



ΜΑΘΗΜΑ: Βάσεις Δεδομένων
Τομέας Λογικού των Υπολογιστών

Χειμερινό Εξάμηνο 2022

Project Εργαστηρίου Βάσεων Δεδομένων Ακαδ. Έτους 2022 – 2023

ΔΙΔΑΣΚΟΝΤΕΣ: Μεγαλοικονόμου Βασίλειος, Βογιατζάκη Ελένη,
Βασιλόπουλος Ιωάννης

ΗΜ/ΝΙΑ ΠΑΡΑΔΟΣΗΣ: 06/02/2023

ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΩΝ:

Καραγιάννης Γεώργιος	1084586	up1084586@upnet.gr	Ε' Εξάμηνο
Κολάγκη Ευαγγελία	1084599	up1084599@upnet.gr	Ε' Εξάμηνο
Μαντές Μηλιτιάδης	1084661	up1084661@upnet.gr	Ε' Εξάμηνο

ΠΕΡΙΕΧΟΜΕΝΑ

Μέρος A : MySQL

A.1 Ορισμός και κατασκευή Βάσης Δεδομένων

A.1.1 Σχεσιακό Διάγραμμα	3
A.1.2 Παραδοχές - Τροποποιήσεις	4
A.1.3 Κατασκευή και Συμπλήρωση Πινάκων	5

A.2 Stored Procedures

A.2.1 Κώδικας Stored Procedures	22
A.2.2 Παραδείγματα εκτέλεσης Stored Procedures	28

A.3 Triggers

A.3.1 Κώδικας Triggers	32
A.3.2 Παραδείγματα εκτέλεσης Triggers	44

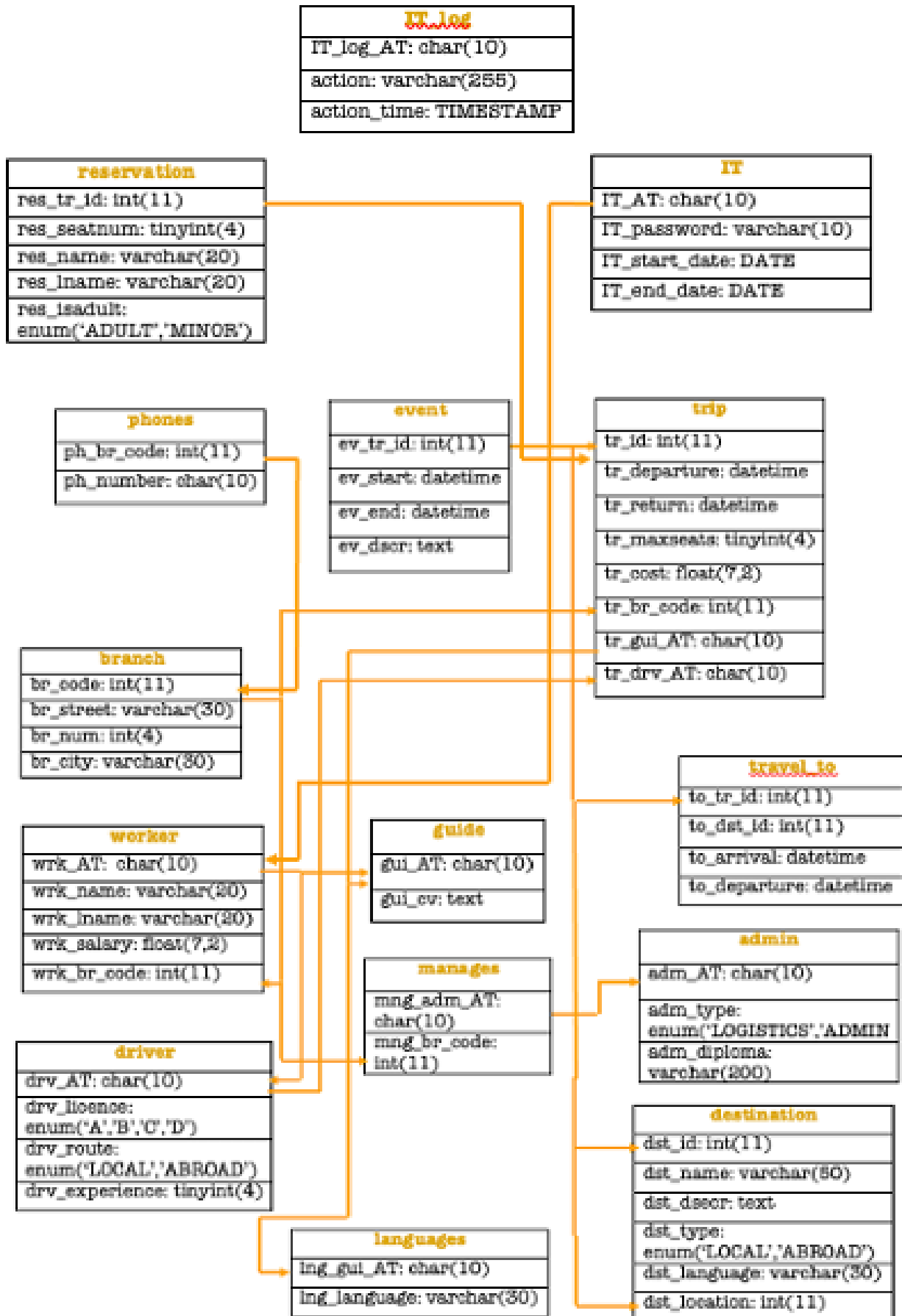
Μέρος B : GUI

B.1 Κατασκευή γραφικής διεπαφής – Επικοινωνία με ΒΔ

B.1.1 Κώδικας διεπαφής σε Java – JDBC	47
B.1.2 Σενάριο χρήσης	53

A.1 Ορισμός και κατασκευή Βάσης Δεδομένων

A.1.1 Σχεσιακό Διάγραμμα



A.1.2 Παραδοχές - Τροποποιήσεις

Αρχική μας προτεραιότητα ήταν να προσαρμόσουμε την βάση μας στις νέες απαιτήσεις. Σύμφωνα με το ερώτημα 3.1.2.1 δημιουργήσαμε έναν νέο πίνακα *IT* και εισαγάγαμε τα στοιχεία του υπευθυνου πληροφορικής στους πίνακες *worker* και *IT*. Έπειτα, ορίσαμε και έναν πίνακα *log*, στον οποίο εκχωρούμε στις στήλες του το όνομα του χρήστη που έκανε μια ενέργεια ('INSERT', 'UPDATE', 'DELETE'), ποιά ενέργεια ήταν αυτή, πότε έγινε και τι αλλαγές προκάλεσε. Τα αντίστοιχα πεδία του πίνακα τα ονομάσαμε **user**, **action**, **changes**, **time**.

A.1.3 Κατασκευή και Συμπλήρωση Πινάκων

A.1.3.1 Κατασκευή βάσης και πινάκων:

```
CREATE DATABASE travel_agency;
```

```
USE travel_agency;
```

```
CREATE TABLE branch(  
    br_code INT(11) NOT NULL AUTO_INCREMENT,  
    br_street VARCHAR(30) NOT NULL DEFAULT 'unknown',  
    br_num INT(4) NOT NULL DEFAULT '0',  
    br_city VARCHAR(30) DEFAULT 'unknown' NOT NULL,  
    PRIMARY KEY(br_code)  
)engine=InnoDB;
```

```
CREATE TABLE phones(  
    ph_br_code INT(11) NOT NULL,  
    ph_number CHAR(10) NOT NULL DEFAULT 'unknown',  
    PRIMARY KEY(ph_br_code, ph_number),  
    CONSTRAINT PHONECODE FOREIGN KEY (ph_br_code) REFERENCES branch(br_code)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)engine=InnoDB;
```

```
CREATE TABLE worker(  
    wrk_AT CHAR(10) DEFAULT 'unknown' 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 BRAWORKER FOREIGN KEY (wrk_br_code) REFERENCES branch(br_code)  
    ON UPDATE CASCADE ON DELETE CASCADE  
)engine=InnoDB;
```

```
CREATE TABLE guide(  
    gui_AT CHAR(10) NOT NULL,  
    gui_cv TEXT,  
    PRIMARY KEY(gui_AT),  
    CONSTRAINT GUIDEAT FOREIGN KEY (gui_AT) REFERENCES worker(wrk_AT)
```

```
ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;
```

```
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) DEFAULT '0' NOT NULL,
    PRIMARY KEY (drv_AT),
    CONSTRAINT WORDRIVER FOREIGN KEY (drv_AT) REFERENCES worker(wrk_AT)
    ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;
```

```
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 default '0',
    tr_cost float(7,2) not null default '0',
    tr_br_code int(11) NOT NULL default '0',
    tr_gui_AT CHAR(10) NOT NULL DEFAULT '0',
    tr_drv_AT char(10) not null default '0',
    PRIMARY KEY (tr_id),
    CONSTRAINT brcode FOREIGN KEY (tr_br_code) REFERENCES branch(br_code)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT drvAT FOREIGN KEY (tr_drv_AT) REFERENCES driver(drv_AT)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT guideaAT FOREIGN KEY (tr_gui_AT) REFERENCES guide(gui_AT)
    ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;
```

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

```
CONSTRAINT EVENTID FOREIGN KEY (ev_tr_id) REFERENCES trip(tr_id)
ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;
```

```
CREATE TABLE admin(
  adm_AT CHAR(10) NOT NULL,
  adm_type ENUM('LOGISTICS','ADMINISTRATIVE','ACCOUNTING') NOT NULL,
  adm_diploma VARCHAR(200) NOT NULL DEFAULT 'unknown',
  PRIMARY KEY(adm_AT),
  CONSTRAINT ADMINAT FOREIGN KEY (adm_AT) REFERENCES worker(wrk_AT)
ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;
```

```
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 MNGAT FOREIGN KEY (mng_adm_AT) REFERENCES admin(adm_AT)
ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT MNGCODE FOREIGN KEY (mng_br_code) REFERENCES branch(br_code)
ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;
```

```
CREATE TABLE destination(
  dst_id INT(11) NOT NULL AUTO_INCREMENT,
  dst_name VARCHAR(50) NOT NULL DEFAULT 'unknown',
  dst_descr TEXT,
  dst_rtype ENUM('LOCAL','ABROAD'),
  dst_language VARCHAR(30) NOT NULL DEFAULT 'unknown',
  dst_location INT(11) DEFAULT '0',
  PRIMARY KEY(dst_id),
  CONSTRAINT LOCATION FOREIGN KEY (dst_location) REFERENCES destination(dst_id)
ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;
```

```
CREATE TABLE languages(
  lng_gui_AT CHAR(10) NOT NULL,
```

```

lng_language VARCHAR(30) NOT NULL DEFAULT 'unknown',
PRIMARY KEY(lng_gui_AT,lng_language),
CONSTRAINT GUIDELANGUAGE FOREIGN KEY (lng_gui_AT) REFERENCES guide(gui_AT)
ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;

```

```

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 TRAVEID FOREIGN KEY (to_tr_id) REFERENCES trip(tr_id)
    ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT DESTINATIONID FOREIGN KEY (to_dst_id) REFERENCES destination(dst_id)
    ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;

```

```

CREATE TABLE reservation(
    res_tr_id INT(11) NOT NULL,
    res_seatnum TINYINT(4) NOT NULL DEFAULT '0',
    res_name VARCHAR(20) NOT NULL DEFAULT 'unknown',
    res_lname VARCHAR(20) NOT NULL DEFAULT 'unknown',
    res_isadult ENUM('ADULT','MINOR'),
    PRIMARY KEY(res_tr_id, res_seatnum),
    CONSTRAINT TRIPRESERVATION FOREIGN KEY (res_tr_id) REFERENCES trip(tr_id)
    ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;

```

```

CREATE TABLE IT(
    IT_AT CHAR(10) NOT NULL,
    IT_password varchar(10) NOT NULL DEFAULT 'password',
    IT_start_date DATE NOT NULL,
    IT_end_date DATE NOT NULL,
    PRIMARY KEY (IT_AT),
    CONSTRAINT WORKERIT FOREIGN KEY
    (IT_AT) REFERENCES worker(wrk_AT)

```


ON UPDATE CASCADE ON DELETE CASCADE
)engine=InnoDB;

A.1.3.2 Εισαγωγή τιμών στους πίνακες:

INSERT INTO branch VALUES

(null, 'AG.NIKOLAOU', 4, 'PATRA'),
(null, 'AIOLOU', 6, 'PATRA'),
(null, 'KALAVRYTON', 23, 'IOANNINA'),
(null, 'EL.VENIZELOU', 89, 'PATRA'),
(null, 'GOUNARI', 1, 'ATHINA'),
(null, 'SMIRNIS', 34, 'PATRA'),
(null, 'MESATIDOS', 9, 'THESALLONIKI'),
(null, 'EGLIKADOS', 7, 'PATRA'),
(null, 'KOLOKOTRONI', 8, 'ATHINA'),
(null, 'DIOIKITIRIOU', 20, 'PTOLEMAIDA'),
(null, 'PALAIOLOGOU', 11, 'LAMIA'),
(null, 'KANAKARI', 24, 'THESALONIKI'),
(null, 'KORINTHOU', 3, 'PATRA'),
(null, 'ANTHIAS', 5, 'KORINTHOS'),
(null, 'DIMOKRATIAS', 47, 'PATRA'),
(null, 'EVOIAS', 27, 'IOANNINA'),
(null, 'FILOPOIMENOS', 46, 'ARTA'),
(null, 'ERMOU', 17, 'ATHINA'),
(null, 'AKROTIRIOU', 89, 'LAMIA'),
(null, 'ZAIMI', 75, 'LARISA'),
(null, 'PANEPISTIMIOU', 34, 'ATHINA'),
(null, 'DOROU', 1, 'LAMIA'),
(null, 'MAIZONOS', 38, 'PATRA'),
(null, 'SAPFOUS', 44, 'VOLOS'),
(null, 'PAPAMAUROU', 27, 'IOANNINA'),
(null, 'AVEROF', 33, 'TRIPOLI'),
(null, 'GEORGIOU', 101, 'ATHINA'),
(null, 'PEILIDON', 9, 'LAMIA'),
(null, 'PATROKLOU', 11, 'RODOS'),
(null, 'ESLIN', 38, 'LARISA');

