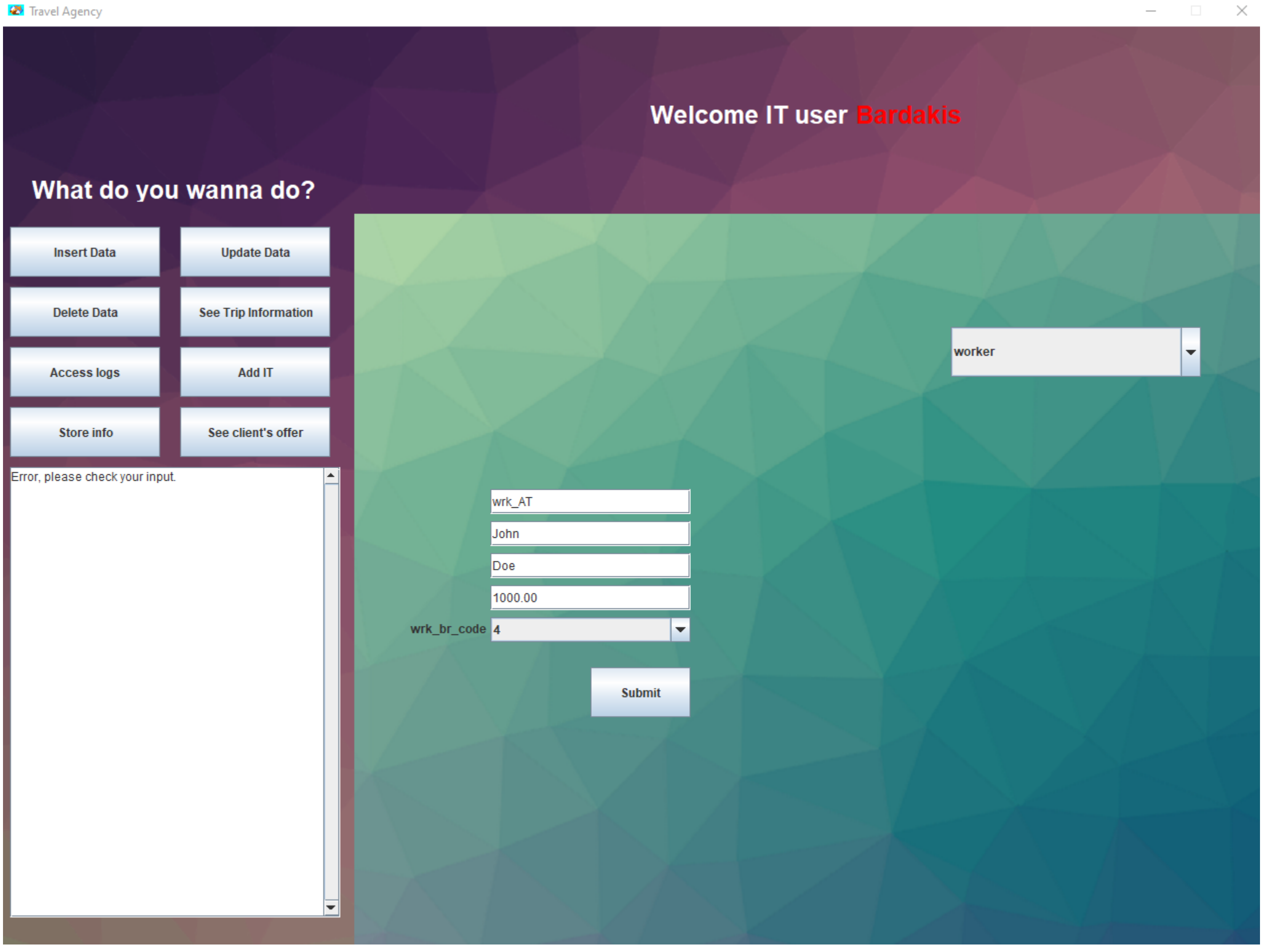
Update Data -> Μείωση μισθού εργάτη. Εμφάνιση λάθους. (Trigger).



dst_id	dst_name	dst_descr	dst_rtype	dst_language	dst_location
15	Kwstantinou...	Istorikh polh	ABROAD	Tourkika	
16	Iraq	Empolemh z...	ABROAD	Arabika	
17	Kina	Megalh xwra...	ABROAD	Kinezika	
18	Finlandia	Polu krua	ABROAD	Finlandika	
19	Fantastikos ...	Den yparxei	ABROAD	Ola	
20	Boreia korea	Hremh xwra	ABROAD	Koreatika	19
21	Mordor	Wraia xwra ...	ABROAD	Orkis	19
22	Tatooine	Polu zesto kl...	ABROAD	Xoutetzika	19
23	Death star	Polu makria	ABROAD	Agglika	19
24	Roumania	Xwra pou m...	ABROAD	Roumanika	
25	Antarktikh	Foreste mpo...	ABROAD	Agglika	
26	Patra	Updated	LOCAL	Ellhnika	3
27	Thessaloniki	H voreia polh	LOCAL	Ellhnika	3
28	Volos	Exei pollous ...	LOCAL	Ellhnika	3
29	Ios	Megalo Nhsi	LOCAL	Ellhnika	3
30	Krhth	H gnwsth se...	LOCAL	Ellhnika	3
100	Tsexia	Xwra ths duti...	ABROAD	Gallika	

The table changed

Update Data -> Ενημέρωση ενός destination. Σωστή ανανέωση δεδομένων.



Error, please check your input.

wrk_AT

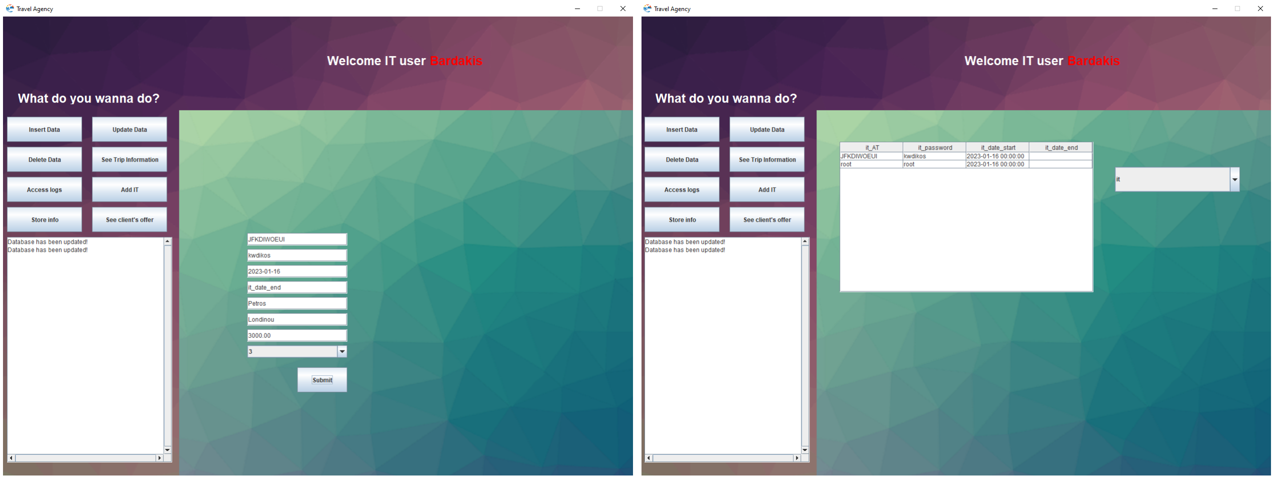
John

Doe

1000.00

wrk_br_code 4

Insert Data -> Worker, αφήνει κενό το AT. Error



Database has been updated!
Database has been updated!

JFKDIWOEUI

kwdikos

2023-01-16

it_date_end

Petros

Londinou

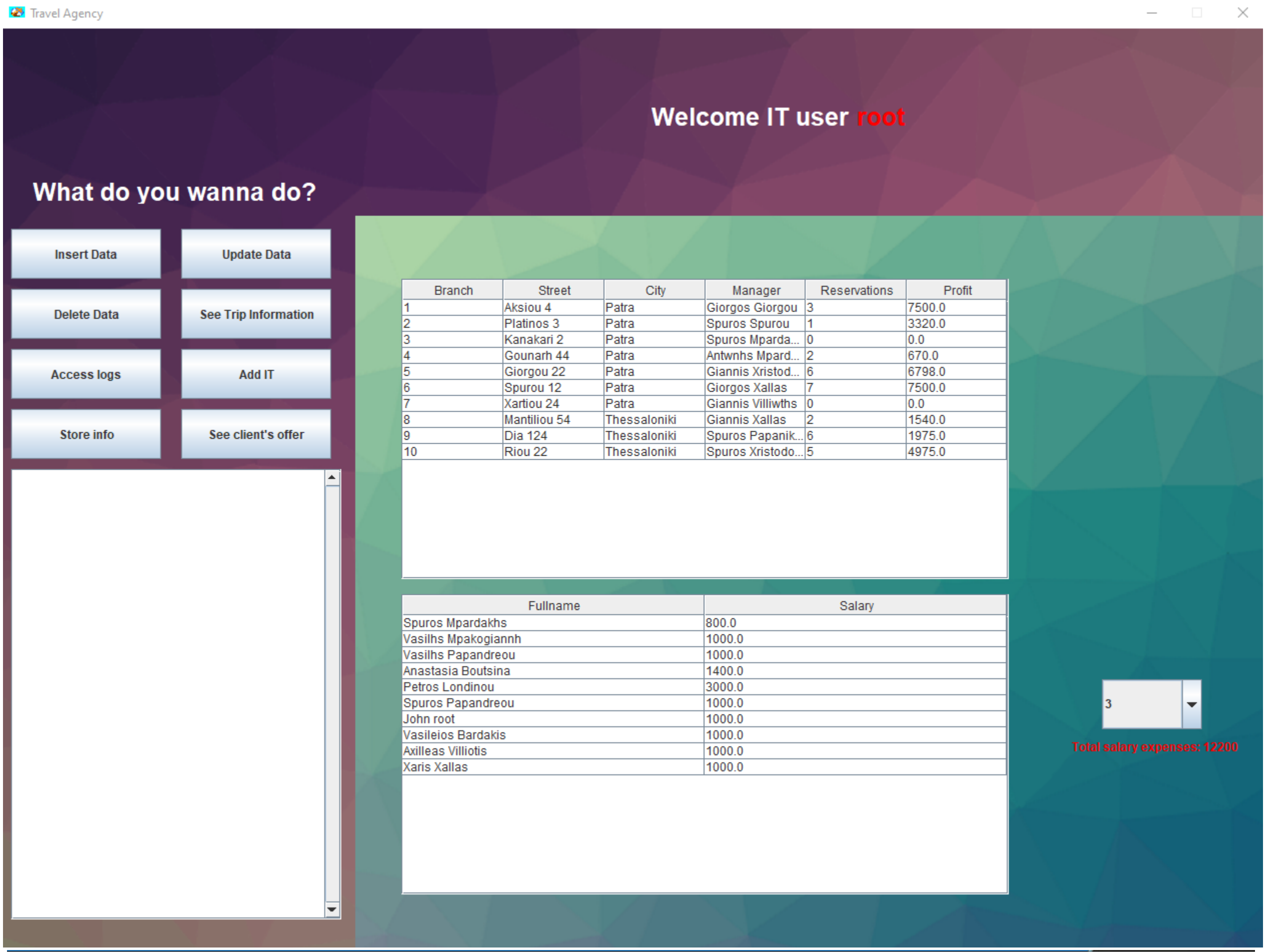
3000.00

3

Database has been updated!
Database has been updated!

it_AT	it_password	it_date_start	it_date_end
JFKDIWOEUI	kwdikos	2023-01-16 00:00:00	
root	root	2023-01-16 00:00:00	

Add IT, αφήνει κενά τα it_date_start και it_date_end και πηγαίνει Update Row -> IT να δει την εγγραφή που έκανε. it_date_start = στιγμή που έγινε το query, it_date_end = null.

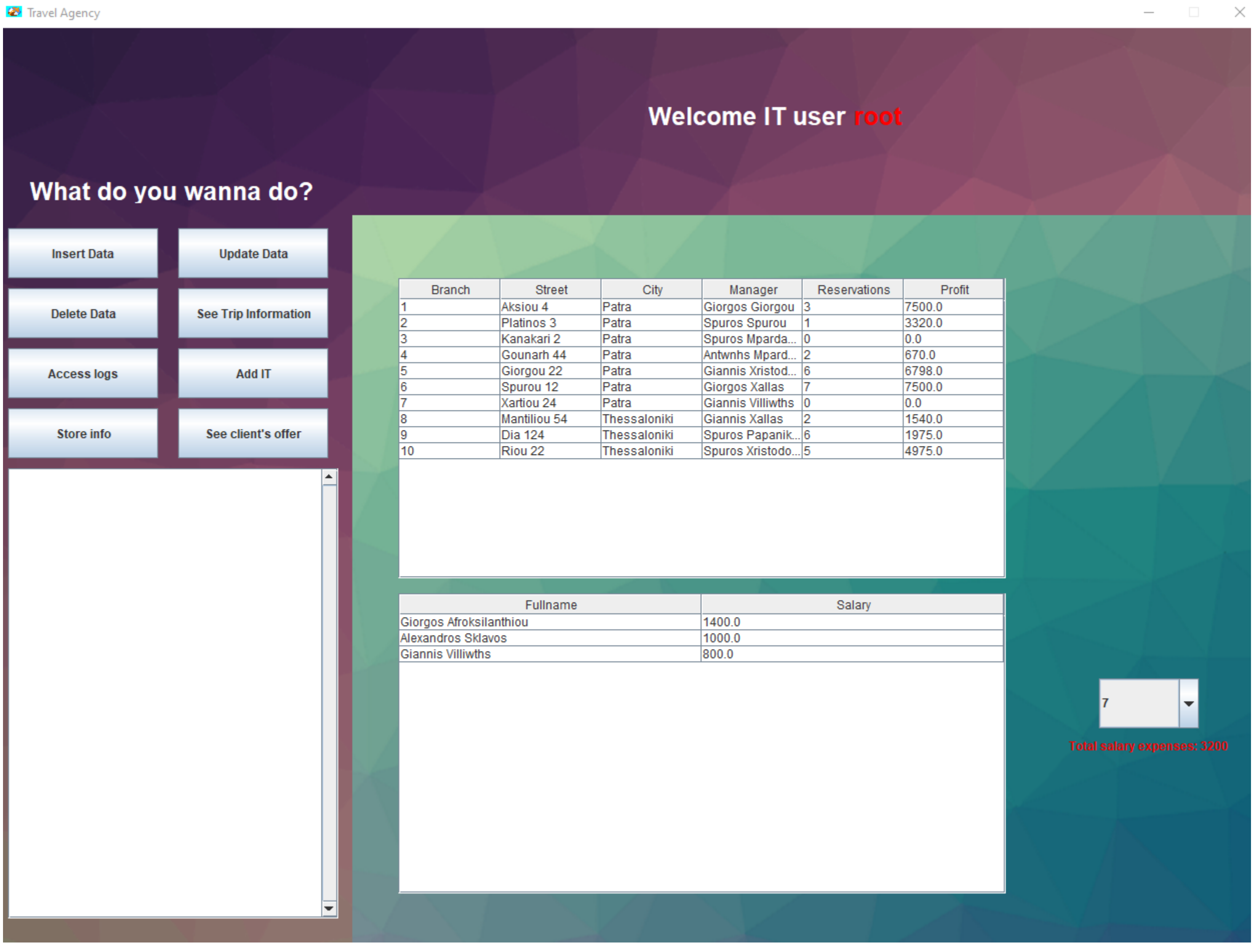


Branch	Street	City	Manager	Reservations	Profit
1	Aksiou 4	Patra	Giorgos Giorgou	3	7500.0
2	Platinos 3	Patra	Spuros Spurou	1	3320.0
3	Kanakari 2	Patra	Spuros Mparda...	0	0.0
4	Gounarh 44	Patra	Antwnhs Mpard...	2	670.0
5	Giorgou 22	Patra	Giannis Xristod...	6	6798.0
6	Spurou 12	Patra	Giorgos Xallas	7	7500.0
7	Xartiou 24	Patra	Giannis Villiwths	0	0.0
8	Mantiliou 54	Thessaloniki	Giannis Xallas	2	1540.0
9	Dia 124	Thessaloniki	Spuros Papanik...	6	1975.0
10	Riou 22	Thessaloniki	Spuros Xristodo...	5	4975.0