INSERT INTO worker VALUES

('AA09895623', 'GIORGOS', 'KARAGIANNIS', 2056, 3),
('AA34520091', 'GIORGOS', 'GIANOPOULOS', 1500, 4),
('AB57391132', 'PANAGIOTIS', 'LAZAS', 1000, 4),
('AB55400231', 'MILTADIS', 'PANAGOPOULOS', 1755, 2),
('AF45554678', 'GIORGOS', 'KAZATZAKIS', 3000, 6),
('AG12343210', 'GIANNIS', 'PAPADOPOULOS', 2355, 6),
('GH96986576', 'HRAKLIS', 'KOULOS', 1010, 6),
('IU90334567', 'SOKRATIS', 'KARAGIANNIS', 95050, 2),
('ER23556438', 'ORESTHS', 'PAPAGIANNIS', 1115, 9),
('DE34554324', 'XRISTOS', 'TZAMIS', 3556, 9),
('AM15375947', 'DIONISIS', 'MANOLOPOULOS', 1500, 11),
('AI83950702', 'MARIA', 'DWROU', 1950, 12),
('AT49862157', 'MANOLIS', 'MANESIS', 1000, 12),
('AM15370047', 'CHRISTOS', 'GEORGOPOULOS', 1850, 11),
('AN03060804', 'EVANGELIA', 'PATZOPOULOY', 1600, 11),
('AN89812526', 'XRISTOFOROS', 'LAGIOS', 1800, 16),
('AK12161135', 'SERGIOS', 'XARAVGIS', 2100, 6),
('IN87486573', 'ALEXANDRA', 'TSAKIRI', 2400, 18),
('AY04554865', 'NIKOLAOS', 'KWNSTANTINOU', 1700, 19),
('AB88988450', 'KONSTANTINA', 'KAFETZI', 1800, 20),
('RY19087762', 'MARIA', 'PAPADOPOULOU', 1000, 21),
('WS12345678', 'ELENI', 'GEORGOPOULOU', 2100, 22),
('TG01020304', 'AGGELOS', 'PAPADAKIS', 1500, 23),
('KK10111213', 'DIMITRIS', 'IOANNOU', 800, 25),
('AS23776510', 'EIRINI', 'ZERVA', 1300, 24),
('QM98566671', 'PAULOS', 'KOSTOPOULOS', 3400, 26),
('AP19992231', 'KOSTAS', 'KATSAROS', 2110, 28),
('MN55776688', 'GEORGIA', 'PANAGOU', 1800, 27),
('HK10010120', 'MANOLIS', 'PETROPOULOS', 990, 7),
('PR20987721', 'SPYROS', 'GEORGIU', 1200, 22),
('AB20967621', 'EVA', 'STERGIU', 1200, 23),
('KO20959000', 'GIORGOS', 'MALTEZOS', 1200, 24),
('IK26285423', 'MILTADIS', 'RODITIS', 1200, 24),
('AT24646601', 'ORESTIS', 'GEORGIADIS', 1200, 7),
('AN89980021', 'PETROS', 'PANAGIOTOU', 1200, 26),
('AM98754326', 'GIANNIS', 'PANOY', 1000.00, 1),

('GH96956706', 'KONSTANTIKOS', 'KANELLOPOULOS', 573.88, 2),
 ('IA00834567', 'IASONAS', 'XARMPILAS', 889.00, 3),
 ('AR23678738', 'FILLIPOS', 'ASTERIWTHS', 1110, 4),
 ('AR98889078', 'ANDRIANNA', 'KARAKWSTA', 980, 5),
 ('AM10999737', 'KATERINA', 'MANOLOPOULOY', 1500, 6),
 ('AI85678002', 'EYGENEIA', 'KOYNAVI', 1700, 7),
 ('AT42323457', 'OLGA', 'THEODOROPOULOY', 1650, 8),
 ('AB15370047', 'EYTYXIA', 'KORAKA', 1800, 9),
 ('GH03060804', 'ARETI', 'XAMPAKAKI', 1600, 10),
 ('JK89699526', 'XARITINI', 'KAKOYRI', 1800, 11),
 ('AW12161135', 'BASILIKI', 'ANTWNIOY', 2000, 12),
 ('NL87486573', 'THEODOIS', 'TSAKAS', 2400, 13),
 ('YA04554865', 'AFRODITI', 'LIANTOY', 1770, 14),
 ('FG88555450', 'GEORGIA', 'HLIOPOULOU', 1890, 15),
 ('SA56789762', 'ALEXANDROS', 'GEORGOPOULOS', 1090, 16),
 ('WE56789068', 'THEODOSIS', 'FILIS', 2150, 17),
 ('TA12345634', 'LEFTERIS', 'RASSIOS', 1500, 18),
 ('VA45678213', 'CHRISTOS', 'STAMATOPOULOS', 970, 19),
 ('AS23785450', 'OLYMPIA', 'ALEXANDROPOULOU', 1320, 20),
 ('AI56366671', 'PANAGIOTA', 'GOURDOUPI', 3100, 21),
 ('OP56782231', 'PAVLOS', 'LAZANAS', 1210, 22),
 ('NM23456688', 'IOANNIS', 'LAGIOS', 1560, 23),
 ('HA45560120', 'BIKY', 'KOYNELI', 790, 24),
 ('IO13342721', 'EYTHIMIS', 'VERRAS', 1280, 25),
 ('AF06219623', 'ARISTEA', 'PAPATHANASIOU', 1400, 26),
 ('KA20956789', 'ZWH', 'TZOYRAMANI', 1500, 27),
 ('IL26285546', 'MARILENA', 'NTERTINI', 1600, 28),
 ('IY64698042', 'ANASTASIA', 'MARGARI', 1370, 29),
 ('OK45678281', 'PARASKEVI', 'GKOTSH', 1900, 30);

INSERT INTO driver VALUES

('RY19087762', 'A', 'LOCAL', 21),
 ('WS12345678', 'A', 'ABROAD', 22),
 ('TG01020304', 'B', 'ABROAD', 23),
 ('KK10111213', 'A', 'LOCAL', 24),
 ('AS23776510', 'B', 'LOCAL', 25),
 ('QM98566671', 'C', 'LOCAL', 26),

('AP19992231', 'C', 'LOCAL', 27),
('MN55776688', 'D', 'ABROAD', 28),
('HK10010120', 'A', 'ABROAD', 29),
('AB20967621', 'C', 'ABROAD', 30),
('KO20959000', 'B', 'ABROAD', 35),
('IK26285423', 'B', 'ABROAD', 47),
('AT24646601', 'B', 'ABROAD', 79),
('AN89980021', 'A', 'ABROAD', 55);

INSERT INTO guide VALUES

('AA09895623', 'EXEI 2 XRONIA PROIPIRESIAS KAI MILLAEI AGGLIKA'),
('AA34520091', 'EXEI 1 XRONO PROIPIRESIAS KAI MILAEI GALLIKA KAI AGGLIKA'),
('AB57391132', 'EXEI 3 XRONIA PROIPIRESIAS KAI MILAEI GERMANIKA'),
('AB55400231', 'EXEI 2 XRONIA PROIPIRESIAS KAI MILAEI PORTOGALIKA'),
('AF45554678', 'EXEI 5 XRONIA PROIPIRESIAS KAI MILAEI GALLIKA KAI GERMANIKA'),
('AG12343210', 'EXEI 4 XRONIA PROIPIRESIAS KAI MILAEI AGGLIKA'),
('GH96986576', 'EXEI 1 XRONIO PROIPIRESIAS MONO'),
('IU90334567', 'EXEI 4 XRONIA PROIPIRESIAS KAI MILAEI GERMANIKA KAI ISPANIKA'),
('ER23556438', 'EXEI 2 XRONIA PROIPIRESIAS KAI MILAEI GERMANIKA'),
('DE34554324', 'EXEI 7 XRONIA PROIPIRESIAS KAI MILAEI GERMANIKA AGGLIKA KAI
TOURKIKA');

INSERT INTO languages VALUES

('AA09895623', 'AGGLIKA'),
('AA34520091', 'GALLIKA KAI AGGLIKA'),
('AB57391132', 'GERMANIKA'),
('AB55400231', 'PORTOGALIKA'),
('AF45554678', 'GALLIKA KAI GERMANIKA'),
('AG12343210', 'AGGLIKA'),
('GH96986576', 'KAMIA'),
('IU90334567', 'GERMANIKA KAI ISPANIKA'),
('ER23556438', 'GERMANIKA'),
('DE34554324', 'GERMANIKA AGGLIKA KAI TOURKIKA');

INSERT INTO phones VALUES

(1, '2610323444'),
(2, '2610343518'),

(3, '2651123456'),
(4, '2610698525'),
(5, '2100345542'),
(6, '2610258961'),
(7, '2310207892'),
(8, '2610741253'),
(9, '2463123124'),
(10, '2231888959'),
(11, '2231076306'),
(12, '2310538897'),
(13, '2610524438'),
(14, '2741361049'),
(15, '2610987780'),
(16, '2651526041'),
(17, '2231990760'),
(18, '2107566433'),
(19, '2231876753'),
(20, '2410533264'),
(21, '2231029875'),
(22, '2610198236'),
(23, '2421054767'),
(24, '2651008568'),
(25, '2710760099'),
(26, '2101098430'),
(27, '2231020001'),
(28, '2241089142'),
(29, '2410187663'),
(30, '2310993564');

INSERT INTO admin VALUES

('AM98754326', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON'),
('GH96956706', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN-IOANNINWN'),
('IA00834567', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN-ASOE'),
('AR23678738', 'ADMINISTRATIVE', 'DIOIKISIS EPIXEIRISEWN-TRIPOLI'),
('AR98889078', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN-PATRWN'),
('AM10999737', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN'),
('AI85678002', 'ADMINISTRATIVE', 'MARKETING'),

('AT42323457', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN KAI TEXNOLOGIAS'),
 ('AB15370047', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN PATRWN'),
 ('GH03060804', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON'),
 ('JK89699526', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON KAI TEXNOLOGIAS'),
 ('AW12161135', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON-TRIPOLIS'),
 ('NL87486573', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN-IOANNINWN'),
 ('YA04554865', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN-PATRWN'),
 ('FG88555450', 'ADMINISTRATIVE', 'MARKETING-OPA'),
 ('SA56789762', 'ADMINISTRATIVE', 'DIMOSIAS DIOIKISIS-PANEPISTIMIO ATHINWN'),
 ('WE56789068', 'ADMINISTRATIVE', 'DIETHNWN KAI EYRWPAIKWN SPOUDWN'),
 ('TA12345634', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON'),
 ('VA45678213', 'ADMINISTRATIVE', 'MARKETING'),
 ('AS23785450', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN'),
 ('AI56366671', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON'),
 ('OP56782231', 'ADMINISTRATIVE', 'DIETHNWN KAI EYRWPAIKWN SPOUDWN'),
 ('NM23456688', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON'),
 ('HA45560120', 'ADMINISTRATIVE', 'DIETHNWN KAI EYRWPAIKWN SPOUDWN'),
 ('IO13342721', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN'),
 ('AF06219623', 'ADMINISTRATIVE', 'DIETHNWN KAI EYRWPAIKWN SPOUDWN'),
 ('KA20956789', 'ADMINISTRATIVE', 'OIKONOMIKVN EPISTHMWN'),
 ('IL26285546', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON'),
 ('IY64698042', 'ADMINISTRATIVE', 'DIETHNWN KAI EYRWPAIKWN SPOUDWN'),
 ('OK45678281', 'ADMINISTRATIVE', 'DIOIKISIS_EPIXEIRISEON'),
 ('AM15375947', 'ACCOUNTING', 'OIKONOMIKHS KAI PERIFEREIAKIS ANAPTYKSHS'),
 ('AI83950702', 'ACCOUNTING', 'EFARMOSMENA_OIKONOMIKA'),
 ('AT49862157', 'LOGISTICS', 'OKONOMIKON'),
 ('AM15370047', 'ACCOUNTING', 'OKONOMIKON'),
 ('AN03060804', 'LOGISTICS', 'DIOIKISIS_EPIXEIRISEON'),
 ('AN89812526', 'ACCOUNTING', 'EFARMOSMENA_OIKONOMIKA'),
 ('AK12161135', 'LOGISTICS', 'EFARMOSMENA_OIKONOMIKA'),
 ('IN87486573', 'ACCOUNTING', 'OKONOMIKON'),
 ('AY04554865', 'LOGISTICS', 'EFARMOSMENA_OIKONOMIKA'),
 ('AB88988450', 'ACCOUNTING', 'DIOIKISIS_EPIXEIRISEON');

INSERT INTO manages VALUES

('AM98754326', 1),
 ('GH96956706', 2),

('IA00834567', 3),
 ('AR23678738', 4),
 ('AR98889078', 5),
 ('AM10999737', 6),
 ('AI85678002', 7),
 ('AT42323457', 8),
 ('AB15370047', 9),
 ('GH03060804', 10),
 ('JK89699526', 11),
 ('AW12161135', 12),
 ('NL87486573', 13),
 ('YA04554865', 14),
 ('FG88555450', 15),
 ('SA56789762', 16),
 ('WE56789068', 17),
 ('TA12345634', 18),
 ('VA45678213', 19),
 ('AS23785450', 20),
 ('AI56366671', 21),
 ('OP56782231', 22),
 ('NM23456688', 23),
 ('HA45560120', 24),
 ('IO13342721', 25),
 ('AF06219623', 26),
 ('KA20956789', 27),
 ('IL26285546', 28),
 ('TY64698042', 29),
 ('OK45678281', 30);

INSERT INTO trip VALUES

(1001101, '2023-02-05 7:30:00', '2023-02-15 20:15:00', 30, 4000, 1, 'AA09895623',
 'RY19087762'),
 (1101101, '2023-02-10 8:25:00', '2023-02-20 19:37:00', 25, 6375, 2, 'AB57391132',
 'WS12345678'),
 (1101010, '2023-04-06 6:50:00', '2023-04-16 17:30:00', 33, 5475, 4, 'AB57391132',
 'TG01020304'),

(0101011, '2023-04-15 10:45:00', '2023-04-25 21:00:00', 40, 6890, 5, 'AB55400231',
'KK10111213'),
(0011001, '2023-05-01 9:55:00', '2023-05-11 22:10:00', 45, 4120, 7, 'AF45554678',
'AS23776510'),
(0000001, '2023-10-04 7:37:00', '2023-10-14 23:15:00', 35, 5900, 9, 'AG12343210',
'QM98566671'),
(0010110, '2023-10-20 5:30:00', '2023-10-30 20:55:00', 48, 8700, 10, 'AG12343210',
'AP19992231'),
(1111110, '2023-11-05 6:25:00', '2023-11-15 21:35:00', 40, 5700, 3, 'IU90334567',
'MN55776688'),
(1011100, '2023-12-15 8:36:00', '2023-12-25 19:45:00', 70, 15000, 6, 'ER23556438',
'HK10010120'),
(0111110, '2023-12-20 12:56:00', '2023-12-30 21:40:00', 70, 20000, 8, 'DE34554324',
'AB20967621'),
(20910040, '2023-06-24 7:30:00', '2023-06-30 20:30:00', 35, 1000, 11, 'DE34554324',
'TG01020304'),
(11100110, '2023-07-16 8:30:00', '2023-07-26 20:37:00', 20, 1550, 12, 'AG12343210',
'RY19087762'),
(10105557, '2023-08-15 9:30:00', '2023-08-25 21:30:00', 20, 2000, 14, 'AA09895623',
'WS12345678'),
(10105558, '2023-08-16 9:45:00', '2023-08-22 22:00:00', 20, 990, 15, 'DE34554324',
'AP19992231'),
(10105559, '2023-05-14 12:00:00', '2023-05-19 21:10:00', 15, 1600, 17, 'ER23556438',
'KK10111213'),
(10105560, '2023-08-12 7:00:00', '2023-08-17 20:15:00', 30, 2100, 19, 'AA34520091',
'AS23776510'),
(10105561, '2023-09-24 8:45:00', '2023-09-29 20:00:00', 30, 1000, 20, 'DE34554324',
'QM98566671'),
(10105562, '2023-09-9 6:55:00', '2023-09-15 21:00:00', 10, 1300, 13, 'AF45554678',
'KO20959000'),
(10105563, '2023-12-20 10:00:00', '2023-12-28 20:35:00', 50, 4000, 16, 'ER23556438',
'MN55776688'),
(10105564, '2023-10-02 12:15:00', '2023-10-16 21:30:00', 30, 2600, 18, 'IU90334567',
'HK10010120'),
(3900000, '2023-01-07 7:00:00', '2023-01-12 21:00:00', 50, 3000.00, 21, 'AA09895623',
'KK10111213'),