Fullname	Salary
Spuros Mpardakhs	800.0
Vasilhs Mpakogiannh	1000.0
Vasilhs Papandreou	1000.0
Anastasia Boutsina	1400.0
Petros Londinou	3000.0
Spuros Papandreou	1000.0
John root	1000.0
Vasileios Bardakis	1000.0
Axilleas Villiotis	1000.0
Xaris Xallas	1000.0

3

Total salary expenses: 12200



Branch	Street	City	Manager	Reservations	Profit
1	Aksiou 4	Patra	Giorgos Giorgou	3	7500.0
2	Platinos 3	Patra	Spuros Spurou	1	3320.0
3	Kanakari 2	Patra	Spuros Mparda...	0	0.0
4	Gounarh 44	Patra	Antwnhs Mpard...	2	670.0
5	Giorgou 22	Patra	Giannis Xristod...	6	6798.0
6	Spurou 12	Patra	Giorgos Xallas	7	7500.0
7	Xartiou 24	Patra	Giannis Villiwths	0	0.0
8	Mantiliou 54	Thessaloniki	Giannis Xallas	2	1540.0
9	Dia 124	Thessaloniki	Spuros Papanik...	6	1975.0
10	Riou 22	Thessaloniki	Spuros Xristodo...	5	4975.0

Fullname	Salary
Giorgos Afroksilanthiou	1400.0
Alexandros Sklavos	1000.0
Giannis Villiwths	800.0

7

Total salary expenses: 3200

Branch info -> Κοιτάζει τους υπαλλήλους ανά branch. Σωστή ανταπόκριση του table.

Μέρος D: Υπόλοιπος Κώδικας

Παρακάτω παραθέτουμε κώδικα τον οποίο δεν δείξαμε πιο πάνω:

D.i 3.1.1_DatabaseSetup.sql

```sql
DROP DATABASE IF EXISTS travel_agency;
CREATE DATABASE travel_agency;
USE travel_agency;

CREATE TABLE branch (
br_code INT(11) NOT NULL AUTO_INCREMENT,
br_street VARCHAR(30) NOT NULL,
br_num INT(4) NOT NULL,
br_city VARCHAR(30) NOT NULL,

PRIMARY KEY(br_code)
);

CREATE TABLE phones(
ph_br_code INT(11) NOT NULL,
ph_number CHAR(10) NOT NULL,

PRIMARY KEY(ph_br_code,ph_number),

CONSTRAINT ph_br_FK
FOREIGN KEY (ph_br_code)
REFERENCES branch(br_code)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE worker (
wrk_AT CHAR(10) NOT NULL,
wrk_name VARCHAR(20) DEFAULT 'unknown' NOT NULL,
wrk_lname VARCHAR(20) DEFAULT 'unknown' NOT NULL,
wrk_salary FLOAT(7,2) NOT NULL,
wrk_br_code INT(11) NOT NULL,

PRIMARY KEY (wrk_AT),

CONSTRAINT wrk_br_code_fk
FOREIGN KEY (wrk_br_code)
REFERENCES branch(br_code)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE driver (
drv_AT CHAR(10) NOT NULL,
drv_license ENUM('A', 'B', 'C', 'D') NOT NULL,
drv_route ENUM('LOCAL', 'ABROAD') NOT NULL,
drv_experience TINYINT(4) NOT NULL,

PRIMARY KEY(drv_AT),

CONSTRAINT drv_AT_FK
FOREIGN KEY (drv_AT)
REFERENCES worker(wrk_AT)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE admin (
adm_AT CHAR(10) NOT NULL,
adm_type ENUM('LOGISTICS', 'ADMINISTRATIVE', 'ACCOUNTING') NOT NULL,
adm_diploma VARCHAR(200) NOT NULL,

PRIMARY KEY(adm_AT),

CONSTRAINT adm_AT_FK
FOREIGN KEY (adm_AT)
REFERENCES worker(wrk_AT)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE guide (
gui_AT CHAR(10) NOT NULL,
gui_cv TEXT NOT NULL,

PRIMARY KEY(gui_AT),

CONSTRAINT gui_AT_FK
FOREIGN KEY (gui_AT)
REFERENCES worker(wrk_AT)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE manages (
mng_adm_AT char(10) NOT NULL,
mng_br_code int(11) NOT NULL,

PRIMARY KEY (mng_adm_AT, mng_br_code),

CONSTRAINT mng_adm_AT_FK
FOREIGN KEY (mng_adm_AT)
REFERENCES admin(adm_AT)
```

```
ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT mng_br_code_FK
FOREIGN KEY (mng_br_code)
REFERENCES branch(br_code)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE languages (
lng_gui_AT char(10) NOT NULL,
lng_language VARCHAR(30) NOT NULL,

PRIMARY KEY (lng_gui_AT,lng_language),

CONSTRAINT lng_gui_AT_FK
FOREIGN KEY(lng_gui_AT)
REFERENCES guide(gui_AT)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE trip (
tr_id INT(11) NOT NULL AUTO_INCREMENT,
tr_departure DATETIME NOT NULL,
tr_return DATETIME NOT NULL,
tr_maxseats TINYINT(4) NOT NULL,
tr_cost FLOAT(7,2) NOT NULL,
tr_br_code INT(11) NOT NULL,
tr_gui_AT CHAR(10) NOT NULL,
tr_drv_AT CHAR(10) NOT NULL,

PRIMARY KEY (tr_id),

CONSTRAINT tr_br_code_FK
FOREIGN KEY(tr_br_code)
REFERENCES branch(br_code)
ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT tr_gui_AT_FK
FOREIGN KEY(tr_gui_AT)
REFERENCES guide(gui_AT)
ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT tr_drv_AT_FK
FOREIGN KEY(tr_drv_AT)
REFERENCES driver(drv_AT)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

```sql
CREATE TABLE event (
ev_tr_id INT(11) NOT NULL AUTO_INCREMENT,
ev_start DATETIME NOT NULL,
ev_end DATETIME NOT NULL,
ev_descr TEXT NOT NULL,

PRIMARY KEY (ev_tr_id, ev_start),

CONSTRAINT ev_tr_id_FK
FOREIGN KEY(ev_tr_id)
REFERENCES trip(tr_id)
ON UPDATE CASCADE ON DELETE CASCADE
);

-- seatnum can be shared in same tr_id
CREATE TABLE reservation (
res_tr_id int(11) NOT NULL AUTO_INCREMENT,
res_seatnum tinyint(4) NOT NULL,
res_name varchar(20) NOT NULL,
res_lname varchar(20) NOT NULL,
res_isadult enum('ADULT', 'MINOR') NOT NULL,

PRIMARY KEY (res_tr_id, res_seatnum),

CONSTRAINT res_tr_id_FK
FOREIGN KEY (res_tr_id)
REFERENCES trip(tr_id)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE destination(
dst_id int(11) NOT NULL AUTO_INCREMENT,
dst_name varchar(50) NOT NULL,
dst_descr text NOT NULL,
dst_rtype enum('LOCAL', 'ABROAD') NOT NULL,
dst_language varchar(30) NOT NULL,
dst_location int(11),

PRIMARY KEY(dst_id),

CONSTRAINT dst_loc_FK
FOREIGN KEY (dst_location)
REFERENCES destination(dst_id)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

```sql
CREATE TABLE travel_to (
to_tr_id int(11) NOT NULL,
to_dst_id int(11) NOT NULL,
to_arrival datetime NOT NULL,
to_departure datetime NOT NULL,

PRIMARY KEY(to_tr_id, to_dst_id),

CONSTRAINT trv_to_tr_id_FK
FOREIGN KEY (to_tr_id)
REFERENCES trip(tr_id)
ON UPDATE CASCADE ON DELETE CASCADE,

CONSTRAINT trv_to_dst_id_FK
FOREIGN KEY (to_dst_id)
REFERENCES destination(dst_id)
ON UPDATE CASCADE ON DELETE CASCADE
);

-- INSERTS

INSERT INTO branch VALUES
(NULL,'Aksiou',4,'Patra'),
(NULL,'Platinos',3,'Patra'),
(NULL,'Kanakari',2,'Patra'),
(NULL,'Gounarh',44,'Patra'),
(NULL,'Giorgou',22,'Patra'),
(NULL,'Spurou',12,'Patra'),
(NULL,'Xartiou',24,'Patra'),
(NULL,'Mantiliou',54,'Thessaloniki'),
(NULL,'Dia',124,'Thessaloniki'),
(NULL,'Riou',22,'Thessaloniki');

INSERT INTO phones VALUES
(1,'2103679080'),
(1,'2106609976'),
(2,'2105813102'),
(2,'2103864592'),
(2,'2105201926'),
(2,'2106081529'),
(3,'2108772622'),
(4,'2105908149'),
(5,'2101017600'),
(5,'2104298896'),
(5,'2105567514'),
(6,'2109819713'),
(7,'2107691209'),
(7,'2106814634'),
(8,'2101812989'),
(8,'2106584920'),
(8,'2100545788'),
(8,'2100955415'),
(8,'2108780556'),
(9,'2100532081'),
(10,'2104623439'),
(10,'2107259204'),
(10,'2100693816'),
(10,'2101355405');


INSERT INTO worker VALUES
-- admins
('84QN23K978','Giorgos','Giorgou',800,1),
('WKL52M4D9G','Xarhs','Magkas',1000,1),
('K6D4WU1HPZ','Spuros','Spurou',800,2),
('1PAAP01ZPW','Spuros','Eutixismenos',1000,2),
('1D4J1ULL6P','Spuros','Mpardakhs',800,3),
('SIY5457UK5','Antwnhs','Mpardakhs',800,4),
('3SKKIXVR5A','Giannis','Xristodoulopoulou',800,5),
('K52Z52HFS8','Thodwrhs','Sorras',1000,5),
('GA3KT18BO6','Giorgos','Villiwths',800,5),
('E2FJ0Z9YN6','Giorgos','Xallas',800,6),
('VJ854CIN0W','Giannis','Villiwths',800,7),
('FJKRL5AOZ9','Giannis','Xallas',800,8),
('3DDHKPFX8L','Giannis','Meraklhs',1000,8),
('CJ896NM1RD','Spuros','Papanikolaou',800,9),
('22NMVDCPLT','Giannis','Psalths',1000,9),
('KW2MKVKQT9','Spuros','Xristodoulopoulou',800,10),
('BBJYF6MIYR','Afroksilanthios','Koutsantwnh',800,10),
-- drivers
('9XJ3TIQUQS','Axilleas','Metras',1000,1),
('S868640HEA','Antwnhs','Voudas',1000,2),
('BTFUL214AU','Giannis','Anestiou',1000,2),
('N84ERH1B1D','Spuros','Papandreou',1000,3),
('7T4R4GUJEH','Vasilhs','Papandreou',1000,3),
('28M3V9OJF5','Vasilhs','Mpakogiannh',1000,3),
('HAK6C32A0L','Giannis','Papandreou',1000,4),
('PNS7SFQMY7','Neoptolemos','Xristou',1000,4),
('FP02ESO9T5','Giorgos','Papathanasiou',1400,5),
('1KJ3W3NEF4','Blessed','Sigmar',1400,6),
('DK9GGY0HY3','Giorgos','Afroksilanthiou',1400,7),
('FVRG67UC7G','Spuros','Fontas',1400,8),
('0PS13UGS7R','Platwnas','Ksenou',1400,9),
('RH31B5HDZ5','Maria','Lampridh',1000,9),
('S88MJCJXOD','Elenh','Lagou',1000,9),
('SCK7F68FHV','Giwrgos','Barwtsos',1000,10),
-- guides
('26EU27BLZ2','Basilikh','Blassh',1000,1),
('L91VRQ11EJ','Sofia','Xristou',1000,1),
('57W862UVH1','Eyaggelia','Nastou',1400,2),
('DUONPWMWVA','Anastasia','Boutsina',1400,3),
('XSJ73VK013','Eirhnh','Xanthopoulou',1400,4),
('H3Z8OW7P9F','Panagiwta','Panou',1400,4),
('PQ2995IOH5','Panagiwths','Ebert',1400,5),
('BLCNOMFTWA','Athanasios','Kalligas',1000,5),
('WJ8ZWJG6E8','Alexandra','Kaloghrou',1000,6),
('U648POE7TO','Alexandros','Sklavos',1000,7),
('9YWXFK15VB','Athina','Skarbelh',1000,8),
('4K7ICFI2WU','Mixahl','Eleytherianos',1000,9),
('H6AAOHV6QO','Stylianh','Daskalakh',1400,9),
('KC7BJJ2RWA','Andreas','Papakwnstantinou',1400,9),
('NI5S6NTKW6','Apostolos','Athanasiadhs',1400,9),
('XUO90F1U2I','Petros','Anastasiou',1400,10),
('EMNLLE1BAW','Thwmas','Kontoleon',1400,10);


INSERT INTO admin VALUES
('84QN23K978','ADMINISTRATIVE','FUSIKO'),
('WKL52M4D9G','ACCOUNTING','MATHIMATIKO'),
('K6D4WU1HPZ','ADMINISTRATIVE','MATHIMATIKO'),
('1PAAP01ZPW','ACCOUNTING','DIOIKHSH'),
('1D4J1ULL6P','ADMINISTRATIVE','DIOIKHSH'),
('SIY5457UK5','ADMINISTRATIVE','DIOIKHSH'),
('3SKKIXVR5A','ADMINISTRATIVE','DIOIKHSH'),
('K52Z52HFS8','ACCOUNTING','CEID'),
('GA3KT18BO6','LOGISTICS','CEID'),
('E2FJOZ9YN6','ADMINISTRATIVE','DIOIKHSH'),
('VJ854CINOW','ADMINISTRATIVE','LOGISTIKH'),
('FJKRL5AOZ9','ADMINISTRATIVE','LOGISTIKH'),
('3DDHKPFX8L','ACCOUNTING','PAPEI'),
('CJ896NM1RD','ADMINISTRATIVE','LOGISTIKH'),
('22NMVDCPLT','LOGISTICS','ARXITEKTWNWN'),
('KW2MKVKQT9','ADMINISTRATIVE','BUSINESS'),
('BBJYF6MIYR','LOGISTICS','WAREHOUSE MANAGEMENT');


INSERT INTO manages VALUES
('84QN23K978',1),
('K6D4WU1HPZ',2),
('1D4J1ULL6P',3),
('SIY5457UK5',4),
('3SKKIXVR5A',5),
('E2FJ0Z9YN6',6),
('VJ854CINOW',7),
('FJKRL5AOZ9',8),
('CJ896NM1RD',9),
('KW2MKVKQT9',10);


INSERT INTO guide VALUES
('26EU27BLZ2', '19 xronia sthn douleia'),
('L91VRQ11EJ', '3 xronia sthn douleia'),
('DUONPWMWVA', '4 xronia sthn douleia'),
('57W862UVH1', '11 xronia sthn douleia'),
('XSJ73VK013', '18 xronia sthn douleia'),
('H3Z8OW7P9F', '20 xronia sthn douleia'),
('PQ2995IOH5', '5 xronia sthn douleia'),
('BLCNOMFTWA', '7 xronia sthn douleia'),
('WJ8ZWJG6E8', '20 xronia sthn douleia'),
('U648POE7TO', '6 xronia sthn douleia'),
('9YWXFK15VB', '17 xronia sthn douleia'),
('4K7ICFI2WU', '4 xronia sthn douleia'),
('H6AA0HV6QO', '3 xronia sthn douleia'),
('KC7BJJ2RWA', '24 xronia sthn douleia'),
('NI5S6NTKW6', '5 xronia sthn douleia'),
('XUO90F1U2I', '0 xronia sthn douleia'),
('EMNLLE1BAW', '3 xronia sthn douleia');


INSERT INTO driver VALUES
('9XJ3TIQUQS','B','ABROAD',54),
('S868640HEA','C','ABROAD',119),
('BTFUL214AU','D','ABROAD',118),
('N84ERH1B1D','C','LOCAL',98),
('7T4R4GUJEH','A','LOCAL',91),
('28M3V90JF5','D','ABROAD',0),
('HAK6C32A0L','D','ABROAD',75),
('PNS7SFQMY7','C','ABROAD',97),
('1KJ3W3NEF4','B','ABROAD',4),
('FP02ES09T5','D','ABROAD',38),
('DK9GGY0HY3','C','ABROAD',56),
('FVRG67UC7G','A','LOCAL',85),
('0PS13UGS7R','B','LOCAL',98 ),
('RH31B5HDZ5','B','LOCAL',37 ),
('S88MJCJX0D','D','ABROAD',91),
('SCK7F68FHV','C','LOCAL',56 );
```

```
INSERT INTO languages VALUES
('26EU27BLZ2', 'Agglika'),
('L91VRQ11EJ', 'Agglika'),
('DUONPWMWVA', 'Gallika'),
('DUONPWMWVA', 'Agglika'),
('57W862UVH1', 'Agglika'),
('57W862UVH1', 'Kinezika'),
('XSJ73VK013', 'Agglika'),
('H3Z80W7P9F', 'Agglika'),
('PQ2995IOH5', 'Agglika'),
('PQ2995IOH5', 'Germanika'),
('BLCNOMFTWA', 'Agglika'),
('WJ8ZWJG6E8', 'Agglika'),
('WJ8ZWJG6E8', 'Italika'),
('U648P0E7T0', 'Agglika'),
('9YWXFK15VB', 'Agglika'),
('4K7ICFI2WU', 'Agglika'),
('4K7ICFI2WU', 'Ispanika'),
('H6AA0HV6Q0', 'Agglika'),
('KC7BJJ2RWA', 'Agglika'),
('KC7BJJ2RWA', 'Iapwnika'),
('NI5S6NTKW6', 'Agglika'),
('XU090F1U2I', 'Agglika'),
('EMNLLE1BAW', 'Agglika');


INSERT INTO trip VALUES
(NULL, '2022-05-20', '2022-05-28', 25, 1000, 6,'WJ8ZWJG6E8' ,'1KJ3W3NEF4'),
(NULL, '2022-01-13', '2022-01-20', 15, 1133, 5,'PQ2995IOH5' ,'FP02ES09T5'),
(NULL, '2022-06-06', '2022-06-16', 20, 1200, 10,'EMNLLE1BAW','SCK7F68FHV'),
(NULL, '2022-12-15', '2022-12-20', 15, 875, 9, 'KC7BJJ2RWA' ,'RH31B5HDZ5'),
(NULL, '2023-11-17', '2023-11-30', 30, 2500, 1,'26EU27BLZ2' ,'9XJ3TIQUQS'),
(NULL, '2022-03-01', '2022-03-15', 40, 3320, 2,'57W862UVH1' ,'S868640HEA'),
(NULL, '2022-06-08', '2022-06-12', 15, 770, 8, '9YWXFK15VB' ,'FVRG67UC7G'),
(NULL, '2022-07-22', '2022-07-25', 12, 335, 4, 'XSJ73VK013' ,'PNS7SFQMY7'),
(NULL, '2023-11-29', '2023-12-02', 35, 175, 10,'XU090F1U2I' ,'SCK7F68FHV'),
(NULL, '2022-04-27', '2022-05-02', 27, 220, 9, 'KC7BJJ2RWA' ,'RH31B5HDZ5'),
(NULL, '2023-05-20', '2023-05-28', 25, 1000, 7,'U648P0E7T0' ,'DK9GGY0HY3'),
(NULL, '2023-01-13', '2023-01-20', 15, 1133, 1,'26EU27BLZ2' ,'9XJ3TIQUQS'),
(NULL, '2023-06-06', '2023-06-16', 20, 1200, 7,'U648P0E7T0' ,'DK9GGY0HY3'),
(NULL, '2023-12-15', '2023-12-20', 15, 875, 6, 'WJ8ZWJG6E8' ,'1KJ3W3NEF4'),
(NULL, '2021-11-17', '2021-11-30', 30, 2500, 9,'KC7BJJ2RWA' ,'RH31B5HDZ5'),
(NULL, '2023-03-01', '2023-03-15', 40, 3320, 3,'DUONPWMWVA' ,'N84ERH1B1D'),
(NULL, '2023-06-08', '2023-06-12', 15, 770, 6, 'WJ8ZWJG6E8' ,'1KJ3W3NEF4'),
(NULL, '2023-07-22', '2023-07-25', 12, 335, 5, 'PQ2995IOH5' ,'FP02ES09T5'),
(NULL, '2021-11-29', '2021-12-02', 35, 175, 10,'XU090F1U2I' ,'SCK7F68FHV'),
(NULL, '2023-04-27', '2023-05-02', 27, 220, 7, 'U648P0E7T0' ,'DK9GGY0HY3'),
(NULL, '2021-05-20', '2021-05-28', 25, 1000, 8,'9YWXFK15VB' ,'FVRG67UC7G'),
(NULL, '2021-01-13', '2021-01-20', 15, 1133, 6,'WJ8ZWJG6E8' ,'1KJ3W3NEF4'),
(NULL, '2021-06-06', '2021-06-16', 20, 1200, 5,'PQ2995IOH5' ,'FP02ES09T5'),
(NULL, '2021-12-15', '2021-12-20', 15, 875, 5, 'BLCNOMFTWA' ,'FP02ES09T5'),
(NULL, '2024-11-17', '2024-11-30', 30, 2500, 10,'XU090F1U2I','SCK7F68FHV'),
(NULL, '2021-03-01', '2021-03-15', 40, 3320, 10,'EMNLLE1BAW','SCK7F68FHV'),
(NULL, '2021-06-08', '2021-06-12', 15, 770, 5, 'PQ2995IOH5' ,'FP02ES09T5'),
(NULL, '2021-07-22', '2021-07-25', 12, 335, 7, 'U648POE7TO' ,'DK9GGY0HY3'),
(NULL, '2024-11-29', '2024-12-02', 35, 175, 9,'NI5S6NTKW6'  ,'0PS13UGS7R'),
(NULL, '2021-04-27', '2021-05-02', 27, 220, 9, 'NI5S6NTKW6' ,'0PS13UGS7R');


INSERT INTO event VALUES
(1, '2022-05-20', '2022-05-25', 'tsagaki'),
(1, '2022-05-25', '2022-05-26', 'vouno'),
(1, '2022-05-26', '2022-05-28', 'ski'),
(2, '2022-01-13', '2022-01-14', 'epiviwsh sthn fysh'),
(2, '2022-01-14', '2022-01-17', 'live metallica'),
(2, '2022-01-17', '2022-01-20', 'sauna'),
(3, '2022-06-06', '2022-06-07', 'make-a-wish'),
(3, '2022-06-09', '2022-06-15', 'akrivo estiatorio'),
(3, '2022-06-15', '2022-06-16', 'oxi toso akrivo estiatorio'),
(4, '2022-12-15', '2022-12-16', 'polemontas elf'),
(4, '2022-12-17', '2022-12-18', 'dnd session'),
(4, '2022-12-18', '2022-12-20', 'epistrofh sthn pragmatikothta'),
(5, '2023-11-17', '2023-11-18', '50cent live'),
(5, '2023-11-19', '2023-11-22', 'argyros live'),
(5, '2023-11-23', '2023-11-25', 'telos ta live, tyropita'),
(5, '2023-11-25', '2023-11-30', 'tyropita KAI tsai'),
(6, '2022-03-01', '2022-03-05', 'kalosorisma'),
(6, '2022-03-07', '2022-03-09', 'anavash sto everest'),
(6, '2022-03-10', '2022-03-12', 'nerotsoulithres'),
(6, '2022-03-12', '2022-03-15', 'exodos'),
(7, '2022-06-08', '2022-06-09', 'thalassa'),
(7, '2022-06-09', '2022-06-10', 'taverna'),
(7, '2022-06-10', '2022-06-11', 'kai allh taverna'),
(7, '2022-06-11', '2022-06-12', 'ksekourash sto aerodromio'),
(8, '2022-07-22', '2022-07-23', 'ypodoxh, check-in'),
(8, '2022-07-23', '2022-07-25', 'aksiotheata'),
(9, '2023-11-29', '2023-12-01', 'ki alla aksiotheata'),
(9, '2023-12-01', '2023-12-02', 'ksekourash prin thn epistrofh'),
(10,'2022-04-27', '2022-04-29', 'kalosorisma'),
(10,'2022-04-29', '2022-05-02', 'zwologikos khpos'),
(11,'2023-05-20', '2023-05-25', 'tsagaki'),
(11,'2023-05-25', '2023-05-26', 'vouno'),
(11,'2023-05-26', '2023-05-28', 'ski'),
(12,'2023-01-13', '2023-01-14', 'epiviwsh sthn fysh'),
(12,'2023-01-14', '2023-01-17', 'live metallica'),
(12,'2023-01-17', '2023-01-20', 'sauna'),
(13,'2023-06-06', '2023-06-07', 'make-a-wish'),
(13,'2023-06-09', '2023-06-15', 'akrivo estiatorio'),
(13,'2023-06-15', '2023-06-16', 'oxi toso akrivo estiatorio'),
(14,'2023-12-15', '2023-12-16', 'polemontas elf'),
(14,'2023-12-17', '2023-12-18', 'dnd session'),
(14,'2023-12-18', '2023-12-20', 'epistrofh sthn pragmatikothta'),
(15,'2021-11-17', '2021-11-18', '50cent live'),
(15,'2021-11-19', '2021-11-22', 'argyros live'),
(15,'2021-11-23', '2021-11-25', 'telos ta live, tyropita'),
(15,'2021-11-25', '2021-11-30', 'tyropita KAI tsai'),
(16,'2023-03-01', '2023-03-05', 'kalosorisma'),
(16,'2023-03-07', '2023-03-09', 'anavash sto everest'),
(16,'2023-03-10', '2023-03-12', 'nerotsoulithres'),
(16,'2023-03-12', '2023-03-15', 'exodos'),
(17,'2023-06-08', '2023-06-09', 'thalassa'),
(17,'2023-06-09', '2023-06-10', 'taverna'),
(17,'2023-06-10', '2023-06-11', 'kai allh taverna'),
(17,'2023-06-11', '2023-06-12', 'ksekourash sto aerodromio'),
(18,'2023-07-22', '2023-07-23', 'ypodoxh, check-in'),
(18,'2023-07-23', '2023-07-25', 'aksiotheata'),
(19,'2021-11-29', '2021-12-01', 'ki alla aksiotheata'),
(19,'2021-12-01', '2021-12-02', 'ksekourash prin thn epistrofh'),
(20,'2023-04-27', '2023-04-29', 'kalosorisma'),
(20,'2023-04-29', '2023-05-02', 'zwologikos khpos'),
(21,'2021-05-20', '2021-05-25', 'tsagaki'),
(21,'2021-05-25', '2021-05-26', 'vouno'),
(21,'2021-05-26', '2021-05-28', 'ski'),
(22,'2021-01-13', '2021-01-14', 'epiviwsh sthn fysh'),
(22,'2021-01-14', '2021-01-17', 'live metallica'),
(22,'2021-01-17', '2021-01-20', 'sauna'),
(23,'2021-06-06', '2021-06-07', 'make-a-wish'),
(23,'2021-06-09', '2021-06-15', 'akrivo estiatorio'),
(23,'2021-06-15', '2021-06-16', 'oxi toso akrivo estiatorio'),
(24,'2021-12-15', '2021-12-16', 'polemontas elf'),
(24,'2021-12-17', '2021-12-18', 'dnd session'),
(24,'2021-12-18', '2021-12-20', 'epistrofh sthn pragmatikothta'),
(25,'2024-11-17', '2024-11-18', '50cent live'),
(25,'2024-11-19', '2024-11-22', 'argyros live'),
(25,'2024-11-23', '2024-11-25', 'telos ta live, tyropita'),
(25,'2024-11-25', '2024-11-30', 'tyropita KAI tsai'),
(26,'2021-03-01', '2021-03-05', 'kalosorisma'),
(26,'2021-03-07', '2021-03-09', 'anavash sto everest'),
(26,'2021-03-10', '2021-03-12', 'nerotsoulithres'),
(26,'2021-03-12', '2021-03-15', 'exodos'),
(27,'2021-06-08', '2021-06-09', 'thalassa'),
(27,'2021-06-09', '2021-06-10', 'taverna'),
(27,'2021-06-10', '2021-06-11', 'kai allh taverna'),
(27,'2021-06-11', '2021-06-12', 'ksekourash sto aerodromio'),
(28,'2021-07-22', '2021-07-23', 'ypodoxh, check-in'),
(28,'2021-07-23', '2021-07-25', 'aksiotheata'),
(29,'2024-11-29', '2024-12-01', 'ki alla aksiotheata'),
(29,'2024-12-01', '2024-12-02', 'ksekourash prin thn epistrofh'),
(30,'2021-04-27', '2021-04-29', 'kalosorisma'),
(30,'2021-04-29', '2021-05-02', 'zwologikos khpos');

INSERT INTO reservation VALUES
(1, 23, 'Giannis', 'Aggelidis', 'ADULT'),
(1, 22, 'Giannis', 'Aggelidis', 'ADULT'),
(1, 21, 'Giannis', 'Aggelidis', 'ADULT'),
(1, 20, 'Giannis', 'Aggelidis', 'ADULT'),
(1, 10, 'Thanos', 'Mylwnas', 'ADULT'),
(1, 9, 'Giannhs', 'Mylwnas', 'MINOR'),
(2, 15, 'Markos', 'Loberdos', 'ADULT'),
(2, 14, 'Markos', 'Loberdos', 'ADULT'),
(2, 13, 'Markos', 'Loberdos', 'ADULT'),
(2, 9, 'Barbara', 'Stathopoulou', 'ADULT'),
(2, 7, 'Barbara', 'Stathopoulou', 'ADULT'),
(2, 10, 'Barbara', 'Iremh', 'ADULT'),
(3, 20, 'Dhmhtrhs', 'Bergas', 'MINOR'),
(3, 19, 'Giannhs', 'Tsapas', 'MINOR'),
(3, 18, 'Eirhnh', 'Mpezou', 'ADULT'),
(3, 11, 'Dhmhtra', 'Karavia', 'ADULT'),
(4, 15, 'Makhs', 'Mamos', 'ADULT'),
(5, 30, 'Anastashs', 'Boulgarhs', 'ADULT'),
(5, 19, 'Grhgorhs', 'Grhgoriou', 'ADULT'),
(5, 09, 'Anna', 'Gennhmata', 'ADULT'),
(6, 40, 'Giannis', 'Markopoulos', 'ADULT'),
(7, 12, 'Takhs', 'Palamas', 'ADULT'),
(7, 08, 'Giannis', 'Mamos', 'ADULT'),
(8, 08, 'Elenh', 'Mosxou', 'ADULT'),
(8, 4, 'Xaralampos', 'Kexagias', 'ADULT'),
(9, 10, 'Maria', 'Panou', 'ADULT'),
(10, 09, 'Giannis', 'Samaras', 'ADULT'),
(10, 12, 'Anastasia', 'Gewrgiou', 'ADULT'),
(10, 11, 'Anastasis', 'Nirvanas', 'ADULT'),
(10, 5, 'Super', 'Marios', 'ADULT'),
(10, 6, 'Super', 'Luigis', 'ADULT');


INSERT INTO destination VALUES
(null, 'Gallia', 'Xwra ths dutikhs Eurwphs', 'ABROAD', 'Gallika', null),
(null, 'Parisi', 'Romantikh polh', 'ABROAD', 'Gallika', 1),
(null, 'Ellada', 'H Ellada', 'LOCAL', 'Ellhnika', null),
(null, 'Salamina', 'Tapeino xorioudaki', 'LOCAL', 'Ellhnika', 3),
(null, 'Dubai', 'Plousia polh', 'ABROAD', 'Ellhnika', null),
(null, 'Tokyo', 'Polh ths iapwnias me jdm amaksia', 'ABROAD', 'Iapwnika', null),
(null, 'Norbhgia', 'Polu krua', 'ABROAD', 'Norbhgika', null),
(null, 'Mosxa', 'Polu krua', 'ABROAD', 'Rwssika', null),
(null, 'Benetia', 'Wraia polh me kanalia', 'ABROAD', 'Italika', null),
(null, 'Florida', 'Polh Amerikhs pou exei disneyland', 'ABROAD', 'Agglika', null),
(null, 'Boulgaria', 'Fthina emporika kentra', 'ABROAD', 'Boulgarika', null),
(null, 'Notia afrikh', 'Tropiko klima', 'ABROAD', 'Gallika', null),
(null, 'Elbetia', 'Wraia xwra me CERN', 'ABROAD', 'Elbetika', null),
(null, 'Athina', 'H prwteuousa ths Elladas', 'LOCAL', 'Ellhnika', 3),
(null, 'Kwstantinoupolh', 'Istorikh polh', 'ABROAD', 'Tourkika', null),
(null, 'Iraq', 'Empolemh zwnh', 'ABROAD', 'Arabika', null),
(null, 'Kina', 'Megalh xwra me wraia bouna', 'ABROAD', 'Kinezika', null),
(null, 'Finlandia', 'Polu krua', 'ABROAD', 'Finlandika', null),
(null, 'Fantastikos Komsmos', 'Den yparxei', 'ABROAD', 'Ola', null),
(null, 'Boreia korea', 'Hremh xwra', 'ABROAD', 'Koreatika', 19),
(null, 'Mordor', 'Wraia xwra me zesto klima', 'ABROAD', 'Orkis', 19),
(null, 'Tatooine', 'Polu zesto klima', 'ABROAD', 'Xoutetzika', 19),
(null, 'Death star', 'Polu makria', 'ABROAD', 'Agglika', 19),
(null, 'Roumania', 'Xwra pou menei o top G', 'ABROAD', 'Roumanika', null),
(null, 'Antarktikh', 'Foreste mpoufan', 'ABROAD', 'Agglika', null),
(null, 'Patra', 'H trith megalyterh polh ths elladws', 'LOCAL', 'Ellhnika', 3),
(null, 'Thessaloniki', 'H voreia polh', 'LOCAL', 'Ellhnika', 3),
(null, 'Volos', 'Exei pollous foithtes', 'LOCAL', 'Ellhnika', 3),
(null, 'Ios', 'Megalo Nhsi', 'LOCAL', 'Ellhnika', 3),
(null, 'Krhth', 'H gnwsth se olous krhth', 'LOCAL', 'Ellhnika', 3);


INSERT INTO travel_to VALUES
(1, 17,'2022-05-20', '2022-05-28'),
(2, 25,'2022-01-13', '2022-01-20'),
(3, 26,'2022-06-06', '2022-06-16'),
(4, 4,'2022-12-15', '2022-12-17'),
(4, 14,'2022-12-17', '2022-12-18'),
(4, 27,'2022-12-18', '2022-12-20'),
(5, 18,'2023-11-17', '2023-11-30'),
(6, 23,'2022-03-01', '2022-03-15'),
(7, 14,'2022-06-08', '2022-06-12'),
(8, 3,'2022-07-22', '2022-07-23'),
(8, 20,'2022-07-23', '2022-07-25'),
(9, 14,'2023-11-29', '2023-12-02'),
(10,4, '2022-04-27', '2022-05-02'),
(11,5, '2023-05-20', '2023-05-28'),
(12,7, '2023-01-13', '2023-01-20'),
(13,21, '2023-06-06', '2023-06-13'),
(13,2, '2023-06-13', '2023-06-16'),
(14,22, '2023-12-15', '2023-12-20'),
(15,27, '2021-11-17', '2021-11-30'),
(16,29, '2023-03-01', '2023-03-15'),
(17,23, '2023-06-08', '2023-06-12'),
(18,24, '2023-07-22', '2023-07-25'),
(19,30, '2021-11-29', '2021-12-02'),
(20,6, '2023-04-27', '2023-05-02'),
(21,4, '2021-05-20', '2021-05-28'),
(22,8, '2021-01-13', '2021-01-20'),
(23,2, '2021-06-06', '2021-06-13'),
(23,16, '2021-06-13', '2021-06-16'),
(24,2, '2021-12-15', '2021-12-20'),
(25,26, '2024-11-17', '2024-11-30'),
(26,4, '2021-03-01', '2021-03-15'),
(27,15, '2021-06-08', '2021-06-12'),
(28,16, '2021-07-22', '2021-07-25'),
(29,27, '2024-11-29', '2024-12-02'),
(30,4, '2021-04-27', '2021-05-02'),
(30,14, '2021-04-01', '2021-05-02');
```