(3901001, '2023-08-10 10:15:00', '2023-08-15 15:30:00', 40, 1500.00, 22, 'AB57391132', 'RY19087762'),
 (3932221, '2023-02-18 9:30:00', '2023-02-24 19:00:00', 30, 2400.00, 26, 'AF45554678', 'AT24646601'),
 (2099801, '2023-01-11 6:30:00', '2023-01-14 16:15:00', 20, 3150.50, 29, 'AA34520091', 'AN89980021'),
 (1999990, '2023-05-05 9:00:00', '2023-05-11 20:00:00', 35, 5000.00, 23, 'AB55400231', 'MN55776688'),
 (8566783, '2023-02-04 8:30:00', '2023-02-10 13:30:00', 50, 4700.10, 25, 'DE34554324', 'RY19087762'),
 (3458800, '2023-07-12 10:00:00', '2023-07-19 18:00:00', 30, 1300.00, 24, 'ER23556438', 'QM98566671'),
 (1009870, '2023-06-08 5:30:00', '2023-06-13 11:00:00', 25, 1950.00, 27, 'ER23556438', 'AB20967621'),
 (1001993, '2023-02-01 9:00:00', '2023-02-08 16:15:00', 40, 2000.00, 28, 'GH96986576', 'HK10010120'),
 (1766571, '2023-09-08 9:15:00', '2023-09-14 17:00:00', 40, 2800.00, 30, 'AA09895623', 'AT24646601');

INSERT INTO event VALUES

(1001101, '2023-02-06 11:30:00', '2023-02-06 14:30:00', 'BOLTA STHN PARALIAKI'),
 (1101101, '2023-02-11 14:00:00', '2023-02-11 17:00:00', 'EPISKEPSI STO VATIKANO'),
 (1101010, '2023-04-07 12:00:00', '2023-04-07 14:00:00', 'EPISKEPSI SE TOPIKI PARAGVGH KRASIWN'),
 (0101011, '2023-04-16 10:00:00', '2023-04-16 11:00:00', 'OPERA GARNIER'),
 (0011001, '2023-05-02 10:0:00', '2023-05-04 20:00:00', 'EKDROMI STA XANIA'),
 (0000001, '2023-10-05 14:30:00', '2023-10-05 16:00:00', 'STADIO TIS BARCHELONA.FC'),
 (0010110, '2023-10-21 14:30:00', '2023-10-21 15:00:00', 'EPISKEPSI STON KATHEDRIKO NAO THS POLHS'),
 (1111110, '2023-11-06 14:20:00', '2023-11-06 13:20:00', 'TEST DRIVE AUTOKINITA FERRARI'),
 (1011100, '2023-12-16 10:30:00', '2023-12-16 13:45:00', 'FAGHTO KAI KAFES STHN PLATEIA TAKSIM'),
 (0111110, '2023-12-21 13:50:00', '2023-12-21 21:40:00', 'BOLTA STHN PALIA POLH'),
 (20910040, '2023-06-25 13:30:00', '2023-06-25 20:30:00', 'EKDORMI STO DILOFO'),
 (20910040, '2023-06-26 13:30:00', '2023-06-27 20:30:00', 'EKDROMI STO TSEPELOBO'),
 (10105557, '2023-08-20 11:30:00', '2023-08-20 12:30:00', 'EPISKEPSI STO MOYSEIO THS EPANASTASHS'),

(10105558, '2023-08-17 14:15:00', '2023-08-17 15:00:00', 'EPISKEPSI STA ANAKTORA TOY BUCKINGHAM'),

(10105559, '2023-05-13 12:00:00', '2023-05-13 21:10:00', 'EKDROMI STA XWRIA APEIRATHOS KAI FILOTI'),

(10105560, '2023-08-13 17:00:00', '2023-08-13 20:15:00', 'THEATRIKI PARASTASI STHN SKALA TOY MILANOU'),

(10105561, '2023-09-25 09:00:00', '2023-09-25 20:00:00', 'EKDROMI STO PALERMO'),

(10105562, '2023-09-10 12:55:00', '2023-09-10 21:00:00', 'EPISKEPSI STO RIJKSMUSEUM'),

(10105563, '2023-12-21 10:00:00', '2023-12-21 13:00:00', 'XRISTOYGENNIATIKH AGORA'),

(10105564, '2023-10-03 11:00:00', '2023-10-16 12:30:00', 'BOLTA STO DASOS'),

(3900000, '2023-01-09 11:00:00', '2023-01-09 13:30:00', 'KSENAGISI STO SPITI TOU MOZART'),

(3901001, '2023-08-11 11:00:00', '2023-08-11 12:00:00', 'PERIPATOS STH CHORA'),

(3932221, '2023-02-22 17:00:00', '2023-02-22 19:00:00', 'TOUR STO MESAIWNIKO KASTRO TOU EDIMBOURGOU'),

(2099801, '2023-01-12 12:00:00', '2023-01-12 12:30:00', 'BOLTA STHN GEFYRA TOU KAROLOU'),

(1999990, '2023-05-06 10:00:00', '2023-05-06 14:00:00', 'KROUAZIERA STO GEIRANGERFJORD'),

(8566783, '2023-02-06 13:30:00', '2023-02-06 14:00:00', 'FOTOGRAFIES ME TON AH-BASILH'),

(3458800, '2023-07-14 18:00:00', '2023-07-14 19:30:00', 'BOLTA STA SOKAKIA THS OIAS'),

(1009870, '2023-06-10 11:00:00', '2023-06-10 13:00:00', 'BOLTA STO LIMANI KAI STHN JAFFA'),

(1001993, '2023-02-01 15:00:00', '2023-02-01 17:00:00', 'FAGHTO SE PARADOSIAKH MPYRARIA'),

(1766571, '2023-09-10 10:30:00', '2023-09-10 11:30:00', 'EPISKEPSI STO MOYSEIO SOKOLATAS CHOCOLATE NATION');

INSERT INTO destination VALUES

(4589618, 'THESALLONIKI', 'TAXIDI STIN MAGEYTIKI THESALLONIKI', 'LOCAL', 'ELLINIKA', NULL),

(2579860, 'ROME', 'PERIPLANISI STIN ARXAIA POLI TIS EUROPIS', 'ABROAD', 'ITALIKA', NULL),

(4884915, 'NAPOLI', 'KSENAGISI STHN POLH TIS AYTHENTIKIS PIZZAS', 'ABROAD', 'ITALIKA', NULL),

(6943780, 'PARIS', 'KSENAGISI STIN POLI TOY EROTA', 'ABROAD', 'GALLIKA', NULL),

(5978483, 'KRETA', 'METAFORA STA PARADOSIKA XORIA TIS KRITIS', 'LOCAL', 'ELLINIKA', NULL),

(0321548, 'BARCELONA', 'KSENAGISI STO STADIO TIS BARCELONA.FC',
'ABROAD','ESPANIKA' , NULL),

(0215870, 'LYON', 'KSENAGISI STON DROMO TOU GALLIKOY KRASIOY ', 'ABROAD', 'GALLIKA',
NULL),

(0222022, 'MONACO', 'PERIHHGSH STHN FIMISMENH POLH THS F1', 'ABROAD', 'GALLIKA',
NULL),

(8895252, 'INSTABULL', 'KSENAGISI STON THRISKEYTIKO XORO TIS POLIS', 'ABROAD',
'TOURKIKA', NULL),

(1145253, 'WARSAW', 'EPISKEPSI SE MOUSEIO TOY 2ou PAGOSMIOY POLEMOU ', 'ABROAD',
'POLONIKA', NULL),

(5274917, 'ZAGOROXVRIA', 'TAKSIDI STA 10 PIO OMORFA XWRIA', 'LOCAL', 'ELLINIKA', NULL),

(1234975, 'ARGENTINA', 'O GYROS THS ARGENTINHS', 'ABROAD', 'ENGLISH', NULL),

(4545678, 'KOYBA', 'ANAKALYPSTE THN OMORFH HAVANA', 'ABROAD', 'SPANISH', NULL),

(8328495, 'LONDON', 'ORGANWMENES EPISKEPSEIS STO BRETANIKO MOYSEIO KAI STON
PYRGO TOY LONDINOY', 'ABROAD','GALLIKA', NULL),

(0102947, 'NAXOS', 'METAFORA STA PARADOSIKA XORIA TIS KRITIS', 'LOCAL', 'ELLINIKA',
NULL),

(2135461, 'MILANO', 'AGORES STHN GALLERIA', 'ABROAD','ITALIKA' , NULL),

(4163264, 'SIKELIA', 'KALOKAIRI STHN OMORFH SIKELIA', 'ABROAD', 'ITALIKA', NULL),

(6743521, 'AMSTERDAM', 'BOLTA STA KANALIA THS POLHS', 'ABROAD', 'OLLANDIKA', NULL),

(3765631, 'NEW YORK', 'CHRISTMAS IN NEW YORK', 'ABROAD', 'AGGLIKA', NULL),

(9348746, 'MALDIVES', 'TAKSIDI TOY MELITOS', 'ABROAD', 'AGGLIKA', NULL),

(1097643, 'SALZBURG', 'EPISKEPSI STO SPITI TOY MOZART', 'ABROAD', 'GERMANIKA', NULL),

(2230973, 'MYKONOS', 'TAKSIDI STO NHSI TON ANEMON', 'LOCAL', 'ELLINIKA', NULL),

(0877622, 'EDINBURGH', 'PERIHHGSH STA HIGHLANDS', 'ABROAD', 'AGGLIKA', NULL),

(1357778, 'PRAGUE', 'H OMORFOTERH MESAIONIKH POLH THS EYROPHS', 'ABROAD',
'TSEXIKA', NULL),

(7544289, 'BERGEN', 'DIAMONH STA MONADIKA FJORDS THS NORBHGIAS', 'ABROAD',
'NORBHGIKA', NULL),

(9088722, 'ROVANIEMI', 'EPISKEPSI STO XORIO TOY AH-BASILH', 'ABROAD', 'FINLANDIKA',
NULL),

(1085667, 'SANTORINI', 'BOLTA ME CATAMARAN STIS PIO OMORFES PARALIES TOY NHSIOY',
'LOCAL', 'ELLINIKA', NULL),

(3330098, 'TEL AVIV', 'KSENAGHSH STH PALIA POLH KAI EPISKEPSH STA IEROSOLYMA',
'ABROAD', 'EBRAIKA', NULL),

(2206676, 'MUNICH', 'DEITE APO KONTA TO KASTRO NEUSCHWANSTEIN TOY BASILIA
LOYDOBIKOY II', 'ABROAD', 'GERMANIKA', NULL),

```
(1000321, 'BRUSSELS', 'DOKIMASTE AYTHENTIKES BELGIKES SOKOLATES', 'ABROAD',  
'GALLIKA', NULL);
```

```
INSERT INTO travel_to VALUES
```

```
(1001101,4589618, '2023-02-05 7:30:00', '2023-02-15 20:15:00' ),  
(1101101, 2579860, '2023-02-10 8:25:00', '2023-02-20 19:37:00'),  
(1101010, 4884915, '2023-04-06 6:50:00', '2023-04-16 17:30:00'),  
(0101011,6943780, '2023-04-15 10:45:00', '2023-04-25 21:00:00'),  
(0011001, 5978483, '2023-05-01 9:55:00', '2023-05-11 22:10:00'),  
(0000001, 0321548, '2023-10-04 7:37:00', '2023-10-14 23:15:00' ),  
(0010110,0215870, '2023-10-20 5:30:00', '2023-10-30 20:55:00'),  
(1111110, 0222022, '2023-11-05 6:25:00', '2023-11-15 21:35:00'),  
(1011100, 8895252, '2023-12-15 8:36:00', '2023-12-25 19:45:00'),  
(0111110, 1145253, '2023-12-20 12:56:00', '2023-12-30 21:40:00'),  
(20910040, 5274917,'2023-06-24 7:30:00', '2023-06-30 20:30:00'),  
(11100110, 5274917,'2023-07-16 8:30:00', '2023-07-26 20:37:00'),  
(10105557,4545678,'2023-08-15 9:30:00', '2023-08-25 21:30:00'),  
(10105558, 8328495,'2023-08-16 9:45:00', '2023-08-22 22:00:00'),  
(10105559,0102947,'2023-05-14 12:00:00', '2023-05-19 21:10:00'),  
(10105560, 2135461,'2023-08-12 7:00:00', '2023-08-17 20:15:00'),  
(10105561, 4163264,'2023-09-24 8:45:00', '2023-09-29 20:00:00'),  
(10105562, 6743521,'2023-09-9 6:55:00', '2023-09-15 21:00:00'),  
(10105563, 3765631,'2023-12-20 10:00:00', '2023-12-28 20:35:00'),  
(10105564, 9348746,'2023-10-02 12:15:00', '2023-10-16 21:30:00'),  
(3900000, 1097643, '2023-01-07 7:00:00', '2023-01-12 21:00:00'),  
(3901001, 2230973, '2023-08-10 10:15:00', '2023-08-15 15:30:00'),  
(3932221, 0877622, '2023-02-18 9:30:00', '2023-02-24 19:00:00'),  
(2099801, 1357778, '2023-01-11 6:30:00', '2023-01-14 16:15:00'),  
(1999990, 7544289, '2023-05-05 9:00:00', '2023-05-11 20:00:00'),  
(8566783, 9088722, '2023-02-04 8:30:00', '2023-02-10 13:30:00'),  
(3458800, 1085667, '2023-07-12 10:00:00', '2023-07-19 18:00:00'),  
(1009870, 3330098, '2023-06-08 5:30:00', '2023-06-13 11:00:00'),  
(1001993, 2206676, '2023-02-01 9:00:00', '2023-02-08 16:15:00'),  
(1766571, 1000321, '2023-09-08 9:15:00', '2023-09-14 17:00:00');
```

```
INSERT INTO reservation VALUES
```

```
(1001101, 3, 'GIORGOS', 'KARAGEORGOPOULOS','ADULT'),
```

(1101101, 12, 'GIANNIS', 'ADAMANTIDIS','ADULT'),
 (1101010, 2, 'PANAGIOTIS', 'GEORGIOU','MINOR'),
 (0101011, 4, 'EIRINI', 'STAMATI','MINOR'),
 (0011001, 14, 'KATERINA', 'PAPADOPOULOU','ADULT'),
 (0000001, 11, 'ADREAS', 'PANAGIOTOU','MINOR'),
 (0010110, 9, 'KOSTADINOS', 'KARAGIANNIS','ADULT'),
 (1111110, 5, 'EVAGGELIA', 'KOLOKOURI','MINOR'),
 (1011100, 10, 'MARIA', 'KOLAGKI','ADULT'),
 (0111110, 5, 'ATHINA', 'PATELI','MINOR'),
 (20910040, 10, 'EVANGELIA', 'LEKKA', 'ADULT'),
 (11100110, 12, 'MARIA', 'ZAXAROPOULOU', 'MINOR'),
 (10105557, 10, 'ELENI', 'MIXALOPOULOU', 'ADULT'),
 (10105558, 9, 'XRYSSA', 'KATSIGIANNI', 'ADULT'),
 (10105559, 5, 'BASILIKI', 'XRYSIKOU', 'MINOR'),
 (10105560, 19, 'XRISTINA', 'TSOUKANELI', 'ADULT'),
 (10105561, 12, 'EIRINI', 'MARINOY', 'MINOR'),
 (10105562, 7, 'NIKOS', 'XIOTIS', 'ADULT'),
 (10105563, 4, 'ORESTIS', 'PANAGOPOULOS', 'ADULT'),
 (10105564, 12, 'MANOLIS', 'STAMATAKIS','MINOR'),
 (3900000, 4, 'DIMITRIS', 'PAYLOPOYLOS','ADULT'),
 (3900000, 7, 'IOANNA', 'PAYLOPOYLOY','ADULT'),
 (3932221, 16, 'MARIA', 'TRIANAFYLLOU','MINOR'),
 (3932221, 17, 'ELENI', 'IOANNIDOU','ADULT'),
 (1999990, 18, 'PETROS', 'ARGYRIOU','ADULT'),
 (8566783, 5, 'BASILHS', 'XONDROGIANNHS','ADULT'),
 (3458800, 4, 'DIMITRA', 'STERGIOU','ADULT'),
 (1009870, 16, 'AGGELIKI', 'KATSIGIANNH','MINOR'),
 (1001993, 37, 'PANAGIOTHS', 'GEORGOPOULOS','MINOR'),
 (1766571, 38, 'PAYLOS', 'ATHANASIOY','ADULT');

```

CREATE TABLE log(
user VARCHAR(50),
action VARCHAR(10),
changes VARCHAR(200),
time timestamp
);

```

A.2 Stored Procedures

A.2.1 Κώδικας Stored Procedures

new_driver():

Καθώς ορίζουμε το procedure περνάμε σαν ορίσματα όλα τα γνωρίσματα που διαθέτουν οι πίνακες *driver* και *worker* τα οποία θα εισάγονται από το χρήστη της βάσης. Αρχικά, ορίζουμε δύο μεταβλητές τύπου INT με όνομα **minworker** και **min_br_code**, στις οποίες αποθηκεύουμε το ελάχιστο πλήθος οδηγών και το υποκατάστημα στο οποίο ανήκουν αυτοί αντίστοιχα. Οι πληροφορίες που ψάχνουμε ανήκουν στους πίνακες *worker*, *driver* και *branch*, οπότε κάνουμε αρχικά inner join τον πίνακα *worker* με τον *branch* στα πεδία **wrk_br_code**, **br_code** και έπειτα inner join τον *driver* με τον *worker* στα πεδία **drv_AT**, **wrk_AT**. Έτσι, δημιουργείται ένας πίνακας που περιέχει όλα τα υποκαταστήματα, τους εργάτες και τους οδηγούς. Από αυτόν τον πίνακα διαλέγουμε μόνο τους εργάτες που είναι οδηγοί, δηλαδή τους εργάτες για τους οποίους ισχύει η συνθήκη '*drv_AT=wrk_AT*', τους οποίους ομαδοποιούμε με βάση το κωδικό του παραρτήματος στο οποίο ανήκουν. Για να βρούμε το πλήθος των οδηγών σε κάθε υποκατάστημα, χρησιμοποιούμε την συναθροιστική συνάρτηση **count(*)**, η οποία μετρά το πλήθος των εγγραφών στο πίνακα που προκύπτει. Για να διαλέξουμε το ελάχιστο πλήθος ταξινομούμε τα αποτελέσματα που επιστρέφει η **count(*)** σε αύξουσα σειρά και διαλέγουμε το πρώτο από αυτά, το οποίο θα είναι όντως και το μικρότερο από όλα τα αποτελέσματα. Έτσι, αποθηκεύουμε την ελάχιστη τιμή του **count(*)** που προέκυψε και τη τιμή του **br_code** που αντιστοιχεί σε αυτή στις μεταβλητές **minworker** και **min_br_code** αντίστοιχα. Στη συνέχεια, εισάγουμε τα στοιχεία του οδηγού που έχει δώσει ο χρήστης στους πίνακες *worker* και *driver* εισάγοντας στο πίνακα *worker* όπου **wrk_br_code** τη τιμή του **min_br_code**.

```
DROP PROCEDURE IF EXISTS new_driver;
DELIMITER $
CREATE PROCEDURE new_driver(IN drv_at CHAR(10),IN name varchar(20), IN lname
VARCHAR(20), IN salary FLOAT(7,2), IN license ENUM('A','B','C','D'), IN route
ENUM('LOCAL','ABROAD'), IN experience TINYINT(4))

BEGIN
DECLARE minworker INT;
DECLARE min_br_code INT(11);

SELECT count(*), br_code INTO minworker, min_br_code
FROM worker INNER JOIN branch ON wrk_br_code = br_code INNER JOIN driver ON
drv_AT=wrk_AT
WHERE drv_AT=wrk_AT GROUP BY wrk_br_code ORDER BY count(*) LIMIT 1;

INSERT INTO worker(wrk_AT,wrk_name,wrk_lname,wrk_salary, wrk_br_code) VALUES(drv_at,
name, lname, salary, min_br_code);
INSERT INTO driver VALUES(drv_at, license, route, experience);

END $
DELIMITER ;
```

find_trip_info():

Καθώς ορίζουμε το procedure περνάμε σαν ορίσματα το id ενός υποκαταστήματος και δύο ημερομηνίες έναρξης και λήξης ταξιδιού. Επειδή ένα υποκατάστημα μπορεί να διαθέτει πολλές κρατήσεις για ταξίδια θα χρησιμοποιήσουμε δομή *cursor*. Όμοια, θα χρησιμοποιήσουμε μια δομή *cursor* που θα επιστρέφει τους οδηγούς για κάθε ταξίδι και άλλη μία που θα επιστρέφει τους ξεναγούς για κάθε ταξίδι. Αρχικά, ορίζουμε μια σημαία *not_found*, την οποία αρχικοποιούμε στη συνέχεια με 0 και τη θέτουμε ίση με 1 όταν δεν υπάρχουν άλλες εγγραφές που πρέπει να μας επιστραφούν από τον πίνακα. Έπειτα, ορίζουμε τις μεταβλητές *departure*, *number_of_res*, *cost*, *driver_name*, *driver_lastname*, *guide_name*, *guide_lastname*, *max_seats*, που αντιπροσωπεύουν τα αντίστοιχα πεδία των πινάκων που θα χρησιμοποιήσουμε, καθώς και τη μεταβλητή *empty_seats*, στην οποία αποθηκεύουμε το πλήθος των κενών θέσεων, δηλαδή τη διαφορά του *max_seats* από το *number_of_res*. Στη συνέχεια, ορίζουμε τον πρώτο *cursor* με όνομα *TR_INFO*, ο οποίος επιστρέφει το πλήθος και τα στοιχεία των ταξιδιών από τους πίνακες *branch*, *trip*, *reservation* (που έχουν ενωθεί σε έναν μέσω του inner join), για τα οποία ισχύει ότι ο κωδικός υποκαταστήματος ισούται με αυτόν που έχει εισάγει ο χρήστης κατά την εκτέλεση του procedure. Ο δεύτερος *cursor* θα είναι ο *WRK_1*, ο οποίος επιστρέφει το ονοματεπώνυμο του οδηγού κάθε ταξιδιού από τους πίνακες *branch*, *trip*, *driver*, *worker* (που έχουν ενωθεί πάλι σε έναν μέσω του inner join), για τον οποίον ισχύει ότι ο κωδικός υποκαταστήματος ισούται με αυτόν που έχει εισάγει ο χρήστης. Όμοια λειτουργεί και ο *cursor* *WRK_2*, ο οποίος αντίστοιχα θα επιστρέφει το ονοματεπώνυμο του ξεναγού του ταξιδιού με βάση το id που έδωσε ο χρήστης. Τέλος, ανοίγουμε τους cursors, τους κάνουμε 'fetch' στις μεταβλητές που αντιστοιχούν στο καθένα και όσο το flag παραμένει 0, δηλαδή όσο υπάρχουν ακόμα εγγραφές που πρέπει να επιστραφούν, αν το εύρος της ημερομηνίας που εισήγαγε ο χρήστης είναι αποδεκτό, τότε εκτυπώνουμε το περιεχόμενο των *cost*, *max_seats*, *number_of_res*, *driver_name*, *driver_lastname*, *guide_name*, *guide_lastname* και *empty_seats*. Αλλιώς, θέτουμε προς εκτύπωση ένα κατάλληλο μήνυμα.

```
DROP PROCEDURE IF EXISTS find_trip_info;
DELIMITER $
CREATE PROCEDURE find_trip_info(IN branch_id int(11), IN Date_1 date, IN Date_2 date)

BEGIN
DECLARE not_found INT ;
DECLARE departure date;
DECLARE number_of_res INT;
DECLARE cost INT(11);
DECLARE driver_name CHAR(25);
DECLARE driver_lastname CHAR(25);
DECLARE guide_name CHAR(25);
DECLARE guide_lastname CHAR(25);
DECLARE max_seats TINYINT(4);
DECLARE empty_seats int;

DECLARE TR_INFO CURSOR FOR
SELECT tr_id, tr_cost, tr_maxseats, tr_departure,count(tr_id)
FROM branch INNER JOIN trip ON br_code=tr_br_code INNER JOIN reservation ON
tr_id=res_tr_id
WHERE br_code=branch_id group by res_tr_id;

DECLARE WRK_1 CURSOR FOR
```

```

SELECT wrk_name, wrk_lname FROM branch INNER JOIN trip ON br_code=tr_br_code INNER
JOIN driver ON tr_drv_AT=drv_AT INNER JOIN worker ON drv_AT=wrk_AT
WHERE br_code=branch_id ;

declare WRK_2 CURSOR FOR
SELECT wrk_name, wrk_lname FROM branch INNER JOIN trip ON br_code=tr_br_code INNER
JOIN guide ON tr_gui_AT=gui_AT INNER JOIN worker ON gui_AT=wrk_AT
WHERE br_code=branch_id ;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET not_found=1;

SET not_found = 0;
OPEN TR_INFO;
OPEN WRK_1;
OPEN WRK_2;

REPEAT
FETCH TR_INFO INTO branch_id, cost, max_seats, departure, number_of_res;
FETCH WRK_1 INTO driver_name, driver_lastname;
FETCH WRK_2 INTO guide_name, guide_lastname;

IF(not_found=0) THEN
    IF Date_1 <= departure AND Date_2 >= departure THEN
        SET empty_seats=max_seats-number_of_res;
        SELECT cost, max_seats, number_of_res, driver_name,
driver_lastname,guide_name, guide_lastname, empty_seats ;
        ELSE SELECT 'NO TRIP FOUND BETWEEN THESE DATES';
        END IF
    END IF;
UNTIL(not_found=1)
END REPEAT;

CLOSE TR_INFO;
CLOSE WRK_1;
CLOSE WRK_2;
END$
DELIMITER ;

```


delete_employee():

Καθώς ορίζουμε το procedure περνάμε σαν ορίσματα το όνομα και το επώνυμο του υπαλλήλου προς διαγραφή. Ορίζουμε τις μεταβλητές **emp_id** και **mng_id**, οι οποίες αντιπροσωπεύουν το id των υπαλλήλων και το id όσων από αυτούς είναι managers. Αρχικά, αποθηκεύουμε τον Α.Τ. του εργαζόμενου στην μεταβλητή **emp_id** και έπειτα από όλους αυτούς διαλέγουμε το **emp_id** αυτού, του οποίου το ονοματεπώνυμο ταιριάζει με τα στοιχεία που έδωσε ο χρήστης και ταυτόχρονα είναι administrator, χωρίς ωστόσο να είναι manager. Η αναζήτηση θα γίνει στους πίνακες *worker*, *admin*, *manages*, οι οποίοι έχουν ενωθεί σε έναν μέσω inner join. Αν υπάρχει τέτοιος εργαζόμενος, τότε μεταβαίνουμε στο πίνακα *worker* και διαγράφουμε τα στοιχεία του εργαζόμενου που έχει Α.Τ. ίσο με το **emp_id** που επιστράφηκε. Σε διαφορετική περίπτωση, αποθηκεύουμε τον Α.Τ στη μεταβλητή **mng_id**, εφόσον μιλάμε για manager, και ακολουθώντας την ίδια διαδικασία με πριν, αν υπάρχει τέτοιος manager του οποίου τα στοιχεία συμφωνούν με αυτά που έδωσε ο χρήστης, εκτυπώνουμε το κατάλληλο μήνυμα ότι ένας manager δεν μπορεί να διαγραφεί.

```
DROP PROCEDURE IF EXISTS delete_employee;
DELIMITER $
CREATE PROCEDURE delete_employee( IN first_name VARCHAR(25), IN last_name
VARCHAR(25))

BEGIN
DECLARE emp_id CHAR(10);
DECLARE mng_id CHAR(10);

SELECT wrk_AT INTO emp_id
FROM worker INNER JOIN admin ON wrk_AT=adm_AT INNER JOIN manages ON
mng_adm_AT=adm_AT
WHERE wrk_name=first_name AND wrk_lname=last_name AND adm_type='ADMINISTRATIVE'
AND mng_adm_AT<>adm_AT;

IF emp_id is NOT NULL THEN
    DELETE FROM worker WHERE wrk_AT=emp_id;
ELSE
    SELECT wrk_AT INTO mng_id
    FROM worker INNER JOIN admin ON wrk_AT=adm_AT INNER JOIN manages ON
mng_adm_AT=adm_AT
    WHERE wrk_name=first_name AND wrk_lname=last_name AND
adm_type='ADMINISTRATIVE' AND mng_adm_AT=adm_AT;

IF mng_id is NOT NULL THEN
    SELECT 'The employee is the manager of a branch and cannot be deleted';
END IF;
END IF;
END $
DELIMITER
```

find_reserv():