D.ii 3.1.2_setupExtras.sql

```sql
USE travel_agency;

CREATE TABLE it (
it_AT CHAR(10) NOT NULL,
it_password VARCHAR(20) DEFAULT 'password' NOT NULL,
it_date_start DATETIME NOT NULL,
it_date_end DATETIME,

PRIMARY KEY (it_AT),

CONSTRAINT it_wrk_FK
FOREIGN KEY (it_AT)
REFERENCES worker(wrk_AT)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE offers (
off_code INT(11) NOT NULL AUTO_INCREMENT,
off_start_date DATETIME NOT NULL,
off_end_date DATETIME NOT NULL,
off_cost FLOAT(7,2) NOT NULL,
off_dst INT(11) NOT NULL,

PRIMARY KEY(off_code),

CONSTRAINT off_dst_FK
FOREIGN KEY (off_dst)
REFERENCES destination(dst_id)
ON UPDATE CASCADE ON DELETE CASCADE
);

CREATE TABLE reservation_offers (
rsof_code INT(11) NOT NULL AUTO_INCREMENT,
rsof_name VARCHAR(20) DEFAULT 'unknown' NOT NULL,
rsof_lname VARCHAR(20) DEFAULT 'unknown' NOT NULL,
rsof_off_code INT(11) NOT NULL,
rsof_downpay FLOAT(7,2) NOT NULL,

PRIMARY KEY(rsof_code),

CONSTRAINT rsof_off_FK
FOREIGN KEY (rsof_off_code)
REFERENCES offers(off_code)
ON UPDATE CASCADE ON DELETE CASCADE
);
```

```sql
INSERT INTO worker VALUES
('ODQKWLEIDK', 'John', 'root', 1000.00, 3),
('ODQKX1EIDK', 'Vasileios', 'Bardakis', 1000.00, 3),
('VDQKWLZIDK', 'Xaris', 'Xallas', 1000.00, 3),
('TRQKDLEIDK', 'Axilleas', 'Villiotis', 1000.00, 3);

INSERT INTO it VALUES
('ODQKWLEIDK', 'root','2015-03-25 00:00:00',null),
('ODQKX1EIDK', 'kwdikos','2016-03-23 00:00:00',null),
('VDQKWLZIDK', 'xalvas','2017-08-21 00:00:00',null),
('TRQKDLEIDK', 'eveonline','2013-07-16 00:00:00',null);


INSERT INTO offers VALUES
(NULL, '2023-01-01', '2023-12-31', 399.99, 20),
(NULL, '2023-01-01', '2023-12-31', 999.99, 4),
(NULL, '2023-01-01', '2023-12-31', 1299.99, 6);
```

D.iii IDandPasswords.java

```java
import java.sql.ResultSet;
import java.util.HashMap;

public class IDandPasswords {
    HashMap<String,String> loginInfo = new HashMap<String,String>();

    IDandPasswords(){
        try {
            ResultSet userIDs = Methods.getTableData("it_AT", "it", null);
            //Vriskw ta it_AT
            String query = "SELECT wrk_lname FROM worker WHERE ";
            while(userIDs.next()) {
                query = query + " wrk_AT = '" + userIDs.getString("it_AT") + "' OR";
                //Vriskw ta epwnunma twn IT staff
            }
            query = query.substring(0, query.length() - 3);
            ResultSet userNames = Methods.customQueryReturn(query);             //Pairnw
            ta epwnuma kai ta password twn IT staff kai ta vazw sta login info.
            ResultSet passwords = Methods.getTableData("it_password", "it", null);
            while (userNames.next() && passwords.next()) {
                loginInfo.put(userNames.getString("wrk_lname"),
                passwords.getString("it_password"));
            }
        }
        catch (Exception e){
            System.out.println("Error, please check your input");
            e.printStackTrace();
        }
    }
    public HashMap<String,String> getLoginInfo(){
        return loginInfo;
    }

}
```

D.iv ItOptionsFrame.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class ItOptionsFrame implements ActionListener {
    //Diafora variables pou xreiazomai
    String selectedTable;
    String[] tables = new String[16];
    String [] branches;
    public static int counter;

    //Images gia omorfo frame
    ImageIcon image = new ImageIcon("Main Files/Travel Agency/img.png"); //To icon tou
    application
    ImageIcon icon = new ImageIcon("Main Files/Travel Agency/companyicon.png"); //Icon
    ths etairias

    //Ftiaxnw frame,labels
    JFrame frame = new JFrame();
    JLabel welcomeLabel = new JLabel();
    JLabel nameLabel = new JLabel();
    JLabel actionsLabel = new JLabel();

    //Ftiaxnw buttons
    JButton insertDataButton = new JButton("Insert Data");
    JButton updateDataButton = new JButton("Update Data");
    JButton deleteDataButton = new JButton("Delete Data");
    JButton seeTripDataButton = new JButton("See Trip Information");
    JButton logButton = new JButton("Access logs");
    JButton branchInfoButton = new JButton("Store info");
    JButton addITButton = new JButton("Add IT");
    JButton clientOfferButton = new JButton("See client's offer");

    //Combo boxes
    static JComboBox<String> chooseTableBox;
    static JComboBox<String> chooseBranchBox;


    //Constructor
    ItOptionsFrame(String userID){

        String query = "SET SQL_SAFE_UPDATES = 0;";    //vgazoume safe mode gia na
        mporoume na svhsoume tous current users
        Methods.customQuery(query);
```

```
query = "SET GLOBAL sql_mode=(SELECT
REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));";     //Wste na mporoume na
kanoume nested select gia branch info button
Methods.customQuery(query);
query = "DELETE FROM currentUser;";
Methods.customQuery(query);
query = "INSERT INTO currentUser values ('"+userID+"');";     //Vazoume ws
current user to epwnumo tou xrhsth
Methods.customQuery(query);


ImageIcon background = new ImageIcon("Main Files/Travel Agency/background.png");
//To background tou frame
//ImageIcon background = new ImageIcon("background.png"); //To background tou
frame
frame.setContentPane(new JPanel(new BorderLayout()) {
    @Override public void paintComponent(Graphics g) {
        g.drawImage(background.getImage(), 0, 0, null);
    }
});


//Ftiaxnw combobox pou deixnei ola ta tables gia to insert button
//INIT CHOOSETABLEBOX (FOR TABLES)
int i=0;
String tablen;
try {
    ResultSet tableSet = Methods.getTableData("table_name",
    "information_schema.tables", "table_schema = 'travel_agency';");
    //Vriskei ola ta onomata twn tables
    while (tableSet.next()) {
        tablen = tableSet.getString("table_name");
        if(!tablen.equals("currentuser") && !tablen.equals("log")) {
            tables[i] = tablen;       //Ta eisagei ston pinaka tables
            i += 1;
        }
    }
}
catch (Exception e){
    System.out.println("Error, please check your input");
    e.printStackTrace();
}
chooseTableBox = new JComboBox<String>(tables);
chooseTableBox.addActionListener(this);
chooseTableBox.setBounds(950,300,250,50);
```