Στο συγκεκριμένο ερώτημα εκτελέσαμε την Stored Procedure, η οποία παίρνει σαν είσοδο 2 τιμές που αντιστοιχούν σε δύο τιμές ποσού. Εάν υπάρχουν πελάτες που έχουν δώσει προκαταβολή που η τιμή της βρίσκεται ανάμεσα στις 2 τιμές εισόδου τότε επιστρέφει το όνομα και το επώνυμο του. Φτιάχνουμε έτσι μια δομή *cursor*, η οποία μας επιστρέφει όλες τις προπληρωμές για τη κράτηση για την οποία ισχύει ότι το ποσό προπληρωμής είναι ανάμεσα στα όρια που δήλωσε ο χρήστης.

```
DROP PROCEDURE IF EXISTS find_reserv;
DELIMITER $
CREATE PROCEDURE find_reserv(IN low FLOAT(7,2), IN high FLOAT(7,2))

BEGIN
DECLARE payment FLOAT(7,2);
DECLARE not_found INT;
DECLARE rsrvcursor CURSOR FOR

SELECT res_advance_payment
FROM reservation_offers
WHERE res_advance_payment BETWEEN low AND high;

DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET not_found=1;

SET not_found=0;
OPEN rsrvcursor;
REPEAT
FETCH rsrvcursor INTO payment;
IF(not_found=0) THEN
    SELECT res_advance_payment FROM reservation_offers WHERE res_advance_payment
    BETWEEN low AND high;
END IF;
UNTIL(not_found=1)
END REPEAT;

CLOSE rsrvcursor;
END $
DELIMITER ;
```

find_client_info():

```
DROP PROCEDURE IF EXISTS find_client_info;
DELIMITER $

CREATE PROCEDURE find_client_info(IN client_lname VARCHAR(20))
sp: BEGIN

DECLARE cl_lname VARCHAR(20);
DECLARE counter INT;
DECLARE not_found INT;
DECLARE clcursor CURSOR FOR
SELECT res_offers_lname, COUNT(res_offers_lname)
FROM reservation_offers
WHERE res_offers_lname = client_lname;

DECLARE CONTINUE HANDLER FOR NOT FOUND
SET not_found=1;

SET not_found=0;
OPEN clcursor;
REPEAT
    FETCH clcursor INTO cl_lname, counter;
    IF(not_found=0) THEN
        SELECT offer_id, res_offers_name, res_offers_lname FROM offers INNER JOIN
reservation_offers ON offer_id = offers_res_offer_id WHERE res_offers_lname = cl_lname;
        SELECT COUNT(res_offers_lname) FROM reservation_offers WHERE res_offers_lname =
cl_lname INTO counter;
        IF(counter>1)
            THEN
                SELECT counter AS 'Plithos pelatwn', offer_id FROM offers INNER JOIN
reservation_offers ON offer_id = offers_res_offer_id WHERE res_offers_lname = cl_lname GROUP
BY offer_id;
                LEAVE sp;
            END IF;
        END IF;
    UNTIL(not_found=1)
    END REPEAT;
CLOSE clcursor;
END $
DELIMITER ;
```

A.2.2 Παραδείγματα εκτέλεσης Stored Procedures

3.1.3.1

Καλούμε την procedure και διαλέγουμε να προσθέσουμε τον οδηγό με στοιχεία 'AT10108434', 'AGGELOS', 'FILIS', 1060.00, 'B', 'LOCAL', '13'. Πράγματι, αν μετά την εκτέλεση εκτυπώσουμε τα περιεχόμενα των πινάκων driver, worker παρατηρούμε ότι ο συγκεκριμένος οδηγός έχει προστεθεί και στους δύο πίνακες.

```
mysql> select * from driver;
```

drv_AT	drv_license	drv_route	drv_experience
AB20967621	C	ABROAD	30
AN89980021	A	ABROAD	55
AP19992231	C	LOCAL	27
AS23776510	B	LOCAL	25
AT10108434	B	LOCAL	13
AT24646601	B	ABROAD	79
HK10010120	A	ABROAD	29
IK26285423	B	ABROAD	47
KK10111213	A	LOCAL	24
KO20959000	B	ABROAD	35
MN55776688	D	ABROAD	28
QM98566671	C	LOCAL	26
RY19087762	A	LOCAL	21
TG01020304	B	ABROAD	23
WS12345678	A	ABROAD	22

```
mysql> select * from worker;
```

wrk_AT	wrk_name	wrk_lname	wrk_salary	wrk_br_code
AA09895623	GIORGOS	KARAGIANNIS	2056.00	3
AA34520091	GIORGOS	GIANOPOULOS	1500.00	4
AB15370047	EYTYXIA	KORAKA	1800.00	9
AB20967621	EVA	STERGIOU	1200.00	23
AB55400231	MILTADIS	PANAGOPOULOS	1755.00	2
AB57391132	PANAGIOTIS	LAZAS	1000.00	4
AB88988450	KONSTANTINA	KAFETZI	1800.00	20
AF06219623	ARISTEA	PAPATHANASIOU	1400.00	26
AF45554678	GIORGOS	KAZATZAKIS	3000.00	6
AG12343210	GIANNIS	PAPADOPOULOS	2355.00	6
AI56366671	PANAGIOTA	GOURDOUPI	3100.00	21
AI83950702	MARIA	DWROU	1950.00	12
AI85678002	EYGENEIA	KOYNAVI	1700.00	7
AK12161135	SERGIOS	XARAVGIS	2100.00	6
AM10999737	KATERINA	MANOLOPOULOU	1500.00	6
AM15370047	CHRISTOS	GEORGOPOULOS	1850.00	11
AM15375947	DIONISIS	MANOLOPOULOS	1500.00	11
AM98754326	GIANNIS	PANOY	1000.00	1
AN03060804	EVANGELIA	PATZOPOULOU	1600.00	11
AN89812526	KRISTOFOROS	LAGIOS	1800.00	16
AN89980021	PETROS	PANAGIOTOU	1200.00	26
AP19992231	KOSTAS	KATSAROS	2110.00	28
AR23678738	FILLIPOS	ASTERIETHS	1110.00	4
AR98889078	ANDRIANNA	KARAKWSTA	980.00	5
AS23776510	EIRINI	ZERVA	1300.00	24
AS23785450	OLYMPIA	ALEXANDROPOULOU	1520.00	20
AT10108434	AGGELOS	FILIS	1060.00	22
AT24646601	ORESTIS	GEORGIADIS	1200.00	7
AT42323457	OLGA	THEODOROPOULOU	1650.00	8
AT49862157	MANOLIS	MANESIS	1000.00	12
AW12161135	BASILIKI	ANTWNIOY	2000.00	12
AY04554865	NIKOLAOS	KWNSTANTINOY	1700.00	19

3.1.3.2

Καλούμε την procedure για το υποκατάστημα με κωδικό 10 και εύρος ημερομηνιών από '2023-08-10' έως '2023-11-20' και εμφανίζονται τα στοιχεία για τα τρία διαφορετικά ταξίδια που παρέχει το συγκεκριμένο παράρτημα. Αντίθετα, αν τη καλέσουμε για το παράρτημα με κωδικό 9 και με ημερομηνίες '2022-03-04' και '2022-04-14' βλέπουμε ότι εκτυπώνεται το αντίστοιχο μήνυμα μη ύπαρξης ταξιδιού.

```
mysql> DELIMITER ;
mysql>
mysql> call find_trip_info(10, '2023-08-10', '2023-11-20');
+-----+-----+-----+-----+-----+-----+-----+-----+
| cost | max_seats | number_of_res | driver_name | driver_lastname | guide_name | guide_lastname | empty_seats |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 8700 | 48 | 1 | KOSTAS | KATSAROS | GIANNIS | PAPADOPOULOS | 47 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.06 sec)

+-----+-----+-----+-----+-----+-----+-----+-----+
| cost | max_seats | number_of_res | driver_name | driver_lastname | guide_name | guide_lastname | empty_seats |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 450 | 35 | 1 | KOSTAS | KATSAROS | GIANNIS | PAPADOPOULOS | 34 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.07 sec)

+-----+-----+-----+-----+-----+-----+-----+-----+
| cost | max_seats | number_of_res | driver_name | driver_lastname | guide_name | guide_lastname | empty_seats |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 220 | 35 | 1 | KOSTAS | KATSAROS | GIANNIS | PAPADOPOULOS | 34 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.08 sec)

Query OK, 0 rows affected (0.10 sec)

mysql> call find_trip_info(9, '2022-03-04', '2022-04-14');
+-----+-----+
| NO TRIP FOUND BETWEEN THESE DATES |
+-----+-----+
| NO TRIP FOUND BETWEEN THESE DATES |
+-----+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> select * from trip where tr_br_code=10;
+-----+-----+-----+-----+-----+-----+-----+-----+
| tr_id | tr_departure | tr_return | tr_maxseats | tr_cost | tr_br_code | tr_gui_AT | tr_drv_AT |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 10110 | 2023-10-20 05:30:00 | 2023-10-30 20:55:00 | 48 | 8700.00 | 10 | AG12343210 | AP19992231 |
| 10910910 | 2023-10-20 00:00:00 | 2023-10-30 00:00:00 | 35 | 450.00 | 10 | AG12343210 | AP19992231 |
| 10910911 | 2023-10-02 00:00:00 | 2023-10-06 00:00:00 | 35 | 220.00 | 10 | AG12343210 | AP19992231 |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

3.1.3.3

Καλούμε την procedure για τον υπάλληλο με όνομα 'GIANNIS' και επώνυμο 'PANOY' και βλέπουμε ότι εκτυπώνεται το μήνυμα σφάλματος, καθώς ο συγκεκριμένος υπάλληλος είναι manager:

```
mysql> DROP PROCEDURE IF EXISTS delete_employee;
Query OK, 0 rows affected (0.02 sec)

mysql> DELIMITER $
mysql> CREATE PROCEDURE delete_employee( IN first_name VARCHAR(25), IN last_name VARCHAR(25))
  -> BEGIN
  ->
  -> DECLARE emp_id CHAR(10);
  -> DECLARE mng_id CHAR(10);
  ->
  -> SELECT wrk_AT INTO emp_id FROM worker
  -> INNER JOIN admin ON wrk_AT=adm_AT
  -> INNER JOIN manages ON mng_adm_AT=adm_AT
  -> WHERE wrk_name=first_name AND wrk_lname=last_name AND adm_type='ADMINISTRATIVE' AND mng_adm_AT<>adm_AT;
  ->
  -> IF emp_id is NOT NULL THEN
  ->   DELETE FROM worker WHERE wrk_AT=emp_id;
  -> ELSE
  ->   SELECT wrk_AT INTO mng_id FROM worker
  ->   INNER JOIN admin ON wrk_AT=adm_AT
  ->   INNER JOIN manages ON mng_adm_AT=adm_AT
  ->   WHERE wrk_name=first_name AND wrk_lname=last_name AND adm_type='ADMINISTRATIVE' AND mng_adm_AT=adm_AT;
  ->
  -> IF mng_id is NOT NULL THEN
  ->   SELECT 'The employee is the manager of a branch and cannot be deleted';
  -> END IF;
  -> END IF;
  -> END $
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql>
mysql> call delete_employee('GIANNIS', 'PANOY');
+-----+
| The employee is the manager of a branch and cannot be deleted |
+-----+
| The employee is the manager of a branch and cannot be deleted |
+-----+
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.03 sec)
```

Καλούμε την procedure `find_reserv()` για τιμές εισόδου 51.00 και 52.00:



```
CREATE INDEX indexres
ON reservation_offers (res_offers_name, res_offers_lname,res_advance_payment);
```

[illegible]

A.3 Triggers

A.3.1 Κώδικας Triggers

Για κάθε πίνακα από τους *trip*, *reservation*, *event*, *travel_to* και *destination* ορίζουμε τρεις διαφορετικούς triggers, έναν για καθεμία επιτρεπτή ενέργεια ('INSERT', 'UPDATE', 'DELETE'), ο καθένας από τους οποίους τίθεται σε ισχύ κάθε φορά που ο **IT** εκτελεί μια από αυτές τις ενέργειες.

Για κάθε έναν από αυτούς τους triggers δουλεύουμε με τον ίδιο τρόπο, δηλαδή ορίζουμε μετά από ποια ενέργεια θα ενεργοποιηθεί και στη συνέχεια αφού ενεργοποιηθεί τον θέτουμε να κάνει εισαγωγή στον πίνακα **log** στα πεδία **user**, **action**, **changes**, **time** το όνομα του χρήστη (μέσω της συνάρτησης **current_user()**), τη ενέργεια που έγινε σε μορφή αλφαριθμητικού, τις αλλαγές και την στιγμή που έγιναν οι αλλαγές (μέσω της συνάρτησης **timestamp()**) αντίστοιχα. Ακόμη, ορίζουμε και τη μεταβλητή **changes**, στην οποία μέσω της συνάρτησης **concat()** ενώνουμε όλες τις τιμές του πίνακα που επηρεάστηκαν και τις θέτουμε να χωρίζονται με κόμμα.

Trigger με την ενημέρωση του πίνακα trip:

```
DROP TRIGGER tr_trip_after_update;

DELIMITER $

CREATE TRIGGER tr_trip_after_update
AFTER UPDATE ON trip
FOR EACH ROW
BEGIN
    DECLARE changes VARCHAR(200);
    SET changes=CONCAT(NEW.tr_id, ',' ,
NEW.tr_departure, ',' ,
NEW.tr_return, ',' ,
NEW.tr_maxseats, ',' ,
NEW.tr_cost, ',' ,
NEW.tr_br_code, ',' ,
NEW.tr_gui_AT, ',' ,
NEW.tr_drv_AT);
INSERT INTO log VALUES(CURRENT_USER(),'update to trip' ,changes ,current_timestamp());
END$

DELIMITER ;
```


Trigger με την ενημέρωση του πίνακα reservation:

```
DROP TRIGGER reservation_update;

DELIMITER $

CREATE TRIGGER reservation_update

AFTER UPDATE ON reservation

FOR EACH ROW

BEGIN

    DECLARE changes VARCHAR(200);

    SET changes=CONCAT(NEW.res_tr_id, ',' ,

                        NEW.res_seatnum, ',' ,

                        NEW.res_name, ',' ,

                        NEW.res_lname, ',' ,

                        NEW.res_isadult);

    INSERT INTO log VALUES(CURRENT_USER(), 'update to reservation' ,changes

, current_timestamp());

END$

DELIMITER ;
```

Trigger με την ενημέρωση του πίνακα destination:

```
DROP TRIGGER tr_destination_after_update;

DELIMITER $

CREATE TRIGGER tr_destination_after_update

AFTER UPDATE ON destination

FOR EACH ROW

BEGIN

    DECLARE changes VARCHAR(200);

    SET changes=CONCAT(NEW.dst_id, ',' ,

                        NEW.dst_name , ',' ,

                        NEW.dst_descr , ',' ,
```

```

NEW.dst_rtype , ',' ,
NEW.dst_language , ',' ,
NEW.dst_location);

INSERT INTO log VALUES(CURRENT_USER(),'update to destination' ,changes
,current_timestamp());

END$

DELIMITER ;

```

Trigger με την ενημέρωση του πίνακα travel_to:

```

DROP TRIGGER tr_travel_to_after_update;

DELIMITER $

CREATE TRIGGER tr_travel_to_after_update
AFTER UPDATE ON travel_to
FOR EACH ROW
BEGIN
    DECLARE changes VARCHAR(200);

    SET changes=CONCAT(NEW.to_tr_id, ',' ,
                        NEW.to_dst_id, ',' ,
                        NEW.to_arrival, ',' ,
                        NEW.to_departure);

    INSERT INTO log VALUES(CURRENT_USER(),'update to travel_to' ,changes
,current_timestamp());

END$

DELIMITER ;

```

Trigger με την ενημέρωση του πίνακα event:

```

DROP TRIGGER tr_event_after_update;

DELIMITER $

CREATE TRIGGER tr_event_after_update

```

```

AFTER UPDATE ON event
FOR EACH ROW
    BEGIN
        DECLARE changes VARCHAR(200);
        SET changes=CONCAT(NEW.ev_tr_id, ',' ,
                                NEW.ev_start, ',' ,
                                NEW.ev_end, ',' ,
                                NEW.ev_descr);
        INSERT INTO log VALUES(CURRENT_USER(),'update to event' ,changes
,current_timestamp());
    END$
DELIMITER ;

```

Trigger με την διαγραφή του πίνακα trip:

```

DROP TRIGGER tr_trip_after_delete;
DELIMITER $
CREATE TRIGGER tr_trip_after_delete
AFTER DELETE ON trip
FOR EACH ROW
    BEGIN
        DECLARE changes VARCHAR(200);
        SET changes=CONCAT(OLD.tr_id, ',' ,
OLD.tr_departure, ',' ,
OLD.tr_return, ',' ,
OLD.tr_maxseats, ',' ,
OLD.tr_cost, ',' ,
OLD.tr_br_code, ',' ,
OLD.tr_gui_AT, ',' ,
OLD.tr_drv_AT);
    END$
DELIMITER ;

```

```

INSERT INTO log VALUES(CURRENT_USER(),'delete to trip' ,changes
,current_timestamp());

END$

DELIMITER ;

```

Trigger με την διαγραφή του πίνακα destination:

```

DROP TRIGGER tr_destination_after_delete;

DELIMITER $

CREATE TRIGGER tr_destination_after_delete
AFTER DELETE ON destination
FOR EACH ROW
BEGIN
    DECLARE changes VARCHAR(200);
    SET changes=CONCAT(OLD.dst_id, ',' ,
OLD.dst_name, ',' ,
OLD.dst_descr, ',' ,
OLD.dst_rtype, ',' ,
OLD.dst_language, ',' ,
OLD.dst_location);

INSERT INTO log VALUES(CURRENT_USER(),'delete to destination' ,changes
,current_timestamp());

END$

DELIMITER ;

```

Trigger με την διαγραφή του πίνακα travel_to:

```

DROP TRIGGER tr_travel_to_after_delete;

DELIMITER $

CREATE TRIGGER tr_travel_to_after_delete
AFTER DELETE ON travel_to

```

```

FOR EACH ROW
    BEGIN
        DECLARE changes VARCHAR(200);
        SET changes=CONCAT(OLD.to_tr_id, ',' ,
OLD.to_dst_id, ',' ,
OLD.to_arrival, ',' ,
OLD.to_departure);
INSERT INTO log VALUES(CURRENT_USER(),'delete to travel_to' ,changes ,current_timestamp());
END$
DELIMITER ;

```

Trigger με την διαγραφή του πίνακα event:

```

DROP TRIGGER tr_event_after_delete;
DELIMITER $
CREATE TRIGGER tr_event_after_delete
AFTER DELETE ON event
FOR EACH ROW
    BEGIN
        DECLARE changes VARCHAR(200);
        SET changes=CONCAT(OLD.ev_tr_id, ',' ,
OLD.ev_start, ',' ,
OLD.ev_end, ',' ,
OLD.ev_descr);
INSERT INTO log VALUES(CURRENT_USER(),'delete to event' ,changes ,current_timestamp());
END$
DELIMITER ;

```

Trigger με την διαγραφή του πίνακα reservation:

```
DROP TRIGGER tr_reservation_after_delete;

DELIMITER $

CREATE TRIGGER tr_reservation_after_delete

AFTER DELETE ON reservation

FOR EACH ROW

BEGIN

    DECLARE changes VARCHAR(200);

    SET changes=CONCAT(OLD.res_tr_id , ',' ,

OLD.res_seatnum , ',' ,

OLD.res_name , ',' ,

OLD.res_lname , ',' ,

OLD.res_isadult);

INSERT INTO log VALUES(CURRENT_USER(),'update to reservation' ,changes

,current_timestamp());

END$

DELIMITER ;
```

Trigger με την εισαγωγή στον πίνακα trip:

```
DROP TRIGGER tr_trip_after_update;

DELIMITER $

CREATE TRIGGER tr_trip_after_update

AFTER INSERT ON trip

FOR EACH ROW

BEGIN

    DECLARE changes VARCHAR(200);

    SET changes=CONCAT(NEW.tr_id, ',' ,

NEW.tr_departure, ',' ,

NEW.tr_return, ',' ,
```

```

NEW.tr_maxseats, ',' ,
NEW.tr_cost, ',' ,
NEW.tr_br_code, ',' ,
NEW.tr_gui_AT, ',' ,
NEW.tr_drv_AT);

INSERT INTO log VALUES(CURRENT_USER(),'update to trip' ,changes ,current_timestamp());

END$

DELIMITER ;

```

Trigger με την εισαγωγή στον πίνακα reservation:

```

DROP TRIGGER reservation_update;

DELIMITER $

CREATE TRIGGER reservation_update
AFTER INSERT ON reservation
FOR EACH ROW
BEGIN
    DECLARE changes VARCHAR(200);
    SET changes=CONCAT(NEW.res_tr_id, ',' ,
NEW.res_seatnum, ',' ,
NEW.res_name, ',' ,
NEW.res_lname, ',' ,
NEW.res_isadult);

INSERT INTO log VALUES(CURRENT_USER(),'update to reservation' ,changes ,current_timestamp());

END$

DELIMITER ;

```

Trigger με την εισαγωγή στον πίνακα destination:

```
DROP TRIGGER tr_destination_after_update;

DELIMITER $

CREATE TRIGGER tr_destination_after_update
AFTER INSERT ON destination
FOR EACH ROW
BEGIN
    DECLARE changes VARCHAR(200);
    SET changes=CONCAT(NEW.dst_id, ',' ,
NEW.dst_name , ',' ,
NEW.dst_descr , ',' ,
NEW.dst_rtype , ',' ,
NEW.dst_language , ',' ,
NEW.dst_location);
INSERT INTO log VALUES(CURRENT_USER(),'update to destination' ,changes
,current_timestamp());
END$

DELIMITER ;
```

Trigger με την εισαγωγή στον πίνακα travel_to:

```
DROP TRIGGER tr_travel_to_after_update;

DELIMITER $

CREATE TRIGGER tr_travel_to_after_update
AFTER INSERT ON travel_to
FOR EACH ROW
BEGIN
    DECLARE changes VARCHAR(200);
    SET changes=CONCAT(NEW.to_tr_id, ',' ,
NEW.to_dst_id, ',' ,
```



```

NEW.to_arrival, ',' ,
NEW.to_departure);
INSERT INTO log VALUES(CURRENT_USER(),'update to travel_to' ,changes ,current_timestamp());
END$
DELIMITER ;

```

Trigger με την εισαγωγή στον πίνακα event:

```

DROP TRIGGER tr_event_after_update;
DELIMITER $
CREATE TRIGGER tr_event_after_update
AFTER INSERT ON event
FOR EACH ROW
BEGIN
    DECLARE changes VARCHAR(200);
    SET changes=CONCAT(NEW.ev_tr_id, ',' ,
NEW.ev_start, ',' ,
NEW.ev_end, ',' ,
NEW.ev_descr);
INSERT INTO log VALUES(CURRENT_USER(),'update to event' ,changes ,current_timestamp());
END$
DELIMITER ;

```

dont_change_info():

Στο συγκεκριμένο ερώτημα εκτελέσαμε ένα before update on trip trigger το οποίο μετράει το πλήθος των κρατήσεων των trips και αν αυτό είναι μεγαλύτερο του 0, δηλαδή υπάρχουν κρατήσεις θα αποτρέπει την αλλαγή των πεδίων της ημερομηνίας αναχώρησης, ημερομηνίας επιστροφής και του κόστους.

```
drop trigger if exists dont_change_info;

delimiter $

create trigger dont_change_info
before update on trip
for each row
begin
DECLARE reservationCount INT;

    SET reservationCount=0;

    SELECT COUNT(*) INTO reservationCount
    FROM reservation
    INNER JOIN trip ON tr_id=res_tr_id
    WHERE res_tr_id=tr_id;

    if reservationCount>0 then set
new.tr_departure=old.tr_departure,
new.tr_return=old.tr_return,
new.tr_cost=old.tr_cost;
    end if;

end $

delimiter ;
```

salary_reduction():

Σε αυτό το ερώτημα πριν εισάγουμε στον πίνακα *worker* ελέγχουμε αν ο νέος μισθός που θέλουμε να εισάγουμε είναι μικρότερος από τον υπάρχοντα. Αν ισχύει αυτός ο ισχυρισμός θέτουμε τον νέο μισθό ίδιο με τον παλαιό.

```
drop trigger if exists salary_reduction;

delimiter $

create trigger salary_reduction
before update on worker
for each row
begin
if new.wrk_salary<old.wrk_salary
then
set new.wrk_salary=old.wrk_salary;
end if;
end $

delimiter ;
```

A.3.2 Παραδείγματα εκτέλεσης Triggers

3.1.4.1

Παρακάτω παραθέτουμε screenshots για διάφορες τυχαίες ενέργειες που κάναμε στους πίνακες και πως οι triggers ενημέρωσαν τον πίνακα *log*:

```
mysql> INSERT INTO travel_to VALUES(3932221,4589618, '2023-05-11 8:30:00', '2023-05-21 21:10:00');
Query OK, 1 row affected (0.28 sec)

mysql> select * from log;
+-----+-----+-----+-----+
| user      | action          | changes                                     | time                |
+-----+-----+-----+-----+
| root@localhost | insert to reservation | 1001993,30,PANAGIOTHIS,KARAS,MINOR      | 2023-01-04 15:36:12 |
| root@localhost | update to trip       | NULL                                     | 2023-01-04 17:30:10 |
| root@localhost | update to trip       | NULL                                     | 2023-01-04 17:31:22 |
| root@localhost | update to event      | 10105557,2023-08-20 11:30:00,2023-08-20 12:30:00,EPIISKEPSI STO MOYSEIO THS EPANASTASHS | 2023-01-16 02:00:16 |
| root@localhost | update to trip       | 1001101,2023-02-05 07:30:00,2023-02-15 20:15:00,10,4000.00,1,AA09895623,RV19087762 | 2023-02-04 19:01:52 |
| root@localhost | update to trip       | 1001101,2023-02-05 07:30:00,2023-02-15 20:15:00,10,4000.00,1,AA09895623,RV19087762 | 2023-02-04 19:02:56 |
| root@localhost | insert to travel_to  | 3932221,4589618,2023-05-11 08:30:00,2023-05-21 21:10:00 | 2023-02-04 19:06:45 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> update reservation set res_name='GIANNIS' where res_tr_id=1001101;
Query OK, 1 row affected (0.22 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from log;
+-----+-----+-----+-----+
| user      | action          | changes                                     | time                |
+-----+-----+-----+-----+
| root@localhost | insert to reservation | 1001993,30,PANAGIOTHIS,KARAS,MINOR      | 2023-01-04 15:36:12 |
| root@localhost | update to trip       | NULL                                     | 2023-01-04 17:30:10 |
| root@localhost | update to trip       | NULL                                     | 2023-01-04 17:31:22 |
| root@localhost | update to event      | 10105557,2023-08-20 11:30:00,2023-08-20 12:30:00,EPIISKEPSI STO MOYSEIO THS EPANASTASHS | 2023-01-16 02:00:16 |
| root@localhost | update to trip       | 1001101,2023-02-05 07:30:00,2023-02-15 20:15:00,10,4000.00,1,AA09895623,RV19087762 | 2023-02-04 19:01:52 |
| root@localhost | update to trip       | 1001101,2023-02-05 07:30:00,2023-02-15 20:15:00,10,4000.00,1,AA09895623,RV19087762 | 2023-02-04 19:02:56 |
| root@localhost | insert to travel_to  | 3932221,4589618,2023-05-11 08:30:00,2023-05-21 21:10:00 | 2023-02-04 19:06:45 |
| root@localhost | update to reservation | 1001101,3,GIANNIS,KARAGEORGOPOULOS,ADULT | 2023-02-04 19:11:25 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> delete from destination where dst_id=1000321;
Query OK, 1 row affected (0.79 sec)

mysql> select * from log;
+-----+-----+-----+-----+
| user      | action          | changes                                     | time                |
+-----+-----+-----+-----+
| root@localhost | insert to reservation | 1001993,30,PANAGIOTHIS,KARAS,MINOR      | 2023-01-04 15:36:12 |
| root@localhost | update to trip       | NULL                                     | 2023-01-04 17:30:10 |
| root@localhost | update to trip       | NULL                                     | 2023-01-04 17:31:22 |
| root@localhost | update to event      | 10105557,2023-08-20 11:30:00,2023-08-20 12:30:00,EPIISKEPSI STO MOYSEIO THS EPANASTASHS | 2023-01-16 02:00:16 |
| root@localhost | update to trip       | 1001101,2023-02-05 07:30:00,2023-02-15 20:15:00,10,4000.00,1,AA09895623,RV19087762 | 2023-02-04 19:01:52 |
| root@localhost | update to trip       | 1001101,2023-02-05 07:30:00,2023-02-15 20:15:00,10,4000.00,1,AA09895623,RV19087762 | 2023-02-04 19:02:56 |
| root@localhost | insert to travel_to  | 3932221,4589618,2023-05-11 08:30:00,2023-05-21 21:10:00 | 2023-02-04 19:06:45 |
| root@localhost | update to reservation | 1001101,3,GIANNIS,KARAGEORGOPOULOS,ADULT | 2023-02-04 19:11:25 |
| root@localhost | delete to destination | NULL                                     | 2023-02-04 19:13:37 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