```
//INIT CHOOSESTOREBOX (FOR BRANCH)
i=0;
try {
    //Get row count
    ResultSet branchSet = Methods.getTableData("count(*)",
    "travel_agency.branch", null);        //Vriskei ola ta onomata twn tables
    branchSet.next();
    int rowCount = branchSet.getInt(1);
    branches = new String[rowCount]; //dynamically allocated

    branchSet = Methods.getTableData("br_code", "travel_agency.branch", null);
    //Vriskei ola ta onomata twn tables

    while (branchSet.next()) {
        branches[i] = branchSet.getString("br_code");        //Ta eisagei ston
        pinaka tables
        i++;
    }
}
catch (Exception e){
    System.out.println("Error, please check your input");
    e.printStackTrace();
}

chooseBranchBox = new JComboBox<String>(branches);
chooseBranchBox.addActionListener(this);
chooseBranchBox.setBounds(1100,650,100,50);

//Diafora labels gia pio omorfo frame
welcomeLabel.setText("Welcome IT user");
welcomeLabel.setBounds(650,70,210,35);
welcomeLabel.setFont(new Font(null, Font.BOLD,25));
welcomeLabel.setForeground(Color.WHITE);

nameLabel.setText(userID);
nameLabel.setBounds(855,70,200,35);
nameLabel.setFont(new Font(null, Font.BOLD,25));
nameLabel.setForeground(Color.RED);

actionsLabel.setText("What do you wanna do?");
actionsLabel.setBounds(32,150,400,25);
actionsLabel.setFont(new Font(null, Font.BOLD,25));
actionsLabel.setForeground(Color.WHITE);

//Vazoume ta buttons ekei pou theloume, tous vazoume action listener gia na
akoune energeies
insertDataButton.setBounds(10, 200, 150, 50);
```

```
        insertDataButton.setFocusable(false);
        insertDataButton.addActionListener(this);

        updateDataButton.setBounds(180, 200, 150, 50);
        updateDataButton.setFocusable(false);
        updateDataButton.addActionListener(this);

        deleteDataButton.setBounds(10, 260, 150, 50);
        deleteDataButton.setFocusable(false);
        deleteDataButton.addActionListener(this);

        logButton.setBounds(10,320,150,50);
        logButton.setFocusable(false);
        logButton.addActionListener(this);

        seeTripDataButton.setBounds(180, 260, 150, 50);
        seeTripDataButton.setFocusable(false);
        seeTripDataButton.addActionListener(this);

        addITButton.setBounds(180, 320, 150, 50);
        addITButton.setFocusable(false);
        addITButton.addActionListener(this);

        branchInfoButton.setBounds(10, 380, 150, 50);
        branchInfoButton.setFocusable(false);
        branchInfoButton.addActionListener(this);

        clientOfferButton.setBounds(180, 380, 150, 50);
        clientOfferButton.setFocusable(false);
        clientOfferButton.addActionListener(this);


        //Vazoume ola ta panels,buttons ktlp sto frame
        frame.add(welcomeLabel);
        frame.add(nameLabel);
        frame.add(actionsLabel);
        frame.add(deleteDataButton);
        frame.add(seeTripDataButton);
        frame.add(updateDataButton);
        frame.add(insertDataButton);
        frame.add(logButton);
        frame.add(addITButton);
        frame.add(branchInfoButton);
        frame.add(clientOfferButton);

        //Ftiaxnoume to frame
        frame.setTitle("Travel Agency"); //set title
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //kanoume main stop otan
        kanoume exit
        frame.setLayout(null);
        frame.setResizable(false); //den afhnoume resize
        frame.setSize(1280, 960);
        frame.setIconImage(image.getImage());


        new RedirectAppOutputStream().guiConsoleTest(frame);


        frame.setVisible(true);
    }


    public static void removeAll (int counter, JFrame frame) {

        // remove previous dropdown fields
        for (int i = 1; i < Methods.fields.size(); i++) {
            frame.remove(Methods.fields.get(i));
            // frame.remove(Methods.fields[i]);
        }

        for (int i=0; i < Methods.chooseKey.size(); i++) {
            frame.remove(Methods.chooseKey.get(i));
        }

        Methods.columnNameStatic.clear();
        frame.remove(Methods.submitButton);
        frame.remove(Methods.deleteButton);
        frame.remove(Methods.scrollPane);
        frame.remove(Methods.submitITButton);
        frame.remove(Methods.scrollPane);
        frame.remove(Methods.scrollPaneSalary);
        frame.remove(Methods.salaryLabel);
        frame.remove(Methods.columnLabel);

        frame.remove(chooseTableBox);
        frame.remove(chooseBranchBox);

        if (counter == 3) {
            for(int i=0;i<7;i++)
                frame.remove(Methods.chooseDateBox[i]);

            frame.remove(Methods.submitDateButton);
            frame.remove(Methods.datetimeLabel);
            frame.remove(Methods.questionLabel);
        }
        if(counter==8){
            frame.remove(Methods.chooselName);
        }

        SwingUtilities.updateComponentTreeUI(frame); //for good measure
    }


    @Override

    public void actionPerformed(ActionEvent e) {     //Koumpi Insert Data
        if(e.getSource()==insertDataButton) {
            removeAll(counter,frame);
            frame.add(chooseTableBox);
            frame.setVisible(true);
            SwingUtilities.updateComponentTreeUI(frame);
            counter=0;

            //Show fields selected at previous usage or index 0 (admin)
            int i = chooseTableBox.getSelectedIndex();
            selectedTable = tables[i];
            Methods.insertTableData(selectedTable,frame);

        }

        else if(e.getSource()==chooseTableBox) {    //Table twn insert data kai update
        data buttons
            removeAll(counter,frame);
            frame.add(chooseTableBox);

            int i = chooseTableBox.getSelectedIndex();
            selectedTable = tables[i];

            switch (counter) {
                case 0:
                    Methods.insertTableData(selectedTable,frame);
                    break;

                case 1:
                    Methods.updateTableButton(selectedTable,frame,false);
                    break;

                case 2:
                    Methods.updateTableButton(selectedTable,frame,true);
                    break;
            }
        }



        else if(e.getSource()==updateDataButton) {   //Koumpi Update Data
            removeAll(counter,frame);
            frame.add(chooseTableBox);
            frame.setVisible(true);
            SwingUtilities.updateComponentTreeUI(frame);
            counter=1;

            //Show table selected at previous usage or index 0 (admin)
            int i = chooseTableBox.getSelectedIndex();
            selectedTable = tables[i];
            Methods.updateTableButton(selectedTable,frame, false);
        }

        else if(e.getSource()==deleteDataButton) {   //Koumpi Update Data
            removeAll(counter,frame);
            frame.add(chooseTableBox);
            frame.setVisible(true);
            SwingUtilities.updateComponentTreeUI(frame);
            counter=2;

            //Show table selected at previous usage or index 0 (admin)
            int i = chooseTableBox.getSelectedIndex();
            selectedTable = tables[i];
            Methods.updateTableButton(selectedTable,frame, true);
        }

        else if(e.getSource()==seeTripDataButton){
            removeAll(counter,frame);


            try {
                Methods.tripInfoButton(frame);
            } catch (SQLException throwables) {
                System.out.println("Error, please check your input");
                throwables.printStackTrace();
            };
            SwingUtilities.updateComponentTreeUI(frame);
            counter=3;
        }

        else if(e.getSource()==logButton) { //Koumpi logs
            removeAll(counter,frame);
            Methods.logTableButton(frame);
            SwingUtilities.updateComponentTreeUI(frame);
            counter=4;
        }

        else if(e.getSource()==addITButton){
            removeAll(counter,frame);
            Methods.addITButton(frame);
            SwingUtilities.updateComponentTreeUI(frame);
            counter=5;
        }

        else if (e.getSource()==branchInfoButton) {
            removeAll(counter,frame);
            frame.add(chooseBranchBox);
            Methods.branchInfoButton(frame);
            SwingUtilities.updateComponentTreeUI(frame);
            counter=6;

            // pre render the latest selected branch for salary overview when clicking
            on branch info button
            // defaults to index 0 which is the first branch
            frame.remove(Methods.scrollPaneSalary);
            int i = chooseBranchBox.getSelectedIndex(); //Returns the item that was
            selected
            selectedTable = branches[i];
            Methods.branchWorkers(frame, selectedTable);
            counter=7;
        }

        else if(e.getSource()==chooseBranchBox) {    //Table twn insert data kai update
        data buttons
            //Right after branchInfo, so all previous panes are cleared by branchInfo.
            frame.remove(Methods.scrollPaneSalary);

            int i = chooseBranchBox.getSelectedIndex(); //Returns the item that was
            selected
            selectedTable = branches[i];

            Methods.branchWorkers(frame, selectedTable);
            counter=7;
        }

        else if(e.getSource()==clientOfferButton){
            removeAll(counter,frame);
            Methods.clientOffersButton(frame);

            counter=8;
        }
    }
}
```

D.v LoginFrame.java

```java
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.HashMap;
import javax.swing.*;

public class LoginFrame extends JFrame implements ActionListener {
    JButton loginButton = new JButton("Login");
    JButton exitButton = new JButton("Exit");
    JTextField userIDField = new JTextField();
    JPasswordField userPasswordField = new JPasswordField();
    JLabel userIDLabel = new JLabel("User ID:");
    JLabel userPasswordLabel = new JLabel("Password:");
    JLabel messageLabel = new JLabel();
    ImageIcon image = new ImageIcon("Main Files/Travel Agency/img.png"); //To icon tou
    application
    ImageIcon icon = new ImageIcon("Main Files/Travel Agency/companyicon.png"); //Icon
    tou login screen
    JLabel label1 = new JLabel("<html>Welcome to <br/> Viljot Halbar<html>");
    JPanel panel1 = new JPanel();
    HashMap<String,String> loginInfo = new HashMap<String,String>(); //Ftiaxnoume to
    hashmap loginInfo wste na einai globally available gia na xrhsimopoihthei apo
    methodous

    LoginFrame(HashMap<String,String> loginInfoOriginal) {
        loginInfo = loginInfoOriginal;


        userIDLabel.setBounds(410,430,80,25);
        userIDLabel.setFont(new Font(null,Font.BOLD,15));
        userPasswordLabel.setBounds(410,475,80,25);
        userPasswordLabel.setFont(new Font(null,Font.BOLD,15));

        messageLabel.setBounds(450,350,500,35);
        messageLabel.setFont(new Font(null,Font.BOLD,21));

        userIDField.setBounds(490,430,200,25);
        userPasswordField.setBounds(490,475,200,25);


        //Ftiaxnoume omorfa to window
        this.setTitle("Travel Agency"); //set title
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); //kanoume main stop otan
        kanoume exit
        this.setLayout(null);
        this.setResizable(false); //den afhnoume resize
        this.setSize(1280, 960);
        this.setIconImage(image.getImage());
        this.getContentPane().setBackground(Color.cyan);  //alazoume xrwma sto
        background

        panel1.setBackground(Color.cyan);
        panel1.setBounds(0, 0, 620, 300);


        label1.setHorizontalTextPosition(JLabel.RIGHT);
        label1.setVerticalTextPosition(JLabel.CENTER);
        label1.setForeground(Color.BLUE);
        label1.setFont(new Font("Mv Boli", Font.BOLD, 38));
        label1.setIcon(icon);
        panel1.add(label1);




        loginButton.setBounds(430, 515, 100, 50);
        loginButton.setFocusable(false);
        loginButton.addActionListener(this);
        exitButton.setBounds(580,515,100,50);
        exitButton.setFocusable(false);
        exitButton.addActionListener(this);




        this.add(loginButton);
        this.add(userIDLabel);
        this.add(userIDField);
        this.add(userPasswordField);
        this.add(messageLabel);
        this.add(userPasswordLabel);
        this.add(exitButton);
        this.add(panel1);

        this.getRootPane().setDefaultButton(loginButton); // enter presses login button

        this.setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==exitButton) {
            System.exit(0);
        }

        if(e.getSource()==loginButton){
            String userID = userIDField.getText().trim();
            String password = String.valueOf(userPasswordField.getPassword());

            if(loginInfo.containsKey(userID)) {
                if (loginInfo.get(userID).equals(password)) {
                    messageLabel.setForeground(Color.magenta);
                    messageLabel.setText("Login Successful");
                    this.dispose();        //diwxnoume to palio window


                    new ItOptionsFrame(userID);

                } else {
                    messageLabel.setForeground(Color.red);
                    messageLabel.setText("You entered the wrong password.");
                }
            }
            else{
                messageLabel.setForeground(Color.red);
                messageLabel.setText("User ID does not exist in database.");
            }
        }
    }
}
```

D.vi Main.java

```
public class Main {
    public static void main(String[] args){

        IDandPasswords idandPasswords = new IDandPasswords();
        new LoginFrame(idandPasswords.getLoginInfo());
    }
}
```

D.vii Methods.java

```java
import org.w3c.dom.events.Event;
import org.w3c.dom.events.EventListener;

import javax.swing.*;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;

//for list of foreign keys
import java.util.*;

public class Methods implements ActionListener, ListSelectionListener, EventListener {

    public static JLabel datetimeLabel;
    public static JLabel questionLabel;
    public static int counterStatic = 0;
    static boolean first=true;
    private static int columnCounterStatic = 0;
    public static int selColumnStatic;
    public static int selRowStatic;
    static JComboBox[] chooseDateBox = new JComboBox[7];    //Gia to trip info button
    static JComboBox<String> chooselName;
    private static String primaryName;
    private static String selectedlName;
    private static String tableStatic;
    //private static String[] columnNameStatic = new String[10];
    public static ArrayList<String> columnNameStatic = new ArrayList<String>();
    private static String[] columnTypeStatic = new String[10];
    public static JLabel columnLabel = new JLabel();
    private static int[] dropdownArray = new int[3];
    //https://stackoverflow.com/questions/16213836/java-swing-jtextfield-set-placeholder
    public static ArrayList<JTextField> fields = new ArrayList<JTextField>() {
        {
            add(new JTextField("First"));
        }
    };

    static JButton submitButton = new JButton("Submit");
    static JButton submitITButton = new JButton("Submit");
    static JButton submitDateButton = new JButton("Submit Date");
    static JButton deleteButton = new JButton("Delete");
```

```
private static String valueStatic[] = new String[10];
private static JFrame frameStatic;
static boolean changed=false;
static boolean changedFeb=false;

static JScrollPane scrollPane = new JScrollPane();
static JScrollPane scrollPaneSalary = new JScrollPane();

static JLabel salaryLabel = new JLabel();
static public ArrayList<JComboBox<String>> chooseKey = new
ArrayList<JComboBox<String>>();