3.1.4.2

Αν καλέσουμε το παραπάνω trigger για το ταξίδι με `tr_id=1001101` για το οποίο υπάρχουν κρατήσεις όπως φαίνεται παρακάτω (τελευταία γραμμή του πίνακα `reservation`):

```
mysql> SELECT * FROM TRIP;
```

tr_id	tr_departure	tr_return	tr_maxseats	tr_cost	tr_br_code	tr_gui_AT	tr_drv_AT
1	2023-10-04 07:37:00	2023-10-14 23:15:00	35	5900.00	9	AG12343210	QM98566671
10110	2023-10-20 05:30:00	2023-10-30 20:55:00	48	8700.00	10	AG12343210	AP19992231
11001	2023-05-01 09:55:00	2023-05-11 22:10:00	45	4120.00	7	AF45554678	AS23776510
101011	2023-04-15 10:45:00	2023-04-25 21:00:00	40	6890.00	5	AB55400231	KK10111213
111110	2023-12-20 12:56:00	2023-12-30 21:40:00	70	20000.00	8	DE34554324	AB20967621
1001101	2023-02-05 07:30:00	2023-02-15 20:15:00	30	4000.00	1	AA09895623	RY19087762
1001993	2023-02-01 09:00:00	2023-02-08 16:15:00	40	2000.00	28	GH96986576	HK10010120
1009870	2023-06-08 05:30:00	2023-06-13 11:00:00	25	1950.00	27	ER23556438	AB20967621
1011100	2023-12-15 08:36:00	2023-12-25 19:45:00	70	15000.00	6	ER23556438	HK10010120
1101010	2023-04-06 06:50:00	2023-04-16 17:30:00	33	5475.00	4	AB57391132	TG01020304
1101101	2023-02-10 08:25:00	2023-02-20 19:37:00	25	6375.00	2	AB57391132	WS12345678
1111110	2023-11-05 06:25:00	2023-11-15 21:35:00	40	5700.00	3	IU09334567	MN55776688
1766571	2023-09-08 09:15:00	2023-09-14 17:00:00	40	2800.00	30	AA09895623	AT24646601
1999990	2023-05-05 09:00:00	2023-05-11 20:00:00	35	5000.00	23	AB55400231	MN55776688
2099801	2023-01-11 06:30:00	2023-01-14 16:15:00	20	3150.50	29	AA34520091	AN89980021
3458800	2023-07-12 10:00:00	2023-07-19 18:00:00	30	1300.00	24	ER23556438	QM98566671
3900000	2023-01-07 07:00:00	2023-01-12 21:00:00	50	3000.00	21	AA09895623	KK10111213
3901001	2023-08-10 10:15:00	2023-08-15 15:30:00	40	1500.00	22	AB57391132	RY19087762
3932221	2023-02-18 09:30:00	2023-02-24 19:00:00	30	2400.00	26	AF45554678	AT24646601
7777777	2025-02-02 07:00:00	2025-02-22 05:00:00	30	2000.00	1	AG12343210	TG01020304
8566783	2023-02-04 08:30:00	2023-02-10 13:30:00	50	4700.10	25	DE34554324	RY19087762
10105557	2023-08-15 09:30:00	2023-08-25 21:30:00	20	2000.00	14	AA09895623	WS12345678
10105558	2023-08-16 09:45:00	2023-08-22 22:00:00	20	990.00	15	DE34554324	AP19992231
10105559	2023-05-14 12:00:00	2023-05-19 21:10:00	15	1600.00	17	ER23556438	KK10111213
10105560	2023-08-12 07:00:00	2023-08-17 20:15:00	30	2100.00	19	AA34520091	AS23776510
10105561	2023-09-24 08:45:00	2023-09-29 20:00:00	30	1000.00	20	DE34554324	QM98566671
10105562	2023-09-09 06:55:00	2023-09-15 21:00:00	10	1300.00	13	AF45554678	K020959000
10105563	2023-12-20 10:00:00	2023-12-28 20:35:00	50	4000.00	16	ER23556438	MN55776688
10105564	2023-10-02 12:15:00	2023-10-16 21:30:00	30	2600.00	18	IU09334567	HK10010120
10910910	2023-10-20 00:00:00	2023-10-30 00:00:00	35	450.00	10	AG12343210	AP19992231
10910911	2023-10-02 00:00:00	2023-10-06 00:00:00	35	220.00	10	AG12343210	AP19992231
11100110	2023-07-16 08:30:00	2023-07-26 20:37:00	20	1550.00	12	AG12343210	RY19087762
20910040	2023-06-24 07:30:00	2023-06-30 20:30:00	35	1000.00	11	DE34554324	TG01020304

33 rows in set (0.00 sec)

```
mysql> SELECT * FROM RESERVATION;
```

res_tr_id	res_seatnum	res_name	res_lname	res_isadult
1	11	ADREAS	PANAGIOTOU	ADULT
10110	9	KOSTADINOS	KARAGIANNIS	ADULT
11001	14	KATERINA	PAPADOPOULOU	ADULT
101011	4	EIRINI	STAMATI	MINOR
111110	5	ATHINA	PATELI	MINOR
1001101	50	GIORGOS	KARAGEORGOPOULOS	ADULT
1001993	27	PANAGIOTIS	GEORGIOPOULOS	MINOR

Αν καλέσουμε την εντολή: **update trip set tr_cost=80000 where tr_id=1001101;** στην οποία ζητάμε να αλλάξει το κόστος του ταξιδιού με 80.000, τότε θα παρατηρήσουμε ότι δεν υπέστη καμία αλλαγή και η τιμή παρέμεινε όπως πριν 4000.00:

```
mysql> delimiter $
mysql> create trigger dont_change_info
-> before update on trip
-> for each row
-> begin
-> DECLARE reservationCount INT;
-> SET reservationCount=0;
->
-> SELECT COUNT(*) INTO reservationCount
-> FROM reservation
-> INNER JOIN trip ON tr_id=res_tr_id
-> WHERE res_tr_id=tr_id;
->
-> if reservationCount>0 then set
-> new.tr_departure=old.tr_departure,
-> new.tr_return=old.tr_return,
-> new.tr_cost=old.tr_cost;
-> end if;
-> end $
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> delimiter ;
mysql> update trip set tr_cost=80000 where tr_id=1001101;
Query OK, 0 rows affected (0.01 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> select * from trip where tr_id=1001101;
```

tr_id	tr_departure	tr_return	tr_maxseats	tr_cost	tr_br_code	tr_gui_AT	tr_drv_AT
1001101	2023-02-05 07:30:00	2023-02-15 20:15:00	30	4000.00	1	AA09895623	RY19087762

1 row in set (0.00 sec)

3.1.4.3

Παρακάτω φαίνεται παράδειγμα όπου αποτρέπεται η μείωση του μισθού του πρώτου εργαζομένου από τον πίνακα worker με wrk_AT=AA09895623 και μισθό 2056.00:

```
mysql> select * from worker;
```

wrk_AT	wrk_name	wrk_lname	wrk_salary	wrk_br_code
AA09895623	GIORGOS	KARAGIANNIS	2056.00	3
AA34520091	GIORGOS	GIANOPOULOS	1500.00	4
AB15370047	EVTYXIA	KORAKA	1800.00	9
AB20967621	EVA	STERGIOU	1200.00	23
AB55400231	MILTIADIS	PANAGOPOULOS	1755.00	2
AB57391132	PANAGIOTIS	LAZAS	1000.00	4
AB88988450	KONSTANTINA	KAFETZI	1800.00	20
AF06219623	ARISTEA	PAPATHANASIOU	1400.00	26
AF45554678	GIORGOS	KAZATZAKIS	3000.00	6
AG12343210	GIANNIS	PAPADOPOULOS	2355.00	6
AI56366671	PANAGIOTA	GOURDOUPI	3100.00	21
AI83950702	MARIA	DWROU	1950.00	12
AI85678002	EYGENEIA	KOYNAVI	1700.00	7
AK12161135	SERGIOS	XARAVGIS	2100.00	6
AM10999737	KATERINA	MANOLOPOULOU	1500.00	6
AM15370047	CHRISTOS	GEORGOPOULOS	1850.00	11
AM15375947	DIONISIS	MANOLOPOULOS	1500.00	11
AM98754326	GIANNIS	PANOY	1000.00	1
AN03060804	EVANGELIA	PATZOPOULOU	1600.00	11
AN89812526	XRISTOFOROS	LAGIOS	1800.00	16
AN89980021	PETROS	PANAGIOTOU	1200.00	26
AP19992231	KOSTAS	KATSAROS	2110.00	28
AR23678738	FILLIPOS	ASTERIWTHS	1110.00	4
AR98889078	ANDRIANNA	KARAKWSTA	980.00	5
AS23776510	EIRINI	ZERVA	1300.00	24
AS23785450	OLYMPIA	ALEXANDROPOULOU	1320.00	20
AT10108434	AGGELOS	FILIS	1060.00	22
AT24646601	ORESTIS	GEORGIADIS	1200.00	7
AT42323457	OLGA	THEODOROPOULOU	1650.00	8
AT49862157	MANOLIS	MANESIS	1000.00	12
AW12161135	BASILIKI	ANTWNIOY	2000.00	12
AY04554865	NIKOLAOS	KWNSTANTINOU	1700.00	19
DE34554324	XRISTOS	TZAMIS	3556.00	9
ER23556438	ORESTHS	PAPAGIANNIS	1115.00	9
FG88555450	GEORGIA	HLIOPOULOU	1890.00	15
WS12345678	ELENI	GEORGOPPOULOU	2100.00	22
YA04554865	AFRODITI	LJANTOY	1770.00	14
ZZZZZZZZ	MARIA	AAAAA	1060.00	21

```
68 rows in set (0.00 sec)
```

```
mysql> drop trigger if exists salary_reduction;
```

```
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> delimiter $
```

```
mysql> create trigger salary_reduction
```

```
  -> before update on worker
```

```
  -> for each row
```

```
  -> begin
```

```
  -> if new.wr_k_salary<old.wr_k_salary
```

```
  -> then
```

```
  -> set new.wr_k_salary=old.wr_k_salary;
```

```
  -> end if;
```

```
  -> end $
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> delimiter ;
```

```
mysql> update worker set wrk_salary=500 where wrk_AT='AA09895623';
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
Rows matched: 1  Changed: 0  Warnings: 0
```

```
mysql> select * from worker;
```

wrk_AT	wrk_name	wrk_lname	wrk_salary	wrk_br_code
AA09895623	GIORGOS	KARAGIANNIS	2056.00	3
AA34520091	GIORGOS	GIANOPOULOS	1500.00	4
AB15370047	EVTYXIA	KORAKA	1800.00	9
AB20967621	EVA	STERGIOU	1200.00	23
AB55400231	MILTIADIS	PANAGOPOULOS	1755.00	2
AB57391132	PANAGIOTIS	LAZAS	1000.00	4
AB88988450	KONSTANTINA	KAFETZI	1800.00	20
AF06219623	ARISTEA	PAPATHANASIOU	1400.00	26
AF45554678	GIORGOS	KAZATZAKIS	3000.00	6
AG12343210	GIANNIS	PAPADOPOULOS	2355.00	6
AI56366671	PANAGIOTA	GOURDOUPI	3100.00	21
AI83950702	MARIA	DWROU	1950.00	12

B.1 Κατασκευή γραφικής διεπαφής – Επικοινωνία με ΒΔ

B.1.1 Κώδικας διεπαφής σε Java – JDBC

Προκειμένου να καταφέρουμε να υλοποιήσουμε τις διεπαφές που μας ζητήθηκαν στο δεύτερο μέρος του project μας , αρχικά χρειάστηκε να συνδέσουμε την υπάρχουσα βάση δεδομένων μας με το πρόγραμμα netbeans. Αυτό το υλοποιήσαμε κατεβάζοντας το mysql connector και αφού εισάγαμε τον φάκελο του στον φάκελο libraries του project μας συνδέσαμε από το section services του net beans την κατάλληλη βάση που θα χρησιμοποιήσουμε. Έπειτα αφού συνδέσαμε την βάση μας , αρχίσαμε να φτιάχνουμε τις διεπαφές που μας ζητήθηκαν. Αρχικά φτιάξαμε την διεπαφή log in με την οποία ο IT της βάσης μας θα εισάγει το όνομα του και τον δικό του χαρακτηριστικό κωδικό , ο οποίος αποτελείται από τα τέσσερα αρχικά γράμματα από το επώνυμο και το όνομα του. Έτσι αν υπάρχει ο IT : PANAGIOTA KARAGIANNH θα πρέπει να εισάγει στο κελί Username το όνομα PANAGIOTA και στο κελί του Password το KARAPANA. Για να μπορέσουμε στις κάθε διεπαφές μας να χρησιμοποιήσουμε εντολές sql προκειμένου να διαβάσουμε ονόματα ή να μπορούμε να εισάγουμε ή να διαγράφουμε ή να ανανεώνουμε αλλά και να εμφανίζουμε χρησιμοποιήσαμε τον παρακάτω κώδικα:

```
String data = "jdbc:mysql://localhost:3306/travel_agency";

try{

    Class.forName("com.mysql.cj.jdbc.Driver");

    Connection con = DriverManager.getConnection(data,"root","karagiannis");

    { κομμάτι κώδικα για την κάθε διεπαφή }

}

catch (Exception e){

System.out.println("Error:" + e.toString() + e.getMessage());

}
```

Ο κώδικας τον οποίο παραθέσαμε πιο πάνω συνδέεται με την βάση δεδομένων μας που έχουμε στην MySQL χρησιμοποιώντας το JDBC API. Πρώτα φορτώνει την κλάση προγράμματος οδήγησης "com.mysql.cj.jdbc.Driver" η οποία και στη συνέχεια δημιουργεί μια σύνδεση με τη βάση δεδομένων χρησιμοποιώντας τη διεύθυνση URL "jdbc:mysql://localhost:3306/travel_agency" με χρήση «root» που είναι το όνομα χρήστη του υπολογιστή που έχουμε συνδεθεί στην sql και κωδικό «karagiannis» που είναι ο δικός μας προσωπικός κώδικας για τη sql .Προκειμένου να χρησιμοποιήσουμε την σύνδεση μας για τις απαραίτητες εντολές στην συνέχεια τον αποθηκεύουμε σε μια μεταβλητή con.Τον συγκεκριμένο κώδικα τον χρησιμοποιούμε σε όλες μας τις διεπαφές προκειμένου να μπορούμε να συνδεθούμε στην βάση μας και να χρησιμοποιήσουμε κατάλληλες εντολές όπου θα μας ζητηθούν στην συνέχεια.