//https://docs.oracle.com/javase/7/docs/api/java/sql/DatabaseMetaData.html
//https://stackoverflow.com/questions/18716651/get-all-foreign-keys-using-jdbc
static public ArrayList<String> getFKeyData(String tableName, int i) throws
SQLException {
    ArrayList<String> fkTableData = new ArrayList<String>();
    try {
        Connection connection =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/travel_agency",
        "root","root");
        DatabaseMetaData dm = connection.getMetaData();
        ResultSet rs = dm.getImportedKeys(null, null, tableName);
        while (rs.next()) {
            fkTableData.add(rs.getString(i));
        }
    }
    catch (Exception e) {
        System.out.println("Error, please check your input.");
        e.printStackTrace();
    }
    return fkTableData;
}


//Custom query. Returns rows as ResultSet
public static ResultSet customQueryReturn(String query){     //Pairnei ws orisma ena
String query, trexei thn entolh sthn mysql kai epistrefei to resultSet
    ResultSet resultSet = null;
    try {
        Connection connection =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/travel_agency",
        "root","root");  //Kanei connection me to database
        Statement statement =
        connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_READ_ONLY);  //Scroll Insensitive gia na mporoume na
        ksanagurisoume sthn arxh tou result set
```

```
            resultSet = statement.executeQuery(query);     //Stelnoume to statement sto
            database mesw tou jdbc
        }
        catch (Exception e) {
            System.out.println("Error, please check your input.");
            e.printStackTrace();
        }
        return resultSet;
    }

    public static void customQuery(String query){     //To idio me thn proigoumenh apla
    den epistrefei resultSet (executeUpdate anti gia executeQuery)
        boolean isException = false;
        try {
            Connection connection =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/travel_agency",
            "root", "root");
            Statement statement = connection.createStatement();
            statement.executeUpdate(query);
        }
        catch (Exception e) {
            System.out.println("Error, please check your input.");
            isException = true;
            e.printStackTrace();
        }

        if (!isException)
            System.out.println("Database has been updated!");

    }

    //SELECT field FROM table [WHERE where]. Returns query's rows as ResultSet
    public static ResultSet getTableData(String field, String table, String where) {
    //Ama to where einai null tote trexei xwris where
        try {
            Connection connection =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/travel_agency",
            "root", "root");
            Statement statement = connection.createStatement();
            String query;
            if (where == null)
                //TODO: Wrap all queries with '' so spaces are allowed
                query = "SELECT " + field + " FROM " + table;
            else
                query = "SELECT " + field + " FROM " + table + " WHERE " + where;
            ResultSet resultSet = statement.executeQuery(query);

            return resultSet;

        } catch (Exception e) {
            System.out.println("Error, please check your input.");
            e.printStackTrace();
        }
        return null;
    }

    //UPDATE table
    public static void updateTableData(String table, String columnName, String
    columnType,String newValue, String condition) {    //Kanei update to mysql database
        String query;
        try {
            Connection connection =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/travel_agency",
            "root","root");
            Statement statement = connection.createStatement();

            query = "UPDATE " + table + " SET " + columnName + " = '" + newValue + "'
            WHERE" + condition;

            statement.executeUpdate(query);
            System.out.println("The table changed");
        } catch (SQLException throwables) {

            //https://docs.oracle.com/javase/6/docs/api/java/sql/SQLException.html#getSQLState()
            System.out.println(throwables.getMessage());
            System.out.println("Error, please check your input.");
            throwables.printStackTrace();
        }
    }


    public static void insertTableData(String table, JFrame frame) {         //Vazoume
    to table kai meta vazoume values

        chooseKey.removeAll(chooseKey);     //Vgazoume oti uphrxe apo proigoumeno
        treksimo ths insertTableData
        chooseKey.trimToSize();             //Prosarmozoume to megethos
        dropdownArray = new int[] {0, 0, 0, 0};    //Arxikopoihsh dropdownArray to opoio
        krataei poia fields mporoun na ginoun me dropdown
        tableStatic = table;                   //To xrhsimopoiw sto button
        submitButton = new JButton("Submit");
        submitButton.setBounds(590, 750, 100, 50);
        submitButton.setFocusable(true);
        frame.add(submitButton);
        frame.setVisible(true);
        submitButton.addActionListener(new Methods());
        int count=0;
        String names="";

        try {
            //List that contains all special cases.
            //If column is detected to be a foreign key (found in arraylist containing
            the foreignkeys) then dropdown. Else normal field

            //https://stackoverflow.com/questions/16213836/java-swing-jtextfield-set-placeholder

            //https://stackoverflow.com/questions/33799800/java-local-variable-mi-defined-in-an-enclo
            ResultSet resultSet = getTableData("*",table,null);         //Pairnw
            plhrofories gia to table
            ResultSetMetaData rsmd = resultSet.getMetaData();

            counterStatic = rsmd.getColumnCount();         //Vriskoume posa columns
            exoume
            frame.setVisible(true);

            //Grabs referenced table name
            ArrayList<String> referenced_tables = getFKeyData(table,3);
            //Grabs key column name from referenced table
            ArrayList<String> referenced_table_column = getFKeyData(table,4);
            //Grabs key column name from current table
            ArrayList<String> current_table_column = getFKeyData(table,8);
            //int keyRowCount = referenced_tables.size();

            //Fields where user types
            int i = 0;
            for (i = 1; i <= counterStatic; i++) {
                fields.add(i,new JTextField(rsmd.getColumnName(i)));       //Vazoume ta
                fields sta opoia tha grafei o xrhsths
                columnNameStatic.add(rsmd.getColumnName(i));
                fields.get(i).setBounds(490, 430 + i *32, 200, 25);
                //workaround
                final int inner_i = i;
                String columnName = rsmd.getColumnName(i);


                int indexOfKey = current_table_column.indexOf(columnName);   //Pairnoume
                to index tou kleidiou

                //column is foreign key
                if (indexOfKey != -1) {      //An einai kleidi, tote kanw dropdown lista
                me auto (JComboBox chooseKey)
                    chooseKey.add(new JComboBox<String>());

                    dropdownArray[count] = i;

                    //Ftiaxnw ta onomata twn jcombobox me ena JLabel
                    count++;
                    if(count==1) {                              //Gia to prwto jcombobox
                    ftiaxnw ta bounds kai to alignment, arxikopoiw to string names
                        names=columnName;
                        columnLabel.setText(names);
                        columnLabel.setVerticalAlignment(SwingConstants.TOP);
                        columnLabel.setBounds(410, 433 + i * 32, 200, 300);
                    }
                    else{
                    //Tis epomenes fores apla prosthetw sto names newlines kai to
                    columnName
                        names ="<html><body>" + names + "<br/> <br/>" +columnName +
                        "<html>";
                        columnLabel.setText(names);
                    }
                    //these are the values that can be selected from combo box
                    resultSet = getTableData(referenced_table_column.get(indexOfKey),
                    referenced_tables.get(indexOfKey), null);  //Pairnoume tis times pou
                    mporei na parei to kleidi

                    while (resultSet.next()) {

                        chooseKey.get(chooseKey.size()-1).addItem(resultSet.getString(1));
                        //Prosthetoume ta item sto jComboBox
                    }



                    chooseKey.get(chooseKey.size()-1).setBounds(490, 430 + i * 32, 200,
                    25);
                    frame.add(chooseKey.get(chooseKey.size()-1));
                    frame.add(columnLabel);
                    frame.setVisible(true);
                } else {
                    fields.get(i).addFocusListener(new FocusListener() {
                        @Override
                        public void focusGained(FocusEvent e) {
                            if (fields.get(inner_i).getText().equals(columnName)) {
                                fields.get(inner_i).setText("");
                                //fields[inner_i].setForeground(Color.BLACK);
                            }
                        }
                        @Override
                        public void focusLost(FocusEvent e) {
                            if (fields.get(inner_i).getText().isEmpty()) {
                                fields.get(inner_i).setText(columnName);
                            }
                        }
                    });

                    frame.add(fields.get(i));
                    // frame.add(fields[i]);
                    frame.setVisible(true);
                    columnTypeStatic[i] = rsmd.getColumnTypeName(i);
                }


            submitButton.setBounds(590, 430 + i*32 + 50, 100, 50);
            }

        } catch (Exception e) {
            System.out.println("Error, please check your input.");
            e.printStackTrace();
        }
    }


    //When "update table" or "delete row" is pressed
    public static void updateTableButton(String table, JFrame frame, boolean delete) {
    //Xrhsimopoieitai sto update data koumpi
        int i = 0;
        tableStatic=table;
        String[] columnNames;
        String[][] rowInfo;

        //delete row
        if(delete){
            tableStatic = table;                   //To xrhsimopoiw sto button
            deleteButton = new JButton("Delete Row");      //Ftiaxnoume neo button
            deleteButton.setBounds(807, 700, 100, 50);
            deleteButton.setFocusable(true);
            frame.add(deleteButton);
            frame.setVisible(true);
            SwingUtilities.updateComponentTreeUI(frame);
            deleteButton.addActionListener(new Methods());
        }
        try {
            //Ftiaxnw connection me to database
            Connection connection =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/travel_agency",
            "root","root"); //Den kalesa apla thn getTableData giati xreiazomai to
            connection gia na vrw to unique key
            Statement statement = connection.createStatement();

            //Vriskw posa rows exw
            String query = "SELECT COUNT(*) FROM " + table;
            ResultSet resultSet = statement.executeQuery(query);
            resultSet.next();
            int rowCount = resultSet.getInt(1);

            //Vriskw to primary key
            DatabaseMetaData dbmd = connection.getMetaData();
            try (ResultSet rs = dbmd.getIndexInfo(null, null, table, true, true)) {
                rs.next();
                primaryName = rs.getString("COLUMN_NAME");
            }



            //Pairnoume ton pinaka se afksousa seira tou primary key (to kanw gia na
            mporw na vrw meta poia timh tha allaksw sto database)
            query = "SELECT * FROM " + table + " ORDER BY " + primaryName + " ASC";
            resultSet = statement.executeQuery(query);
            ResultSetMetaData rsmd = resultSet.getMetaData();




            columnCounterStatic = rsmd.getColumnCount();         //Vriskoume posa
            columns exoume
            rowInfo = new String[rowCount][columnCounterStatic];

            columnNames = new String[columnCounterStatic];
            for(i=0;i<columnCounterStatic;i++){
                columnNames[i] = rsmd.getColumnName(i+1);     //Vazoume ta onomata twn
                columns ston pinaka columnNames
                columnNameStatic.add(columnNames[i]);
                columnTypeStatic[i] = rsmd.getColumnTypeName(i+1);
            }
            for(i=0;i<rowCount;i++){
                resultSet.next();
                for(int j=0;j<columnCounterStatic;j++){
                    rowInfo[i][j]=resultSet.getString(columnNames[j]);       //Vazoume
                    ta data sthn lista rowInfo pou tha xrhsimopoihsoume gia to JTable
                }
            }

            //Ftiaxnoume to table kai to pane kai to vazoume sto frame, yo
            JTable jTable = new JTable(rowInfo,columnNames);
            jTable.setPreferredScrollableViewportSize(new Dimension(50,63));
            jTable.setFillsViewportHeight(true);
            if(delete){     //Periptwsh pou thelw na diagrapsw
                jTable.setCellSelectionEnabled(false);         //Den afhnw ta kelia na
                einai editable
                jTable.setRowSelectionAllowed(true);
            }
            scrollPane = new JScrollPane(jTable);
            scrollPane.setBounds(400, 250, 507, 300);
            SwingUtilities.updateComponentTreeUI(frame);    //Kanoume refresh gia na
            fugei oti exei ksemeinei
            frame.add(scrollPane);


            //Action listener gia to jTable, pairnei to row kai to column otan epilegw
            kapoio kelh
            jTable.getSelectionModel().addListSelectionListener(e -> {
                int selColumn = jTable.getSelectedColumn();
                int selRow = jTable.getSelectedRow();
                selColumnStatic=selColumn;
                selRowStatic=selRow;
                int j;
                for(j=0;j<columnCounterStatic;j++) {
                    valueStatic[j] = (String) jTable.getValueAt(selRow, j);
                }
            });

            //Se periptwsh pou exw pathsei to update button
            if(!delete) {
                String[] finalColumnNames1 = columnNames;         //Den me afhne xwris an
                to kanw final giati einai lamda expression
                jTable.getModel().addTableModelListener(e -> {         //Energopoieitai
                otan teleiwsw na kanw tropopoihsh ena kelh
                    String condition=" ";      //Tha einai o elegxos gia thn WHERE sto
                    query ths mysql
                    int j;
                    for(j=0;j<columnCounterStatic;j++){
                        if(valueStatic[j]!=null) {
                            condition = condition + columnNames[j] + "='" +
                            valueStatic[j] + "'";
                            if (columnCounterStatic - 1 > j && valueStatic[j+1]!=null)
                            //An den eimai sto teleutaio column vazw AND
                                condition = condition + " AND ";
                        }
                    }

                    if (valueStatic[selColumnStatic] != null &&
                    jTable.getValueAt(selRowStatic, selColumnStatic) != null) {
                    //Periptwsh pou den exw null value prin h meta thn tropopoihsh,
                        if (!valueStatic[selColumnStatic].equals((String)
                        jTable.getValueAt(selRowStatic, selColumnStatic))) {
                        //Elegxw an allakse h timh pou mphke sto kelh einai idia me auth
                        pou htan hdh mesa
                            try {
                                updateTableData(table,
                                finalColumnNames1[selColumnStatic],
                                rsmd.getColumnTypeName(selColumnStatic+1), (String)
                                jTable.getValueAt(selRowStatic, selColumnStatic),
                                condition);
                                valueStatic[selColumnStatic]=(String)
                                jTable.getValueAt(selRowStatic, selColumnStatic);
                                //Enhmerwnw ton pinaka me ta values tou table
                            } catch (SQLException throwables) {
                                System.out.println(throwables.getMessage());
                                System.out.println("Error, please check your input.");
                                throwables.printStackTrace();
                            }
                        }
                    } else                //Periptwsh pou eixa null value, tote den kanw
                    check an h proigoumenh timh einai idia me prin
                        try {
                            updateTableData(table, finalColumnNames1[selColumnStatic],
                            rsmd.getColumnTypeName(selColumnStatic+1), (String)
                            jTable.getValueAt(selRowStatic, selColumnStatic),
                            condition);
                            valueStatic[selColumnStatic]=(String)
                            jTable.getValueAt(selRowStatic, selColumnStatic);
                        } catch (SQLException throwables) {
                            System.out.println(throwables.getMessage());
                            System.out.println("Error, please check your input.");
                            throwables.printStackTrace();
                        }
                });
            }



        } catch (SQLException throwables) {
            System.out.println(throwables.getMessage());
            System.out.println("Error, please check your input.");
            throwables.printStackTrace();
        }


    }

    public static void logTableButton(JFrame frame) {
        String[] columnNames;
        String[][] rowInfo;

        try {
            ResultSet resultSet = getTableData("COUNT(*)", "log", null);    //Vriskw
            posa rows exw sto log
            resultSet.next(); //move cursor to start
            int rowCount = resultSet.getInt(1);

            resultSet = getTableData("lg_tableName 'Table', lg_actionName Action,
            lg_actionNum ID, lg_wrk_lname Lastname, lg_timeofchange Datetime", "log",
            null);
            //resultSet.next(); //Dont do next before getmetadata! (stored before first
            index)
            //https://bugs.mysql.com/bug.php?id=40256

            //https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-connp-props-jdbc-compliance.ht
            ResultSetMetaData rsmd = resultSet.getMetaData();
            columnCounterStatic = rsmd.getColumnCount();
            rowInfo = new String[rowCount][columnCounterStatic];

            //Store column names
            columnNames = new String[columnCounterStatic];    //Vriskw ta onomata twn
            columns
            for(int i=0;i<columnCounterStatic;i++){
                columnNames[i] = rsmd.getColumnLabel(i+1);
            }

            //Fill every row first, then column
            for(int i=0;i<rowCount;i++) {               //Arxikopoiw ton pinaka rowInfo
            pou exei ta dedomena twn rows
                resultSet.next();
                for(int j=0;j<columnCounterStatic;j++){
                    rowInfo[i][j]=resultSet.getString(columnNames[j]);
                }
            }

            //Ftiaxnoume to table kai to pane kai to vazoume sto frame, yo
            JTable jTable = new JTable(rowInfo,columnNames) {
                @Override
                public boolean isCellEditable(int row, int column) {        //Den thelw
                na einai editable ta kelia
                    return false;
                }
            };
            jTable.setPreferredScrollableViewportSize(new Dimension(450,63));
            jTable.setFillsViewportHeight(true);
            jTable.setCellSelectionEnabled(false);
            jTable.setRowSelectionAllowed(false);

            scrollPane = new JScrollPane(jTable);
            scrollPane.setBounds(400, 250, 607, 300);
            SwingUtilities.updateComponentTreeUI(frame);
            frame.add(scrollPane);
        }

        catch (SQLException e) {
            System.out.println(e.getMessage());
            System.out.println("Error, please check your input.");
            e.printStackTrace();
        }
    }

    public static void addITButton(JFrame frame) {
        submitITButton = new JButton("Submit");      //Ftiaxnoume neo button
        submitITButton.setBounds(590, 700, 100, 50);
        submitITButton.setFocusable(true);
        frame.add(submitITButton);
        frame.setVisible(true);
        submitITButton.addActionListener(new Methods());

        try {
            ResultSet resultSet = getTableData("*","it",null);         //Pairnw
            plhrofories gia to table
            ResultSetMetaData rsmd = resultSet.getMetaData();
            counterStatic = 8; //4 apo it kai 4 apo worker!
            frame.setVisible(true);

            for (int i = 1; i < 5; i++) {                //Ftiaxnoume fields gia na
            eisagei o xrhsths ta dedomena
                fields.add(i, new JTextField(rsmd.getColumnName(i)));
                columnNameStatic.add(rsmd.getColumnName(i));
                fields.get(i).setBounds(490, 400 + i * 32, 200, 25);

                //workaround
                final int inner_i = i;
                String columnName = rsmd.getColumnName(i);

                fields.get(i).addFocusListener(new FocusListener() {
                    @Override
                    public void focusGained(FocusEvent e) {
                        if (fields.get(inner_i).getText().equals(columnName)) {
                            fields.get(inner_i).setText("");
                        }

                    }
                    @Override
                    public void focusLost(FocusEvent e) {
                        if (fields.get(inner_i).getText().isEmpty()) {
                            fields.get(inner_i).setText(columnName);
                        }
                    }
                    });

                frame.add(fields.get(i));       //Vazoume ta fields sto frame
                frame.setVisible(true);
                columnTypeStatic[i] = rsmd.getColumnTypeName(i);

            }

            resultSet = getTableData("*","worker",null);         //Pairnw plhrofories
            gia to table
            rsmd = resultSet.getMetaData();
            frame.setVisible(true);

            for (int i = 1; i < 4; i++) {                //Ftiaxnoume fields gia na
            eisagei o xrhsths ta dedomena
                fields.add(i+4, new JTextField(rsmd.getColumnName(i+1)));
                columnNameStatic.add(rsmd.getColumnName(i+1));
                fields.get(i+4).setBounds(490, 400 + (i+4) * 32, 200, 25);

                //workaround
                final int inner_i = i+4;
                String columnName = rsmd.getColumnName(i+1);

                fields.get(i+4).addFocusListener(new FocusListener() {
                    @Override
                    public void focusGained(FocusEvent e) {
                        if (fields.get(inner_i).getText().equals(columnName)) {
                            fields.get(inner_i).setText("");
                        }
                    }
                    @Override
                    public void focusLost(FocusEvent e) {
                        if (fields.get(inner_i).getText().isEmpty()) {
                            fields.get(inner_i).setText(columnName);
                        }
                    }
                });

                frame.add(fields.get(i+4));

                frame.setVisible(true);
                columnTypeStatic[i+4] = rsmd.getColumnTypeName(i+1);

            }

            //wrk_br_code
            chooseKey.removeAll(chooseKey);     //Vgazoume oti uphrxe apo proigoumeno
            treksimo ths insertTableData
            chooseKey.trimToSize();             //Prosarmozoume to megethos
            chooseKey.add(new JComboBox<String>());     //Ftiaxnoume dropdown lista gia
            to branch code
            resultSet = getTableData("br_code", "branch ORDER BY br_code ASC", null);

            while (resultSet.next()) {
                chooseKey.get(chooseKey.size()-1).addItem(resultSet.getString(1));
                //Vazoume ola ta branches sto JComboBox chooseKey
            }

            // i =4
            chooseKey.get(chooseKey.size()-1).setBounds(490, 400 + 8 * 32, 200, 25);
            frame.add(chooseKey.get(chooseKey.size()-1));
            frame.setVisible(true);
        }

        catch (SQLException e) {
            System.out.println(e.getMessage());
            System.out.println("Error, please check your input.");
            e.printStackTrace();
        }
    }

    public static void branchInfoButton(JFrame frame) {
        String [] columnNames;
        String[][] rowInfo;
```

```
        SwingUtilities.updateComponentTreeUI(frame);    //Kanoume refresh gia na fugei
        oti exei ksemeinei

        try {


            ResultSet resultSet = getTableData("COUNT(*)",    //Pairnoume ton arithmo
            rows twn apotelesmatwn ths stored procedure
                    "(SELECT Branch, Street, City, Manager, SUM(Reservations) as
                    Reservations, SUM(Profit) as Profit FROM (SELECT b.br_code Branch,
                    CONCAT(b.br_street, ' ', b.br_num) Street, b.br_city City,
                    CONCAT(w.wrk_name, ' ', w.wrk_lname) Manager, "
                            + "t.tr_id Trip, COUNT(r.res_tr_id) Reservations,
                            tr_cost*COUNT(r.res_tr_id) Profit FROM branch b LEFT JOIN
                            manages m ON b.br_code = m.mng_br_code LEFT JOIN worker w ON
                            m.mng_adm_AT = w.wrk_AT LEFT JOIN trip t ON t.tr_br_code =
                            b.br_code LEFT JOIN reservation r ON t.tr_id = r.res_tr_id "
                            + "GROUP BY tr_id) AS T GROUP BY branch) AS T2", null);
            resultSet.next();
            int rowCount = resultSet.getInt(1);



            //https://stackoverflow.com/questions/6367737/resultset-exception-set-type-is-type-forwar
            //resultSet.beforeFirst();

            resultSet = getTableData("Branch, Street, City, Manager, SUM(Reservations)
            as Reservations, SUM(Profit) as Profit",
                    "(SELECT b.br_code Branch, CONCAT(b.br_street, ' ', b.br_num)
                    Street, b.br_city City, CONCAT(w.wrk_name, ' ', w.wrk_lname)
                    Manager, t.tr_id Trip, COUNT(r.res_tr_id) Reservations,
                    tr_cost*COUNT(r.res_tr_id) Profit FROM branch b LEFT JOIN manages m
                    ON b.br_code = m.mng_br_code LEFT JOIN worker w ON m.mng_adm_AT =
                    w.wrk_AT LEFT JOIN trip t ON t.tr_br_code = b.br_code LEFT JOIN
                    reservation r ON t.tr_id = r.res_tr_id  GROUP BY tr_id) AS T GROUP
                    BY branch", null);

            ResultSetMetaData rsmd = resultSet.getMetaData();
            columnCounterStatic = rsmd.getColumnCount();  //vriskoume posa columns
            exoume


            rowInfo = new String[rowCount][columnCounterStatic];

            //https://bugs.mysql.com/bug.php?id=40256

            //https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-connp-props-jdbc-compliance.ht
        columnNames = new String[columnCounterStatic];
        for (int i=0; i<columnCounterStatic; i++) {     //Arxikopoioume thn
        columnNames
          columnNames[i] = rsmd.getColumnLabel(i+1);
        }

        for(int i=0;i<rowCount;i++) {
          resultSet.next();
          for(int j=0;j<columnCounterStatic;j++){
            rowInfo[i][j]=resultSet.getString(columnNames[j]);       //Vazoume
            ta data sthn lista rowInfo pou tha xrhsimopoihsoume gia to JTable

            // EDGE CASE
            if ((columnNames[j].equals("Reservations") ||
            columnNames[j].equals("Profit")))
              if (rowInfo[i][j] == null)
                rowInfo[i][j] = "0";

          }
        }

        //Ftiaxnoume to table kai to pane kai to vazoume sto frame, yo
        JTable jTable = new JTable(rowInfo,columnNames) {
          @Override
          public boolean isCellEditable(int row, int column) {
            return false;
          }
        };

        jTable.setPreferredScrollableViewportSize(new Dimension(450,63));
        jTable.setFillsViewportHeight(true);
        jTable.setCellSelectionEnabled(false);
        jTable.setRowSelectionAllowed(false);

        scrollPane = new JScrollPane(jTable);
        scrollPane.setBounds(400, 250, 607, 300);
        SwingUtilities.updateComponentTreeUI(frame);    //Kanoume refresh gia na
        fugei oti exei ksemeinei
        frame.add(scrollPane);
      } catch (SQLException e) {
        System.out.println(e.getMessage());
        System.out.println("Error, please check your input.");
        e.printStackTrace();
      }
    }

    public static void branchWorkers(JFrame frame, String branch) {
        String[] columnNames;
        String[][] rowInfo;
        int columnCount;
        int totalSalaries = 0;


        SwingUtilities.updateComponentTreeUI(frame);

        try {
            ResultSet resultSet = Methods.getTableData("COUNT(*)", "worker",
            "wrk_br_code = " + branch);  //Vriskoume posous worker exoume
            resultSet.next();
            int rowCount = resultSet.getInt(1);

            //Pairnei ths plhrofories pou zhtountai apo ton pinaka worker
            resultSet = Methods.getTableData("CONCAT(wrk_name, ' ', wrk_lname) as
            Fullname, wrk_salary Salary", "worker", "wrk_br_code = " + branch);
            ResultSetMetaData rsmd = resultSet.getMetaData();
            columnCount = rsmd.getColumnCount();
            rowInfo = new String[rowCount][columnCount];

            columnNames = new String[columnCount];
            for(int i=0;i<columnCount;i++){
                columnNames[i] = rsmd.getColumnLabel(i+1);     //Vazoume ta onomata twn
                columns ston pinaka columnName
            }

            for(int i=0;i<rowCount;i++) {
                resultSet.next();
                for(int j=0;j<columnCount;j++){
                    rowInfo[i][j]=resultSet.getString(columnNames[j]);       //Vazoume
                    ta data sthn lista rowInfo pou tha xrhsimopoihsoume gia to JTable

                    if (columnNames[j].equals("Salary")) {
                        totalSalaries += resultSet.getInt(columnNames[j]);
                    }
                }
            }

            //Ftiaxnoume to table kai to pane kai to vazoume sto frame, yo
            JTable jTable = new JTable(rowInfo,columnNames) {
                @Override
                public boolean isCellEditable(int row, int column) {
                    return false;
                }
            };
            jTable.setPreferredScrollableViewportSize(new Dimension(450,63));
            jTable.setFillsViewportHeight(true);
            jTable.setCellSelectionEnabled(false);
            jTable.setRowSelectionAllowed(false);

            scrollPaneSalary = new JScrollPane(jTable);     //Ftiaxnoume to scrollPane
            gia ton teliko mistho
            scrollPaneSalary.setBounds(400, 565, 607, 300);
            SwingUtilities.updateComponentTreeUI(frame);    //Kanoume refresh gia na
            fugei oti exei ksemeinei
            frame.add(scrollPaneSalary);

            salaryLabel.setText("Total salary expenses: " + totalSalaries);     //Settin
            up salaryLabel
            salaryLabel.setBounds(1070,700,200,35);
            salaryLabel.setFont(new Font(null, Font.BOLD,12));
            salaryLabel.setForeground(Color.RED);
            frame.add(salaryLabel);




        } catch (SQLException eSQL) {
            System.out.println(eSQL.getMessage());
            System.out.println("Error, please check your input.");
            eSQL.printStackTrace();
        }
    }
    public static void tripInfoButton(JFrame frame) throws SQLException {
        submitDateButton.doClick();
        int i;
        frameStatic=frame;
        datetimeLabel = new JLabel("DAY         MONTH       YEAR");     //Ftiaxnoume ta
        labels
        questionLabel = new JLabel("Branch Code:");

        datetimeLabel.setFont(new Font(null, Font.BOLD, 22));
        datetimeLabel.setBounds(812,350,340,450);
        questionLabel.setFont(new Font(null, Font.BOLD, 18));
        questionLabel.setBounds(1100,50,340,450);
        frame.add(datetimeLabel);
        frame.add(questionLabel);
        String[] days = new String[31];
        String[] months = new String[12];
        String[] years = new String[5];

        for(i=0;i<31;i++){
            days[i]=Integer.toString(i+1);          //Meres
        }
        for(i=0;i<12;i++){
            months[i]=Integer.toString(i+1);        //Mhnes
        }
        for(i=0;i<5;i++){
            years[i]=Integer.toString(i+2020);      //Xronia
        }



        ResultSet resultSet = customQueryReturn("SELECT COUNT(*) FROM branch");
        //Vriskw posa branch exw
        resultSet.next();
        int count = resultSet.getInt(1);
        String[] branchIDs = new String[count];             //Ftiaxnw pinaka gia ta
        branchIDs

        for(i=0;i<count;i++){
            branchIDs[i]=Integer.toString(i+1);
        }
        chooseDateBox[6] = new JComboBox<>(branchIDs);         //JComboBox me ta branch
        IDs
        chooseDateBox[6].setBounds(1100,300,50,50);
        frame.add(chooseDateBox[6]);

        for(i=0;i<2;i++){
            chooseDateBox[i] = new JComboBox<>(days);
            chooseDateBox[i].setBounds(800,600+i*100,100,50);     //Meres
            frame.add(chooseDateBox[i]);
            chooseDateBox[i+2] = new JComboBox<>(months);
            chooseDateBox[i+2].setBounds(920,600+i*100,100,50);     //Mhnes
            chooseDateBox[i+2].addActionListener(new Methods());
            frame.add(chooseDateBox[i+2]);
            chooseDateBox[i+4] = new JComboBox<>(years);
            chooseDateBox[i+4].setBounds(1040,600+i*100,100,50);        //Xronia
            frame.add(chooseDateBox[i+4]);
        }
        submitDateButton.setBounds(902,810,130,60);
        submitDateButton.addActionListener(new Methods());
        frame.add(submitDateButton);
        SwingUtilities.updateComponentTreeUI(frame);    //Kanoume refresh gia na fugei
        oti exei ksemeinei


        //ResultSet resultSet = customQueryReturn();
    }
    public static void clientOffersButton(JFrame frame){
        String[] lastNames = new String[0];
        frameStatic = frame;
        int rowCount = 0;
        int i;
        try {
            ResultSet resultSet = customQueryReturn("SELECT COUNT(DISTINCT rsof_lname)
            FROM reservation_offers");      //Vriskw posa rows exw apo onomata
            resultSet.next(); //move cursor to start
            rowCount = resultSet.getInt(1);
            resultSet = customQueryReturn("SELECT DISTINCT rsof_lname FROM
            reservation_offers");      //Vriskw ola ta last names pou uparxoun sto
            reservation offers
            lastNames = new String[rowCount];
            i = 0;
            while (resultSet.next()) {
                lastNames[i] = resultSet.getString("rsof_lname");          //Ftiaxnw
                pinaka me last names
                i++;
            }
        }
        catch (SQLException throwables) {
            System.out.println(throwables.getMessage());
            System.out.println("Error, please check your input.");
            throwables.printStackTrace();

        }



        chooselName = new JComboBox<String>(lastNames);
        //Ftiaxnw drop down list me ola ta epwnuma

        if(first){              //Periptwsh pou kaleitai prwth fora to koumpi, thn
        xreiazomaste gia na arxikopoihsoume ton pinaka
            selectedlName = (String) chooselName.getSelectedItem();
            first=false;
        }

        chooselName.setSelectedItem(selectedlName);
        chooselName.addActionListener(new Methods());
        chooselName.setBounds(950,300,250,50);
        frame.add(chooselName);



        try {
            ResultSet resultSet = customQueryReturn("CALL findInfoByLastName('" +
            selectedlName + "')");        //Kanw call to stored procedure 3.1.3.4
            ResultSetMetaData rsmd = null;                       //MetaData gia na vrw
            posa columns exw kai ta onomata autwn
            rsmd = resultSet.getMetaData();
            columnCounterStatic = rsmd.getColumnCount();         //Vriskoume posa
            columns exoume


            String[] columnNames = new String[columnCounterStatic];
            for (i = 0; i < columnCounterStatic; i++) {
                columnNames[i] = rsmd.getColumnLabel(i + 1);     //Vazoume ta onomata
                twn columns ston pinaka columnNames
            }

            i=0;
            while (resultSet.next()){                   //Vriskoume posa rows exei to
            resultSet
                i++;
            }
            String[][] rowInfo = new String[i][columnCounterStatic];

            i=0;
            resultSet.beforeFirst();
            while (resultSet.next()) {                           //Vazoume ta data sthn
            lista rowInfo pou tha xrhsimopoihsoume gia to JTable
                for (int j = 0; j < columnCounterStatic; j++) {
                    rowInfo[i][j] = resultSet.getString(columnNames[j]);
                }
                i++;
            }
            JTable jTable = new JTable(rowInfo, columnNames);               //Ftiaxnw to
            table
            jTable.setPreferredScrollableViewportSize(new Dimension(50, 63));
            jTable.setFillsViewportHeight(true);

            scrollPane = new JScrollPane(jTable);                   //Ftiaxnw to pane
            gia na valw mesa to table
            scrollPane.setBounds(400, 210, 507, 300);
            jTable.setCellSelectionEnabled(false);
            jTable.setRowSelectionAllowed(true);
            frame.add(scrollPane);
        }
        catch (SQLException throwables) {
            System.out.println(throwables.getMessage());
            System.out.println("Error, please check your input.");
            throwables.printStackTrace();
        }

    }

    public static boolean contains(final int[] array, final int key) {        //Vriskei
    an uparxei ena key se ena array
        return Arrays.stream(array).anyMatch(i -> i == key);
    }


    @Override
//ACTION PERFORMED GIA TO SUBMIT BUTTON TOU INSERT DATA
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == submitButton) {
            String query = "INSERT INTO " + tableStatic + " VALUES (";     //Ftiaxnw thn
            arxh tou query

            int k=0;
            for (int i = 1; i <=counterStatic; i++) {

                String columnText = fields.get(i).getText();


                if(contains(dropdownArray, i)){                 //Periptwsh pou exw
                dropdown array
                    query = query + "'" + chooseKey.get(k).getSelectedItem() + "'";
                    k++;
                }
                else if (fields.get(i).getText().equals("") ||
                columnNameStatic.contains(columnText)) {      //Periptwsh pou den exw
                grapsei sto field
                    query = query + "null";
                }
                else
                    query = query + "'" + fields.get(i).getText().trim() + "'";

                if (i < counterStatic)
                    query = query + ",";
            }

            query = query + ");";
            try {
                customQuery(query);
            }
            catch (Exception e2){
                System.out.println("Error, please check your input.");
                e2.printStackTrace();
            }
        }

        if (e.getSource() == deleteButton) {
            String condition=" ";
            int j;
            for(j=0;j<columnCounterStatic;j++){
                if(valueStatic[j]!=null) {
                    condition = condition +
                    columnNameStatic.get(j)/*columnNameStatic[j]*/ + "='" +
                    valueStatic[j] + "'";
                    if (columnCounterStatic - 1 > j && valueStatic[j+1]!=null)
                        condition = condition + " AND ";
                }
            }
            //String condition = primaryName + " IN ( " + "SELECT " + primaryName + "
            FROM ( SELECT * FROM " + tableStatic + " ORDER BY " + primaryName + " ASC
            LIMIT " + selRowStatic + ",1)tmp);";
            String query="DELETE FROM " + tableStatic + " WHERE " + condition;
            customQuery(query);
        }



        if (e.getSource() == submitITButton) {

            // Insert into worker
            String query = "INSERT INTO worker VALUES (";   //Ftiaxnw thn arxh tou query
            query += "'" + fields.get(1).getText().trim() + "',";
            for (int i = 5; i <=counterStatic-1; i++) {
                String columnText = fields.get(i).getText();

                //TODO: at_IT doesnt get caught because real name is wrk_AT
                if (fields.get(i).getText().equals("") ||
                columnNameStatic.contains(columnText)) {      //Periptwsh pou den exw
                grapsei sto field
                    query = query + "null";
                }
                else
                    query = query + "'" + fields.get(i).getText().trim() + "'";

                if (i < counterStatic)
                    query = query + ",";
            }
            //4th field (index = 8) is dropdown menu
            query = query + "'" + chooseKey.get(0).getSelectedItem() + "'";
            query = query + ");";
            try {
                customQuery(query);

            }
            catch (Exception e2){
                System.out.println("Error, please check your input.");
                e2.printStackTrace();
            }

            //Insert into IT
            query = "INSERT INTO it VALUES (";    //Ftiaxnw thn arxh tou query
            for (int i = 1; i <=counterStatic-4; i++) {

                if (i==3 && (fields.get(i).getText().equals("") ||
                fields.get(i).getText().equals("it_date_start"))){     //Periptwsh null
                date start
                    fields.get(i).setText(String.valueOf(java.time.LocalDate.now()));
                }

                else if (i==4 && (fields.get(i).getText().equals("") ||
                fields.get(i).getText().equals("it_date_end"))){     //Periptwsh null
                date end
                    query = query + "NULL";
                    continue; //TODO: continue doesn't enter a value for corresponding
                    field?
                }

                //rest cases
                query = query + "'" + fields.get(i).getText().trim() + "'";
                if (i<counterStatic-4) {
                    query = query + ",";
                }
            }
            query = query + ");";
            try {
                customQuery(query);
            }
            catch (Exception e2){
                System.out.println("Error, please check your input.");
                e2.printStackTrace();
            }

        }
```

```
//If month changes, change days
else if (e.getSource() == chooseDateBox[2] || e.getSource() == chooseDateBox[4])
{
    chooseDateBox[0].removeAllItems();


    int selection = chooseDateBox[2].getSelectedIndex()+1;
    if (selection >= 1 && selection <= 7)
        for (int i=1; i<=30+(selection%2); i++)
            chooseDateBox[0].addItem(i);

    else
        for (int i=1; i<=30+((selection-1)%2); i++)
        chooseDateBox[0].addItem(i);


    //FEBRUARY
    if (selection == 2) {
        int isLeapYear = 0;



        if ((chooseDateBox[4].getSelectedIndex()+2020)%4 == 0)
            isLeapYear = 1;



        chooseDateBox[0].removeAllItems();
        for (int i=1; i<=28+isLeapYear; i++) {
            chooseDateBox[0].addItem(i);
        }
    }

    }

    else if (e.getSource() == chooseDateBox[3] || e.getSource() ==
    chooseDateBox[5]) {
        chooseDateBox[1].removeAllItems();


        int selection = chooseDateBox[3].getSelectedIndex()+1;
        if (selection >= 1 && selection <= 7)
            for (int i=1; i<=30+(selection%2); i++)
                chooseDateBox[1].addItem(i);

        else
            for (int i=1; i<=30+((selection-1)%2); i++)
```

```
                    chooseDateBox[1].addItem(i);


                if (selection == 2) {
                    int isLeapYear = 0;


                    if ((chooseDateBox[5].getSelectedIndex()+2020)%4 == 0)
                        isLeapYear = 1;



                    chooseDateBox[1].removeAllItems();
                    for (int i=1; i<=28+isLeapYear; i++) {
                        chooseDateBox[1].addItem(i);
                    }
                }

            }

        if(e.getSource()==submitDateButton) {
            frameStatic.remove(scrollPane);
            SwingUtilities.updateComponentTreeUI(frameStatic);    //Kanoume refresh gia
            na fugei oti exei ksemeinei
            String query;
            ResultSet resultSet;
            int i;
            int rowCount=0;
            String date1 = (chooseDateBox[4].getSelectedIndex() + 2020) + "-" +
            (chooseDateBox[2].getSelectedIndex() + 1) + "-" +
            (chooseDateBox[0].getSelectedIndex() + 1);
            String date2 = (chooseDateBox[5].getSelectedIndex() + 2020) + "-" +
            (chooseDateBox[3].getSelectedIndex() + 1) + "-" +
            (chooseDateBox[1].getSelectedIndex() + 1);
            query = "CALL viewTripsInRange(" + (chooseDateBox[6].getSelectedIndex() + 1)
            + ", '" + date1 + " 00:00:00', '" + date2 + " 00:00:00')";    //Kaloume thn
            stored procedure
            resultSet = customQueryReturn(query);
            ResultSetMetaData rsmd = null;
            try {
                while (resultSet.next()){           //Vriskw posa rows exoume
                    rowCount++;
                }

                rsmd = resultSet.getMetaData();
                columnCounterStatic = rsmd.getColumnCount();         //Vriskoume posa
                columns exoume
                String[][] rowInfo = new String[rowCount][columnCounterStatic];

                String[] columnNames = new String[columnCounterStatic];
                for (i = 0; i < columnCounterStatic; i++) {
                    columnNames[i] = rsmd.getColumnLabel(i + 1);     //Vazoume ta
                    onomata twn columns ston pinaka columnNames
                    columnNameStatic.add(columnNames[i]);
                    //columnNameStatic[i] = columnNames[i];
                    columnTypeStatic[i] = rsmd.getColumnTypeName(i + 1);
                }
                i = -1;
                resultSet.beforeFirst();
                while (resultSet.next()) {
                    i++;
                    for (int j = 0; j < columnCounterStatic; j++) {
                        rowInfo[i][j] = resultSet.getString(columnNames[j]);
                        //Vazoume ta data sthn lista rowInfo pou tha xrhsimopoihsoume
                        gia to JTable
                    }
                }

                //Ftiaxnoume to table kai to pane kai to vazoume sto frame, yo
                JTable jTable = new JTable(rowInfo, columnNames);
                jTable.setPreferredScrollableViewportSize(new Dimension(50, 63));
                jTable.setFillsViewportHeight(true);

                scrollPane = new JScrollPane(jTable);
                scrollPane.setBounds(400, 210, 650, 300);
                SwingUtilities.updateComponentTreeUI(frameStatic);    //Kanoume refresh
                gia na fugei oti exei ksemeinei
                frameStatic.add(scrollPane);
            } catch (SQLException throwables) {
                System.out.println(throwables.getMessage());
                System.out.println("Error, please check your input.");
                throwables.printStackTrace();
            }
        }

        if(e.getSource()==chooselName){

            ItOptionsFrame.removeAll(ItOptionsFrame.counter,frameStatic);
            selectedlName = (String) chooselName.getSelectedItem();
            clientOffersButton(frameStatic);
        }
    }

    //Ta parakatw ulopoihhthhkan sthn UpdateTableButton
    @Override
    public void valueChanged(ListSelectionEvent e) {
    }
    @Override
    public void handleEvent(Event evt) {

    }
}
```

D.viii RedirectAppOutputStream.java

```
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintStream;
import javax.swing.*;
import javax.swing.text.DefaultCaret; // ?


public class RedirectAppOutputStream {          //Kanei ta stream tou System.out na
phgainoun se ena JTextField

    private class GuiOutputStream extends OutputStream {
        JTextArea textArea;

        public GuiOutputStream(JTextArea textArea) {
            this.textArea = textArea;
        }

        @Override
        public void write(int data) throws IOException {           //Grafei sto textArea
        anti gia to IDE
            textArea.append(new String(new byte[] { (byte) data }));
        }
    }

    public void guiConsoleTest(JFrame frame) {

        //Ftiaxnoume to JTextArea, ScrollPane kai ta vazoume sto frame

        //https://docs.oracle.com/javase/7/docs/api/javax/swing/text/JTextComponent.html#setEditable(
        JTextArea textArea = new JTextArea();
        textArea.setEditable(false);
        //https://kodejava.org/how-do-i-wrap-the-text-lines-in-jtextarea/
        textArea.setLineWrap(true);
        textArea.setWrapStyleWord(true);
        textArea.setBounds(10,440,330,450);

        //Autoscroll
        //https://tips4java.wordpress.com/2008/10/22/text-area-scrolling/
        //https://docs.oracle.com/javase/7/docs/api/javax/swing/text/DefaultCaret.html
        DefaultCaret caret = (DefaultCaret)textArea.getCaret();
        caret.setUpdatePolicy(DefaultCaret.ALWAYS_UPDATE);

        JScrollPane scroll = new JScrollPane (textArea,
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS, JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
        scroll.setBounds(10,440,330,450);
        frame.add(scroll);
        SwingUtilities.updateComponentTreeUI(frame);
        frame.setVisible (true);

        GuiOutputStream rawout = new GuiOutputStream(textArea);     //Vazei to stream
        sto textArea
        System.setOut(new PrintStream(rawout, true));
    }
}
```