Έτσι για την log in διεπαφή μας χρησιμοποιούμε το συγκεκριμένο ερώτημα:

```
String sql = "select wrk_name,IT_password from worker,IT where wrk_AT=IT_AT  
AND wrk_name=? AND IT_password=?";
```

έτσι ώστε να καταφέρουμε να ανακτήσουμε το όνομα του IT που βρίσκεται στον πίνακα worker αλλά και τον κωδικό του που βρίσκεται στον πίνακα IT. Έπειτα χρησιμοποιούμε την κλάση PreparedStatement για να δημιουργήσουμε μια προετοιμασμένη δήλωση έτσι ώστε να ορίσουμε τις τιμές για τα σύμβολα κράτησης θέσης στο ερώτημα SQL. Από τις τιμές αυτές που έχουμε κρατήσει τις ελέγχουμε με τις τιμές των κελίων Username και Password και άμα κάποια από αυτές τις τιμές ταιριάζει τότε μας εμφανίζεται κατάλληλο μήνυμα ότι τα στοιχεία ταιριάζουν και ότι μπορούμε να προχωρήσουμε την διαδικασία μας. Και αφού εμφανιστεί το μήνυμα αυτό τότε εκτελείται η εντολή :

```
indTable ft = new findTable( );
```

```
ft.setVisible(true);
```

```
setVisible(false);
```

με την οποία πάμε σε μια νέα διεπαφή στη οποία θα μπορέσουμε να επιλέξουμε τον κατάλληλο πίνακα για να προχωρήσουμε. Αυτόν τον κώδικα τον χρησιμοποιούμε σε κάθε μας διεπαφή για την δημιουργία των κουμπιών Back, Next και Log out.

Έπειτα αφού μπούμε στην διεπαφή FindTable έχουμε εισαγει σε μια λιστα όλα τα ονοματα των πινακων μας προκειμενου να μπορεσουμε να επιλέξουμε σε ποιον πίνακα θέλουμε να προχωρήσουμε να κάνουμε ενέργειες .Για την μετάβαση σε κάθε πίνακα έχουμε εκτελέσει μια δομή switch-case όπου όταν ο δίκτης της λίστας δείξει σε μια από τις τιμές που έχουμε εισάγει τότε μεταφερόμαστε στον και στον αντίστοιχο πίνακα.

Για κάθε πίνακα που έχουμε στην βάση μας έχουμε φτιάξει και μια αντίστοιχη διεπαφή. Σε κάθε διεπαφή έχουμε δημιουργήσει και αντίστοιχα κελία για κάθε στοιχείο του πίνακα δλδ για τον πίνακα worker θα φτιάξουμε τόσα κελία όσα και οι στήλες του πίνακα μας. Έπειτα έχουμε δημιουργήσει 4 κουμπιά (Buttons) όπου θα μπορούμε να κάνουμε και τις αντίστοιχες διεργασίες (Insert , update , delete , back). Για κάθε ένα button έχουμε εισάγει και τις κατάλληλες. Έτσι πχ για την worker διεπαφή μας αλλά και σε όλες τις άλλες διεπαφές μας στο insert button χρησιμοποιούμε τον συγκεκριμένο κώδικα:

```
String data = "jdbc:mysql://localhost:3306/travel_agency";  
  
try{  
  
    Class.forName("com.mysql.cj.jdbc.Driver");  
  
    java.sql.Connection con = DriverManager.getConnection(data,"root","karagiannis");  
  
    String sql = "insert into worker values(?,?,?,?,?)";  
  
    PreparedStatement pst = con.prepareStatement(sql);  
  
    pst.setString(1, at.getText());
```



```

        pst.setString(2, name.getText());
        pst.setString(3, lastname.getText());
        pst.setString(4, salary.getText());
        pst.setString(5, code.getText());
        pst.executeUpdate();

        JOptionPane.showMessageDialog(null, "Successfully insert");
        con.close();
    }

    catch (Exception e){
        System.out.println("Error:" + e.toString() + e.getMessage());
    }

```

Έτσι καταφέρνουμε να εισάγουμε στον πίνακα worker τις τιμές από τα πεδία κειμένου `at.getText()`, `name.getText()`, `lastname.getText()`, `salary.getText()` και `code.getText()` τα οποία είναι οι τιμές των κελίων μας. Η ίδια διαδικασία ακολουθητέ και στις υπόλοιπες διεπαφές που χειριζόμαστε πίνακες.

Για την worker διεπαφή μας αλλά και σε όλες τις άλλες διεπαφές μας στο update button χρησιμοποιούμε τον συγκεκριμένο κώδικα:

```

if(searching()){
    String data = "jdbc:mysql://localhost:3306/travel_agency";
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
        java.sql.Connection con = DriverManager.getConnection(data,"root","karagiannis");

        String sql = "update worker set wrk_name=?,wrk_lname=?,wrk_salary=? where wrk_br_code=? AND wrk_AT=?";

        PreparedStatement pst = con.prepareStatement(sql);

        pst.setString(1, name.getText());
        pst.setString(2, lastname.getText());
        pst.setString(3, salary.getText());

```

```

        pst.setString(4, code.getText());

        pst.setString(5, at.getText());

        pst.executeUpdate();

        JOptionPane.showMessageDialog(null, "Successfully update");

        con.close();

    }

    catch (Exception e){

        System.out.println("Error:" + e.toString() + e.getMessage());

    }

}

else{

    JOptionPane.showMessageDialog(null, "You cannot update a non existing value ");

}

```

Έτσι επιτυγχάνουμε να ανανεώσουμε στον πίνακα worker τις τιμές του από τα τιμές από τα πεδία κειμένου at.getText(), name.getText(), lastname.getText(), salary.getText() και code.getText() τα οποία είναι οι τιμές των κελίων μας και το update γίνεται πάντα βάση του κλειδιού του εκαστωτε πίνακα . Η ίδια διαδικασία ακολουθητε και στις υπόλοιπες διεπαφές που χειριζόμαστε πίνακες.

Για την worker διεπαφή μας αλλά και σε όλες τις άλλες διεπαφές μας στο delete button χρησιμοποιούμε τον συγκεκριμένο κώδικα:

```

if(searching()){

    String data = "jdbc:mysql://localhost:3306/travel_agency";

    try{

        Class.forName("com.mysql.cj.jdbc.Driver");

        java.sql.Connection con = DriverManager.getConnection(data,"root","karagiannis");

        String sql = "delete from worker where wrk_AT=? and wrk_name=? and wrk_lname=? and wrk_salary=? and wrk_br_code=? ";

        PreparedStatement pst = con.prepareStatement(sql);

        pst.setString(1, at.getText());

        pst.setString(2, name.getText());

```

```

        pst.setString(3, lastname.getText());

        pst.setString(4, salary.getText());

        pst.setString(5, code.getText());

        pst.executeUpdate();

        JOptionPane.showMessageDialog(null, "Successfully deleted");

    }

    con.close();

}

catch (Exception e){

System.out.println("Error:" + e.toString() + e.getMessage());

}

}

else{

    JOptionPane.showMessageDialog(null, "You cannot delete a non existing value ");

}

```

Έτσι διαγράφουμε στον πίνακα worker τις τιμές του από τις τιμές από τα πεδία κειμένου at.getText(), name.getText(), lastname.getText(), salary.getText() και code.getText() τα οποία είναι οι τιμές των κελίων μας. Η ίδια διαδικασία ακολουθητέ και στις υπόλοιπες διεπαφές που χειριζόμαστε πίνακες.

Για να αποφύγουμε τυχών λάθη έχουμε δημιουργήσει την εξής συνάρτηση:

```

public boolean searching (){

    String data = "jdbc:mysql://localhost:3306/travel_agency";

    String m = "";

    boolean chk = false;

    try {

        Class.forName("com.mysql.cj.jdbc.Driver");

        java.sql.Connection con = DriverManager.getConnection(data,"root","karagiannis");

        String sql = "SELECT wrk_AT FROM worker where wrk_AT=?";

        PreparedStatement pst = con.prepareStatement(sql);
    }
}

```

```

        pst.setString(1, at.getText());

        ResultSet rs = pst.executeQuery();

        while(rs.next()){

            m = rs.getString(1);

            System.out.println(m);

        }

    }

    catch (Exception e){

        System.out.println("Error:" + e.toString() + e.getMessage());

    }

    if(m.equalsIgnoreCase(at.getText())){

        chk = true;

    }

    return chk;

}

```

Αυτή η συνάρτηση μας « προστατεύει » και την διάρκεια του update και του delete με τον εξής τρόπο: Όταν θα πάμε να εκτελέσουμε μια από αυτές τις διαδικασίες άμα ο κωδικός που έχουμε εισάγει δεν υπάρχει σε κάποια από τις εγγραφές μας θα μας εκτυπώνει μήνυμα ότι ο κωδικός που εισάγαμε είναι λάθος και ότι δεν μπορούμε να εκτελέσουμε αυτή την διεργασία.

Για την υλοποίηση του ερωτήματος 3.2.2 προκειμένου να αποφευχθούν λάθη και τα την εισαγωγή δεδομένων θέτουμε στις διαπαφές πινάκων στους οποίους έχουμε ENUM στοιχεία όπως παραδείγματος χάριν στον Admin όπου στο τύπος μπορούμε να εισάγουμε μόνο την τιμή logistics, administrative και accounting έχουμε εισάγει δίπλα από το κελί type μια λίστα η οποία περιέχει αυτές τις τρεις συγκεκριμένες τιμές όταν διαλέξουμε μια συμπληρώνεται αυτόματα το κελί που θέλουμε. Αυτό το επιτυγχάνουμε στις υπόλοιπες διαπαφές όπου έχουμε τέτοιους τύπους enum. Ο Κώδικας που χρησιμοποιούμε είναι μια μεταβλητή τύπου string στην οποία αποθηκεύουμε μια από τις επιλεγόμενες τιμές της λίστας και έπειτα με την μέθοδο .setText() εισάγουμε την τιμή της μεταβλητής string στο κελί type και ο κωδικας μας είναι ο ακόλουθος:

```

String selectedValue = jComboBox1.getSelectedItemAt().toString();

admttype.setText(selectedValue);

```

B.1.2 Σενάριο χρήσης

Το σενάριο χρήσης που ακολουθούμε είναι το εξής:

- Αρχικά θα μπορούμε στην εφαρμογή μας με Username PANAGIOTA και Password KARAPANA.
- Έπειτα αφού θα μας εμφανίσει κατάλληλο μήνυμα ότι τα στοιχεία ταιριάζουν μεταξύ τους τότε θα πρέπει να επιλέξουμε την επιλογή OK έτσι ώστε να προχωρήσουμε στις επόμενες ενέργειες μας.
- Επιλέγουμε τον πίνακα worker στον οποίο και θα προσπαθήσουμε να εισάγουμε τα νέα δεδομένα: AT<AA90909765>, Name<KATERINA>, Lastname<PAPAGEORGIOU>, Salary<2050.0> και Branche code <12>.
- Αφού κάνουμε την εισαγωγή και μας εμφανίσει το κατάλληλο μήνυμα θα επιλέξουμε το κουμπί Back έτσι ώστε να πάμε σε νέο πίνακα να κάνουμε εντολή Update.
- Έπειτα επιλέγουμε τον πίνακα Driver όπου αρχικά θα βάλουμε ένα τυχαίο AT έτσι ώστε να δούμε άμα ο περιορισμός που θέσαμε για τα διαφορετικά AT ισχύει. Αφού γίνει ο έλεγχος θα αλλάξουμε τον Route του οδηγού με AT AB20967621 από ABROAD σε LOCAL.
- Έπειτα αφού μας εμφανίσει κατάλληλο μήνυμα ότι έγινε με επιτυχία το update τότε θα επιστρέψουμε πίσω και θα επιλέξουμε τον πίνακα Reservation όπου και στον οποίο θα εκτελέσουμε την εντολή delete για το ID 3932221.
- Τέλος αφού μας εμφανιστεί κατάλληλο μήνυμα ότι έγινε με επιτυχία η διαγραφή θα επιλέξουμε το κουμπί Back και έπειτα από αυτό το κουμπί exit ώστε να ξανά πάμε στη επιφάνεια login του προγράμματος μας .

Παραθέτουμε τα απαραίτητα screenshot και τη διαδικασία που ακολουθήσαμε σχηματικά:

AT: AA90909765
Name: KATERINA
Lastname: PAPAGEORGIOU
Salary: 2050.0
Branch code: 12

AT	Name	Lastname	Salary	Branch Code
AG12343210	GIANNIS	PAPADOPOULOS	2355.0	6
AI56366671	PANAGIOTA	GOURDOUPI	3100.0	21
AI83950702	MARIA	DWROU	1950.0	12
AI85678002	EYGENEIA	KOYNAVI	1700.0	7
AK12161135	SERGIOS	XARAVGIS	2100.0	6
AM10999737	KATERINA	MANOLOPOUL...	1500.0	6
AM15370047	CHRISTOS	GEORGOPOUL...	1850.0	11
AM15375947	DIONISIS	MANOLOPOUL...	1500.0	11
AN03060804	EVANGELIA	PATZOPOULOU	1600.0	11
AN89812526	XRISTOFOROS	LAGIOS	1800.0	16
AN89980021	PETROS	PANAGIOTOU	1200.0	26
AO66468665	PANAGIOTIS	KARAGIANNIS	1000.0	10
AP19992231	KOSTAS	KATSAROS	2110.0	28
AP23678728	PHILIPPOS	ASTERIMATHS	1110.0	4

AT: AB20967999
License: C
Route: LOCAL
Experience: 30

Driver AT	License	Route	Experience
AB20967621	C	ABROAD	30
AN89980021	A	ABROAD	55
AP19992231	A	ABROAD	28
AS23776510	A	ABROAD	28
AT24646601	A	ABROAD	28
HK10010120	A	ABROAD	28
IK26285423	A	ABROAD	28
KK10111213	A	ABROAD	28
KO20959000	A	ABROAD	28
MN55776688	A	ABROAD	28
QM98566671	A	ABROAD	28
RY19087762	A	LOCAL	21

AT:

License:

Route:

Experience:

Driver AT	License	Route	Experience
AB20967621	C	ABROAD	30
AN89980021	A	ABROAD	55
AP19992231			27
AS23776510			25
AT24646601			79
HK10010120			29
IK26285423			47
KK10111213			24
KO20959000			35
MN55776688			28
QM98566671			26
RY19087762	A	LOCAL	21

Hello , choose the appropriate table to move on

- Worker
- Driver
- Guide
- Admin
- Manages
- Languages
- Trip
- Event
- Destination
- Travel_to
- Reservation**
- log
- Offers
- Reservation_Offers
- IT

Id:

Seatnum:

Name:


Lastname:

Adult or Minor:

Id	Seatnum	Name	Lastname	Adult or Minor
1011100				ADULT
1101010				MINOR
1101011				ADULT
1111110				MINOR
1766571				ADULT
1999990				ADULT
3458800				ADULT
3900000				ADULT
3900000				ADULT
3932221	16	MARIA	TRIANTAFY...	MINOR
3932221	17	ELENI	IOANNIDOU	ADULT
8566783	5	BASILHS	XONDROGI...	ADULT
10105557	10	ELENI	MIXALOPO...	ADULT
10105558	9	XRYSSA	KATSIGIAN...	ADULT

Hello , choose the appropriate table to move on

- Branch
- Phones
- Worker
- Driver
- Guide
- Admin
- Manages
- Languages
- Trip
- Event
- Destination
- Travel_to
- Reservation
- log
- Offers



Username:

Password: