**Project Βάσεις Δεδομένων**

****

*Προσωπικά στοιχεία ομάδας*

Σενή Εβελίνα

Α.Μ: 1080416, 4ο έτος, [up1080416@upnet.gr](mailto:up1080416@upnet.gr)

Στεφοπούλου Γεωργία

Α.Μ: 1080439, 4ο έτος, [up1080439@upnet.gr](mailto:up1080439@upnet.gr)

*Περιεχόμενα αναφοράς*

Μέρος Α : Σχεδιασμός ΒΔ και SQL

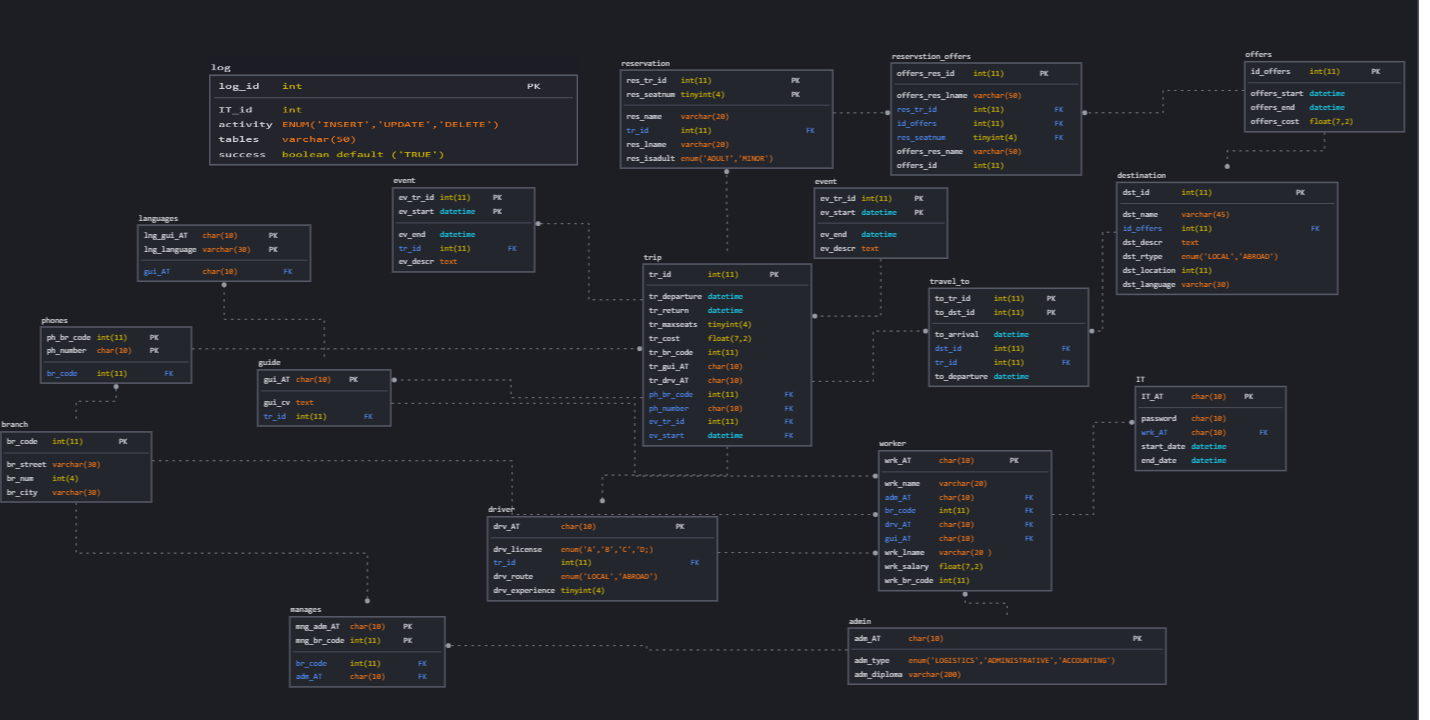
1. Κεφάλαιο 1ο
   1. Ολοκληρωμένο σχεσιακό μοντέλο
   2. Προπαρασκευαστική φάση (create,insert)
   3. Περιγραφή νέων πινάκων
   4. Εισαγωγή δεδομένων στους πίνακες (Insert)
2. Κεφάλαιο 2ο : Κώδικας και παραδείγματα των stored procedures
3. Κεφάλαιο 3ο : Κώδικας, παραδείγματα και screenshots των triggers

Μέρος Β : GUIs

Δεν υλοποιήσαμε το Β μέρος του project.

1. **ΚΕΦΆΛΑΙΟ 1Ο**

**1.1** Ολοκληρωμένο σχεσιακό μοντέλο



Ο σχεδιασμός του παραπάνω σχεσιακού μοντέλου έγινε με την βοήθεια της εφαρμογής SQLDBM. Για καλύτερη ανάγνωση και ανάλυση, θα γίνει επισύναψη της εικόνας στο αρχείο zip.

**1.2** Προπαρασκευαστική φάση

#1

CREATE TABLE branch(

br\_code INT(11) NOT NULL,

br\_street VARCHAR(30) DEFAULT 'unknown' NOT NULL,

br\_num INT(4) NOT NULL,

br\_city VARCHAR(30) DEFAULT 'unknown' NOT NULL,

PRIMARY KEY(br\_code)

);

#2

CREATE TABLE phones(

ph\_br\_code INT(11) NOT NULL,

ph\_number CHAR(10) DEFAULT '0' NOT NULL,

PRIMARY KEY(ph\_br\_code, ph\_number),

CONSTRAINT PHONESBRANCH FOREIGN KEY(ph\_br\_code) REFERENCES branch(br\_code)

ON DELETE CASCADE ON UPDATE CASCADE

);

#3

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 WORKERBRANCH FOREIGN KEY(wrk\_br\_code) REFERENCES branch(br\_code)

ON DELETE CASCADE ON UPDATE CASCADE

);

#4

CREATE TABLE admin(

adm\_AT CHAR(10) DEFAULT 'unknown' NOT NULL,

adm\_type ENUM('LOGISTICS', 'ADMINISTRATIVE', 'ACCOUNTING'),

adm\_diploma VARCHAR(200) DEFAULT 'unknown' NOT NULL,

PRIMARY KEY(adm\_AT),

CONSTRAINT ADMINWORKER FOREIGN KEY(adm\_AT) REFERENCES worker(wrk\_AT)

ON DELETE CASCADE ON UPDATE CASCADE

);

#5

CREATE TABLE manages(

mng\_adm\_AT CHAR(10) DEFAULT 'unknown' NOT NULL,

mng\_br\_code INT(11) NOT NULL,

PRIMARY KEY(mng\_adm\_AT, mng\_br\_code),

CONSTRAINT MANAGESADMIN FOREIGN KEY(mng\_adm\_AT) REFERENCES admin(adm\_AT)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT MANAGESBRANCH FOREIGN KEY(mng\_br\_code) REFERENCES branch(br\_code)

ON DELETE CASCADE ON UPDATE CASCADE

);

#6

CREATE TABLE driver(

drv\_AT CHAR(10) DEFAULT 'unknown' NOT NULL,

drv\_licence ENUM('A', 'B', 'C', 'D'),

drv\_route ENUM('LOCAL', 'ABROAD'),

drv\_experience TINYINT(4) NOT NULL,

PRIMARY KEY(drv\_AT),

CONSTRAINT DRIVERWORKER FOREIGN KEY(drv\_AT) REFERENCES worker(wrk\_AT)

ON DELETE CASCADE ON UPDATE CASCADE

);

#7

CREATE TABLE guide(

gui\_AT CHAR(10) DEFAULT 'unknown' NOT NULL,

gui\_cv TEXT NOT NULL,

PRIMARY KEY(gui\_AT),

CONSTRAINT GUIDEWORKER FOREIGN KEY(gui\_AT) REFERENCES worker(wrk\_AT)

ON DELETE CASCADE ON UPDATE CASCADE

);

#8

CREATE TABLE languages(

lng\_gui\_AT CHAR(10) DEFAULT 'unknown' NOT NULL,

lng\_language VARCHAR(10) DEFAULT 'unknown' NOT NULL,

PRIMARY KEY(lng\_gui\_AT, lng\_language),

CONSTRAINT LANGUAGESGUIDE FOREIGN KEY(lng\_gui\_AT) REFERENCES guide(gui\_AT)

ON DELETE CASCADE ON UPDATE CASCADE

);

#9

CREATE TABLE trip(

tr\_id INT(11) NOT NULL,

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) DEFAULT 'unknown' NOT NULL,

tr\_drv\_AT CHAR(10) DEFAULT 'unknown' NOT NULL,

PRIMARY KEY(tr\_id),

CONSTRAINT TRIPBRANCH FOREIGN KEY(tr\_br\_code) REFERENCES branch(br\_code)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT TRIPGUIDE FOREIGN KEY(tr\_gui\_AT) REFERENCES guide(gui\_AT)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT TRIPDRIVER FOREIGN KEY(tr\_drv\_AT) REFERENCES driver(drv\_AT)

ON DELETE CASCADE ON UPDATE CASCADE

);

#10

CREATE TABLE event(

ev\_tr\_id INT(11) NOT NULL,

ev\_start DATETIME NOT NULL,

ev\_end DATETIME NOT NULL,

ev\_descr TEXT NOT NULL,

PRIMARY KEY(ev\_tr\_id),

CONSTRAINT EVENTTRIP FOREIGN KEY(ev\_tr\_id) REFERENCES trip(tr\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

#11

CREATE TABLE reservation(

res\_tr\_id INT(11) NOT NULL,

res\_seatnum TINYINT(4) NOT NULL,

res\_name VARCHAR(20) DEFAULT 'unknown' NOT NULL,

res\_lname VARCHAR(20) DEFAULT 'unknown' NOT NULL,

res\_isadult ENUM('ADULT', 'MINOR'),

PRIMARY KEY(res\_tr\_id),

CONSTRAINT RESERVATIONTRIP FOREIGN KEY(res\_tr\_id) REFERENCES trip(tr\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

#12

CREATE TABLE destination(

dst\_id INT(11) NOT NULL,

dst\_name VARCHAR(50) DEFAULT 'unknown' NOT NULL,

dst\_descr TEXT NOT NULL,

dst\_rtype ENUM('LOCAL', 'ABROAD'),

dst\_language VARCHAR(30) DEFAULT 'unknown' NOT NULL,

dst\_location INT(11) NOT NULL,

PRIMARY KEY(dst\_id),

CONSTRAINT DESTINATION FOREIGN KEY(dst\_location) REFERENCES destination(dst\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

#13

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 TRAVELTOTRIP FOREIGN KEY(to\_tr\_id) REFERENCES trip(tr\_id)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT TRAVELTODEST FOREIGN KEY(to\_dst\_id) REFERENCES destination(dst\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

**1.3** Περιγραφή νέων πινάκων

Οι ζητούμενοι πίνακες είναι οι εξής:

* Πίνακας IT

CREATE TABLE IT(

IT\_AT CHAR(10) NOT NULL,

password CHAR(10) DEFAULT 'password' NOT NULL,

start\_date DATETIME NOT NULL,

end\_date DATETIME NOT NULL,

PRIMARY KEY(IT\_AT),

CONSTRAINT ITWORKER FOREIGN KEY(IT\_AT) REFERENCES worker(wrk\_AT)

ON DELETE CASCADE ON UPDATE CASCADE

);

Στον παραπάνω πίνακα εισάγουμε πληροφορίες για τον υπάλληλο πληροφορικής. Οι πληροφορίες αυτές είναι ένας μοναδικός αριθμός ταυτότητας, ένας κωδικός, και οι ημερομηνία πρόσληψης του και ημερομηνία αποχώρησης προηγούμενων ΙΤ.

* Πίνακας log

CREATE TABLE log(

log\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

IT\_id INT NOT NULL,

activity ENUM('INSERT', 'UPDATE', 'DELETE') NOT NULL,

tables VARCHAR(50) NOT NULL,

success BOOLEAN DEFAULT 'TRUE',

PRIMARY KEY(log\_id)

);

Ο παραπάνω πίνακας δημιουργήθηκε με σκοπό να εισάγουμε πληροφορίες για τις ενέργειες που γίνονται στην βάση δεδομένων από τον ΙΤ. Οι πληροφορίες που εισάγονται είναι ένας μοναδικός κωδικός, ο μοναδικός κωδικός του ΙΤ, η ενέργεια που έχει εκτελέσει, τον πίνακα που έχει γίνει η ενέργεια.

* Πίνακας offers

CREATE TABLE offers(

id\_offers INT(11) NOT NULL AUTO\_INCREMENT,

offers\_start DATETIME NOT NULL,

offers\_end DATETIME NOT NULL,

offers\_cost FLOAT(7,2) NOT NULL,

PRIMARY KEY(id\_offers),

CONSTRAINT OFFERSDESTINATION FOREIGN KEY(id\_offers) REFERENCES destination(dst\_id)

ON DELETE CASCADE ON UPDATE CASCADE

);

Ο πίνακας offers εισάγει πληροφορίες σχετικά με την διαθέσιμη προσφορά και συγκεκριμένα τον μοναδικό κωδικό της προσφοράς, την ημερομηνία έναρξης και λήξης καθώς και το κόστος της προσφοράς.

* Πίνακας reservation\_offers

CREATE TABLE reservation\_offers(

offers\_res\_id INT(11) NOT NULL AUTO\_INCREMENT,

offers\_res\_lname VARCHAR(50) DEFAULT 'unknown' NOT NULL,

offers\_res\_name VARCHAR(50) DEFAULT 'unknown' NOT NULL,

offers\_id INT(11) NOT NULL,

offers\_advance FLOAT(7,2) NOT NULL,

PRIMARY KEY(offers\_res\_id),

CONSTRAINT OFFERSRESERVATION FOREIGN KEY(offers\_res\_id) REFERENCES reservation(res\_tr\_id)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT RESERVATIONOFFERS FOREIGN KEY(offers\_id) REFERENCES offers(id\_offers)

ON DELETE CASCADE ON UPDATE CASCADE

);

Τέλος ο reservation\_offers δημιουργήθηκε για τις κρατήσεις στις προσφορές ταξιδιών. Πληροφορίες που εισάγονται είναι το το μοναδικό κλειδί της κράτησης της προφοράς, όνομα και επίθετο του ενδιαφερόμενου για την προσφορά, τον κωδικό πρόσβασης του ταξιδιού και το ποσό προκαταβολής.

**1.4** Εισαγωγή δεδομένων στους πίνακες

Στους ήδη υπάρχοντες πίνακες οι εισαγωγή δεδομένων γίνεται ως εξής:

#BRANCH#

INSERT INTO branch VALUES

(1, 'KONSTANTINOUPOLEWS', 1, 'PATRA'),

(2, 'ARATOU', 2, 'ATHINA'),

(3, 'PSILA ALONIA', 3, 'PATRA'),

(4, 'KALAMIA', 4, 'KORINTHOS'),

(5, 'ERMOU', 5, 'ATHINA'),

(6, 'FILIAS', 6, 'FLORINA'),

(7, 'KILKIS', 7, 'PATRA'),

(8, 'AGIAS', 8, 'KASTORIA'),

(9, 'NIKOLAOU', 9, 'KOZANI'),

(10, 'ERMOU', 10, 'KORINTHOS'),

(11, 'KONSTANTINOUPOLEWS', 3, 'THESSAONIKH'),

(12, 'ARATOU', 7, 'KOMOTINI'),

(13, 'ALONIA', 8, 'ALEXANDROUPOLH'),

(14, 'KALAMIA', 10, 'KERKURA'),

(15, 'ERMOU', 6, 'AGRINIO'),

(16, 'FILIAS', 9, 'ATHINA'),

(17, 'KILKI', 8, 'DRAMA'),

(18, 'AGIAS', 1, 'BEROIA'),

(19, 'NIKOLAOU', 20, 'KRHTH'),

(20, 'ERMOU', 25, 'KALAMATA');

#PHONES#

INSERT INTO phones VALUES

(1, 2102563985),

(2, 2105632145),

(3, 2741039588),

(4, 2741039865),

(5, 2741039744),

(6, 2741039636),

(7, 2825246395),

(8, 2104569874),

(9, 2741031455),

(10, 2741031563),

(11, 2102563982),

(12, 2105632141),

(13, 2741039589),

(14, 2741039867),

(15, 2741039743),

(16, 2741039632),

(17, 2825246394),

(18, 2104569870),

(19, 2741031457),

(20, 2741031567);

#WORKER#

INSERT INTO worker VALUES

('AM20005', 'GIWRGOS', 'GEWRGIOU', '750.60', 1),

('AN26441', 'MARIA', 'STERGIOU', '1000.0', 2),

('AM27445', 'GIANNHS', 'PETROU', '2050.3', 3),

('AM20004', 'GEORGIA', 'STEFOPOULOU', '850.6', 4),

('AT25563', 'EVELINA', 'SENI', '900.0', 5),

('AX23654', 'MARIOS', 'KALPAXIS', '950.5', 6),

('AK23262', 'IOANNA', 'AXELI', '800.9', 7),

('AK26445', 'GRIGORIS', 'GRIGORIOU', '7000.0', 8),

('AP25478', 'ANNA', 'ZAMPOUNI', '541.6', 9),

('AP26553', 'SPIROS', 'SENIS', '654.3', 10),

('AM20001', 'GIWRGOS', 'ALEXIOU', '250.60', 11),

('AM26440', 'MARIA', 'VOLIRI', '6000.0', 12),

('AN27449', 'GIANNHS', 'IOANNIDIS', '2050.3', 13),

('AM20008', 'GEORGIA', 'VELAORA', '2850.6', 14),

('AL25567', 'EVELINA', 'PAPADOPOULOU', '1900.0', 15),

('AL23656', 'XRHSTOS', 'KONSTANTINIDIS', '950.5', 16),

('AP23265', 'VANIA', 'AXELI', '800.9', 17),

('AI26444', 'GRIGORIS', 'GKEKAS', '7000.0', 18),

('AU25473', 'MARIA', 'LEKKA', '530.6', 19),

('AX26552', 'GIWRGOS', 'SENIS', '650.3', 20);

#ADMIN#

INSERT INTO admin VALUES

('AM20005', 'LOGISTICS', 'LOGISTIKA'),

('AN26441', 'ADMINISTRATIVE', 'DIOIKISIS'),

('AM27445', 'ACCOUNTING', 'OIKONOMIKO'),

('AM20004', 'LOGISTICS', 'LOGISTIKA'),

('AT25563', 'ADMINISTRATIVE', 'DIOIKISIS'),

('AX23654', 'ACCOUNTING', 'OIKONOMIKO'),

('AK23262', 'LOGISTICS', 'LOGISTIKA'),

('AK26445', 'ADMINISTRATIVE', 'DIOIKISIS'),

('AP25478', 'ACCOUNTING', 'OIKONOMIKO'),

('AP26553', 'LOGISTICS', 'LOGISTIKA'),

('AM20001', 'ACCOUNTING', 'OIKONOMIKO'),

('AM26440', 'LOGISTICS', 'LOGISTIKA'),

('AN27449', 'LOGISTICS', 'LOGISTIKA'),

('AM20008', 'ADMINISTRATIVE', 'DIOIKISIS'),

('AL25567', 'ACCOUNTING', 'OIKONOMIKO'),

('AL23656', 'ACCOUNTING', 'OIKONOMIKO'),

('AP23265', 'LOGISTICS', 'LOGISTIKA'),

('AI26444', 'ADMINISTRATIVE', 'DIOIKISIS'),

('AU25473', 'ADMINISTRATIVE', 'DIOIKISIS'),

('AX26552', 'LOGISTICS', 'LOGISTIKA');

#MANAGES#

INSERT INTO manages VALUES

('AM20005', 1),

('AN26441', 2),

('AM27445', 3),

('AM20004', 4),

('AT25563', 5),

('AX23654', 6),

('AK23262', 7),

('AK26445', 8),

('AP25478', 9),

('AP26553', 10),

('AM20001', 11),

('AM26440', 12),

('AN27449', 13),

('AM20008', 14),

('AL25567', 15),

('AL23656', 16),

('AP23265', 17),

('AI26444', 18),

('AU25473', 19),

('AX26552', 20);

#DRIVER#

INSERT INTO driver VALUES

('AM20005', 'A', 'LOCAL', 2),

('AN26441', 'B', 'LOCAL', 4),

('AM27445', 'A', 'ABROAD', 3),

('AM20004', 'C', 'LOCAL', 10),

('AT25563', 'D', 'ABROAD', 12),

('AX23654', 'D', 'LOCAL', 11),

('AK23262', 'B', 'ABROAD', 9),

('AK26445', 'A', 'ABROAD', 8),

('AP25478', 'A', 'ABROAD', 7),

('AP26553', 'C', 'LOCAL', 6),

('AM20001', 'A', 'LOCAL', 4),

('AM26440', 'B', 'LOCAL', 5),

('AN27449', 'C', 'ABROAD', 6),

('AM20008', 'D', 'ABROAD', 12),

('AL25567', 'D', 'ABROAD', 13),

('AL23656', 'A', 'LOCAL', 10),

('AP23265', 'A', 'LOCAL', 2),

('AI26444', 'B', 'LOCAL', 3),

('AU25473', 'C', 'ABROAD', 5),

('AX26552', 'D', 'ABROAD', 4);

#GUIDE#

INSERT INTO guide VALUES

('AM20005', 'Guide for trip to Florina' ),

('AN26441', 'Guide for trip to Athens'),

('AM27445', 'Guide for trip to Korinthos'),

('AM20004', 'Guide for trip to Kalamata'),

('AT25563', 'Guide for trip to Skiathos'),

('AX23654', 'Guide for trip to Paros'),

('AK23262', 'Guide for trip to Rodos'),

('AK26445', 'Guide for trip to Thessaloniki'),

('AP25478', 'Guide for trip to Andros'),

('AP26553', 'Guide for trip to Nauplio'),

('AM20001', 'Guide for trip to London'),

('AM26440', 'Guide for trip to Kerkira'),

('AN27449', 'Guide for trip to Germany'),

('AM20008', 'Guide for trip to Patra'),

('AL25567', 'Guide for trip to Sparti'),

('AL23656', 'Guide for trip to Italy'),

('AP23265', 'Guide for trip to Spain'),

('AI26444', 'Guide for trip to Komotini'),

('AU25473', 'Guide for trip to Patmo'),

('AX26552', 'Guide for trip to Crete');

#LANGUAGES#

INSERT INTO languages VALUES

('AM20005', 'French, Greek, English' ),

('AN26441', 'French, Greek, English'),

('AM27445', 'Greek, English, Arab'),

('AM20004', 'English, German, Greek'),

('AT25563', 'English, Greek, French'),

('AX23654', 'Greek, English'),

('AK23262', 'Spanish, English, Greek'),

('AK26445', 'Arab, Greek, English'),

('AP25478', 'Greek, English'),

('AP26553', 'English, Greek, Persian'),

('AM20001', 'English'),

('AM26440', 'Italian, English'),

('AN27449', 'Spanish, Greek, English'),

('AM20008', 'Italian, Spanish, Greek'),

('AL25567', 'French, Greek, Italian'),

('AL23656', 'English'),

('AP23265', 'Greek'),

('AI26444', 'Spanish'),

('AU25473', 'Italian'),

('AX26552', 'German');

#TRIP#

INSERT INTO trip VALUES

(21, '2023-03-03 07:00:00', '2023-03-07 21:00:00', 60, 120.0, 1, 'AM20005', 'AM20005'),

(22, '2023-11-01 08:00:00', '2023-11-03 21:20:00', 58, 100.0, 2, 'AN26441', 'AN26441'),

(23, '2023-11-10 08:00:00', '2023-11-15 22:00:00', 60, 200.0, 3, 'AM27445', 'AM27445'),

(24, '2023-01-15 09:00:00', '2023-01-15 23:00:00', 20, 35.0, 4, 'AM20004', 'AM20004'),

(25, '2023-03-03 09:09:00', '2023-03-13 20:00:00', 43, 650.0, 5, 'AT25563', 'AT25563'),

(26, '2023-06-02 10:00:00', '2023-06-02 19:20:00', 15, 25.0, 6, 'AX23654', 'AX23654'),

(27, '2023-12-01 11:00:00', '2023-12-06 18:00:00', 60, 360.0, 7, 'AK23262', 'AK23262'),

(28, '2023-12-10 10:00:00', '2023-12-13 18:00:00', 50, 310.0, 8, 'AK26445', 'AK26445'),

(29, '2023-04-20 06:00:00', '2023-04-24 19:00:00', 16, 456.0, 9, 'AP25478', 'AP25478'),

(30, '2023-02-17 08:00:00', '2023-02-20 21:00:00', 80, 145.0, 10, 'AP26553', 'AP26553'),

(31, '2023-05-12 07:00:00', '2023-05-17 21:00:00', 50 ,250.0 ,11, 'AM20001', 'AM20001'),

(32, '2023-05-02 06:30:00', '2023-05-07 21:00:00', 20 ,380.50 ,12, 'AM26440', 'AM26440'),

(33, '2023-06-01 14:00:00', '2023-06-02 20:00:00', 20, 150.00 ,13, 'AN27449', 'AN27449'),

(34, '2023-07-05 08:30:00', '2023-07-06 21:00:00', 30 ,199.90 ,14, 'AM20008', 'AM20008'),

(35, '2023-08-04 08:40:00', '2023-08-08 23:00:00', 11 ,100.00 ,15, 'AL25567', 'AL25567'),

(36, '2023-01-01 08:30:00', '2023-01-05 22:00:00', 50 ,350.50 ,16, 'AL23656', 'AL23656'),

(37, '2023-05-05 09:10:20', '2023-05-10 23:00:00', 24 ,460.00 ,17, 'AP23265', 'AP23265'),

(38, '2023-04-05 10:30:00', '2023-04-10 21:30:00', 26 ,230.00 ,18, 'AI26444', 'AI26444'),

(39, '2023-05-05 11:00:00', '2023-05-10 23:30:00', 60 ,330.00 ,19, 'AU25473', 'AU25473'),

(40, '2023-02-30 08:40:00', '2023-03-04 21:40:00', 14 ,120.50 ,20, 'AX26552', 'AX26552');

#EVENT#

INSERT INTO event VALUES

(21, '2023-03-03 06:30:00', '2023-03-07 22:00:00', 'Travel to Skiathos'),

(22, '2023-11-01 17:00:00', '2023-11-03 23:00:00', 'Travel to Paros'),

(23, '2023-11-10 10:30:00', '2023-11-15 19:00:00', 'Travel to Nauplio'),

(24, '2023-01-15 11:30:00', '2023-01-15 18:00:00', 'Travel to Kalamata'),

(25, '2023-03-03 12:00:30', '2023-03-13 22:30:00', 'Travel to Kerkira'),

(26, '2023-06-02 13:40:58', '2023-06-02 15:00:00', 'Travel to Zakinthos'),

(27, '2023-12-01 13:00:58', '2023-12-06 16:30:00', 'Travel to Patra'),

(28, '2023-12-10 08:50:00', '2023-12-13 16:00:50', 'Travel to Athens'),

(29, '2023-04-20 04:00:00', '2023-04-24 17:30:00', 'Travel to Kastoria'),

(30, '2023-02-17 03:00:20', '2023-02-20 20:00:00', 'Travel to Florina'),

(31, '2023-08-10 15:00:00', '2023-08-15 16:00:50', 'Travel to Akrata'),

(32, '2023-09-20 08:30:00', '2023-09-24 20:00:40', 'Travel to Rome'),

(33, '2023-05-07 09:00:00', '2023-05-12 20:00:00', 'Travel to Bucharest'),

(34, '2023-10-20 09:45:00', '2023-10-24 22:00:48', 'Travel to Paris'),

(35, '2023-06-10 05:00:00', '2023-06-14 16:00:00', 'Travel to Livadeia'),

(36, '2023-09-22 03:00:00', '2023-09-27 18:00:00', 'Travel to Larisa'),

(37, '2023-11-10 08:00:00', '2023-11-14 22:00:02', 'Travel to Patra'),

(38, '2023-12-02 04:00:00', '2023-12-07 23:00:49', 'Travel to Thessaloniki'),

(39, '2023-07-12 07:00:00', '2023-07-15 19:00:00', 'Travel to Naxos'),

(40, '2023-10-01 11:00:00', '2023-10-05 22:00:28', 'Travel to Korinthos');

#DESTINATION#

INSERT INTO destination VALUES

(21, 'Florina', 'Xeimerini Ekdromi', 'LOCAL', 'Greek', 21),

(22, 'Nauplio', 'Eksormisi sto Mpourtzi', 'LOCAL', 'Greek', 22),

(23, 'Kalavrita', 'Xionodromiko kentro', 'LOCAL', 'Greek', 23),

(24, 'Paros', 'Kalokairines diakopes', 'LOCAL', 'Greek', 24),

(25, 'Germany', 'Episkepsi se autokinitoviomixanies', 'ABROAD', 'German', 25),

(26, 'Instabull', 'Episkepsi stin Kapadokia', 'ABROAD', 'Turkish', 26),

(27, 'Athens', 'Episkepsi ston Parthenwna', 'LOCAL', 'Greek', 27),

(28, 'Kerkira', 'Episkepsi sto palati tis prigkipisas Sisis', 'LOCAL', 'Greek',28),

(29, 'London', 'Periodia sta aksiotheata tis Agglias', 'ABROAD', 'English', 29),

(30, 'Arxaia Korinthos', 'Episkepsi sto kastro', 'LOCAL', 'Greek', 30),

(31, 'Akrata','Ekthesi Zografikis','LOCAL','Greek',31),

(32, 'Rome', 'Episkepsi sto Pantheon','ABROAD','Italian',32),

(33, 'Bucharest','Loutra','ABROAD','Roumanika',33),

(34, 'Paris','Episkepsi sto louvro','ABROAD','Gallika',34),

(35, 'Livadeia','Episkepsi stis phges','LOCAL','Greek',35),

(36, 'Larisa','Ekdromi sto arxaio theatro','LOCAL','Greek',36),

(37, 'Patra','Xenagisi sto arxaiologiko mouseio','LOCAL','Greek',37),

(38, 'Thessaloniki','Episkepsi ston leuko pirgo','LOCAL','Greek',38),

(39, 'Naxos', 'Portara', 'LOCAL', 'Greek', 39),

(40, 'Korinthos','Volta ston Isthmo','LOCAL','Greek',40);

#TRAVEL\_TO#

INSERT INTO travel\_to VALUES

(21, 21, '2023-03-03 10:10:10', '2023-03-07 17:10:30'),

(22, 22, '2023-11-01 11:20:30', '2023-11-03 23:50:20'),

(23, 23, '2023-11-10 14:15:15', '2023-11-15 13:15:16'),

(24, 24, '2023-01-15 20:10:50', '2023-01-15 00:20:10'),

(25, 25, '2023-03-03 16:15:14', '2023-03-13 00:30:50'),

(26, 26, '2023-06-02 07:05:06', '2023-06-02 20:30:50'),

(27, 27, '2023-12-01 08:06:05', '2023-12-06 21:21:21'),

(28, 28, '2023-12-10 06:08:04', '2023-12-13 21:00:00'),

(29, 29, '2023-04-20 06:00:00', '2023-04-24 21:00:00'),

(30, 30, '2023-02-17 06:00:00', '2023-02-20 22:00:00'),

(31, 31, '2023-04-03 05:00:00', '2022-04-07 21:00:00'),

(32, 32, '2023-12-01 06:00:00', '2022-12-03 21:00:00'),

(33, 33, '2023-10-10 07:00:00', '2022-10-15 21:00:00'),

(34, 34, '2023-02-15 08:04:00', '2023-02-16 22:15:00'),

(35, 35, '2023-05-03 06:05:00', '2023-05-13 15:00:00'),

(36, 36, '2023-06-02 06:30:00', '2023-06-03 20:20:33'),

(37, 37, '2023-11-01 07:00:00', '2022-11-06 21:00:00'),

(38, 38, '2023-11-10 08:00:00', '2022-11-13 22:00:00'),

(39, 39, '2023-06-20 09:00:00', '2023-06-24 23:00:00'),

(40, 40, '2023-07-17 17:15:00', '2023-07-20 20:00:00');

#RESERVATION#

INSERT INTO reservation VALUES

(21, 60, 'Manolis', 'Kalogirou', 'ADULT'),

(22, 5, 'Georgia', 'Stephopoulou', 'MINOR'),

(23, 7, 'Evelina', 'Seni', 'ADULT'),

(24, 2, 'Marios', 'Kalpaxis', 'ADULT'),

(25, 1, 'Maria', 'Georgiou', 'MINOR'),

(26, 23, 'Giorgos', 'Marinos', 'ADULT'),

(27, 35, 'Anna', 'Zampouni', 'MINOR'),

(28, 46, 'Spiros', 'Axelis', 'MINOR'),

(29, 34, 'Ioanna', 'Axeli', 'ADULT'),

(30, 50, 'Grigoris', 'Senis', 'ADULT'),

(31, 16, 'Manolis', 'Andreou', 'ADULT'),

(32, 15, 'Georgina', 'Stefou', 'MINOR'),

(33, 7, 'Maria', 'Seni', 'ADULT'),

(34, 21, 'Marios', 'Ioannidis', 'ADULT'),

(35, 11, 'Mariza', 'Georgari', 'MINOR'),

(36, 22, 'Giorgos', 'Marinopoulos', 'ADULT'),

(37, 36, 'Anna', 'Papastathopoulou', 'MINOR'),

(38, 44, 'Spiridwn', 'Aggelou', 'MINOR'),

(39, 33, 'Ioanna', 'Alexiou', 'ADULT'),

(40, 55, 'Xristos', 'Senidis', 'ADULT');

Ενώ στους νέους πίνακες ως εξής:

INSERT INTO IT VALUES

('AM20005', 'ABCS', '2023-02-03', '2023-12-01'),

('AN26441', 'ABVGS', '2023-05-06', '2023-12-09'),

('AM27445', 'DSLKD', '2023-07-03', '2023-12-08'),

('AM20004', 'DLSDLS', '2023-01-03', '2023-06-02'),

('AT25563', 'DSDSD', '2023-03-03', '2023-09-03'),

('AX23654', 'WPWPW', '2023-08-02', '2023-12-06'),

('AK23262', 'LSSLSLS', '2023-06-06', '2023-12-01'),

('AK26445', 'ASPLSAPL', '2020-04-08', '2023-12-30'),

('AP25478', 'DSDMSLKD', '2021-09-05', '2023-12-04'),

('AP26553', 'SDKMSLDS', '2022-02-08', '2023-11-01'),

('AM20001', 'SLDSLD', '2023-01-01', '2023-12-08'),

('AM26440', 'DSLDS', '2019-03-01', '2023-01-01'),

('AN27449', 'FLGFL', '2023-08-08', '2023-12-16'),

('AM20008', 'GFLKKLGF', '2023-05-06', '2023-06-04'),

('AL25567', 'GLPFGLF', '2020-04-08', '2023-08-07'),

('AL23656', 'FDLFD', '2020-04-08', '2023-04-04'),

('AP23265', 'FDLFD', '2020-04-08', '2023-01-09'),

('AI26444', 'FDLFKDF', '2021-01-09', '2023-12-05'),

('AU25473', 'FDLFK', '2021-05-03', '2023-02-02'),

('AX26552', 'FDFKDFD', '2021-05-03', '2023-03-06');

INSERT INTO offers VALUES

(21, '2023-06-10', '2023-06-20', 350.0),

(22, '2023-07-10', '2023-07-20', 450.0),

(23, '2023-08-10', '2023-08-20', 500.0);

1. **ΚΕΦΑΛΑΙΟ 2**

Στη συνέχεια παραθέτουμε τις stored procedures με σειρά όπως και στην εκφώνηση.

Stored showDriver

DROP PROCEDURE IF EXISTS showDriver;

DELIMITER $

CREATE PROCEDURE showDriver(

IN driver\_AT INT,

IN driver\_name VARCHAR(25),

IN driver\_lname VARCHAR(25),

IN driver\_salary FLOAT(7,2),

IN driver\_per ENUM('A','B','C','D'),

IN driver\_route ENUM('LOCAL', 'ABROAD'),

IN driver\_exp INT

)

BEGIN

SELECT driver.drv\_AT, worker.wrk\_name, worker.wrk\_lname,worker.wrk\_salary, driver.drv\_licence, driver,drv\_route, driver.drv\_experience

FROM worker

INTO driver\_AT, driver\_name, driver\_lname, driver\_salary, driver\_per, driver\_route, driver\_exp

INNER JOIN driver ON worker.wrk\_AT = driver.drv\_AT

INNER JOIN branch ON worker.wrk\_br\_code = branch.br\_code

END$

DELIMITER ;

Stored showInfo

Δεν καταφέραμε να το υλοποιήσουμε.

Stored whoAmI

DROP PROCEDURE IF EXISTS whoAmI;

DELIMITER $

CREATE PROCEDURE whoAmI(

IN name CHAR(25),

IN lname CHAR(25)

)

BEGIN

SELECT worker.wrk\_name, worker.wrk\_lname

INTO name, lname

FROM worker

WHERE worker.wrk\_lname=lname AND worker.wrk\_name=name;

IF (name=worker.wrk\_name AND lname=worker.wrk\_lname) THEN

SELECT 'OK';

ELSE

SELECT 'You cant delete, you are Admin';

END IF;

END$

DELIMITER ;

Stored returnCostumersa

DROP PROCEDURE IF EXISTS returnCostumersa;

DELIMITER $

CREATE PROCEDURE returnCostumersa(

IN last\_name VARCHAR(50),

OUT total SMALLINT UNSIGNED

)

BEGIN

SELECT offers\_res\_lname, offers\_res\_name FROM reservation\_offers

WHERE offers\_res\_lname = last\_name;

SELECT COUNT(offers\_res\_lname) INTO total FROM reservation\_offers

WHERE offers\_lname = last\_name;

END$

DELIMITER ;

Stored returnCostumers

DELIMITER $

CREATE PROCEDURE returnCostumers(

IN first\_num INT,

IN second\_num INT,

OUT costumers VARCHAR(25)

)

BEGIN

SELECT offers\_res\_name, offers\_res\_lname FROM reservation\_offers

WHERE offers\_res\_id BETWEEN first\_num AND second\_num;

SELECT offers\_res\_name, offers\_res\_lname INTO costumers FROM reservation\_offers

WHERE offers\_res\_id BETWEEN first\_num AND second\_num;

END$

DELIMITER ;

Στις παραπάνω Stored Procedures δεν καταφέραμε να φτάσουμε στο επιθυμητό αποτέλεσμα.

**3. ΚΕΦΑΛΑΙΟ 3**

Στη συνέχεια παραθέτουμε τα triggers με σειρά όπως ζητούνται και στην εκφώνηση.

Trigger log

Εκτελούμε τις ενέργειες insert, update, delete αντίστοιχα για καθένα από τους πίνακες trip,event, reservation, travel\_to και destination. Δεν καταφέραμε με κάποιο τρόπο να τρέξουμε παράδειγμα για τα insert και update

*Για τον trip*

-- INSERT `trip`

DROP TRIGGER IF EXISTS insert\_trip;

DELIMITER $

CREATE TRIGGER insert\_trip BEFORE INSERT ON trip

FOR EACH ROW

BEGIN

INSERT INTO log(IT\_id, activity, tables)

VALUES ('IT\_id', 'INSERT', 'trip');

END$

DELIMITER ;

-- UPDATE `trip`

DROP TRIGGER IF EXISTS update\_trip;

DELIMITER $

CREATE TRIGGER update\_trip BEFORE UPDATE ON trip

FOR EACH ROW

BEGIN

INSERT INTO log(IT\_id, activity, tables)

VALUES ('IT\_id', 'UPDATE', 'trip');

END$

DELIMITER ;

*Για τον πίνακα reservation*

-- INSERT `reservation`

DROP TRIGGER IF EXISTS insert\_reservation;

DELIMITER $

CREATE TRIGGER insert\_reservation BEFORE INSERT ON reservation

FOR EACH ROW

BEGIN

INSERT INTO log(IT\_id, activity, tables)

VALUES ('IT\_id', 'INSERT', 'reservation');

END$

DELIMITER ;

-- UPDATE `reservation`

DROP TRIGGER IF EXISTS update\_reservation;

DELIMITER $

CREATE TRIGGER update\_reservation BEFORE UPDATE ON reservation

FOR EACH ROW

BEGIN

INSERT INTO log(IT\_id, activity, tables)

VALUES ('IT\_id', 'UPDATE', 'reservation');

END$

*DELIMITER ;*

*Για τον πίνακα travel\_to*

-- INSERT `travel\_to`

DROP TRIGGER IF EXISTS insert\_travel\_to;

DELIMITER $

CREATE TRIGGER insert\_travel\_to BEFORE INSERT ON travel\_to

FOR EACH ROW

BEGIN

INSERT INTO log(IT\_id, activity, tables)

VALUES ('IT\_id', 'INSERT', 'travel\_to');

END$

DELIMITER ;

-- UPDATE `travel\_to`

DROP TRIGGER IF EXISTS update\_travel\_to;

DELIMITER $

CREATE TRIGGER update\_travel\_to BEFORE UPDATE ON travel\_to

FOR EACH ROW

BEGIN

INSERT INTO log(IT\_id, activity, tables)

VALUES ('IT\_id', 'UPDATE', 'travel\_to');

END$

DELIMITER ;

*Για τον πίνακα destination*

*-- INSERT `destination`*

*DROP TRIGGER IF EXISTS insert\_destination;*

*DELIMITER $*

*CREATE TRIGGER insert\_destination BEFORE INSERT ON destination*

*FOR EACH ROW*

*BEGIN*

*INSERT INTO log(IT\_id, activity, tables)*

*VALUES ('IT\_id', 'INSERT', 'destination');*

*END$*

*DELIMITER ;*

*-- UPDATE `destination`*

*DROP TRIGGER IF EXISTS update\_destination;*

*DELIMITER $*

*CREATE TRIGGER update\_destination BEFORE UPDATE ON destination*

*FOR EACH ROW*

*BEGIN*

*INSERT INTO log(IT\_id, activity, tables)*

*VALUES ('IT\_id', 'UPDATE', 'destination');*

*END$*

*DELIMITER ;*

Trigger denyChange

Το παρακάτω κομμάτι κώδικα είναι λειτουργικό και αποτρέπει την αλλαγή την της ημερομηνίας αναχώρησης και επιστροφής. Δεν καταφέραμε με κάποιο τρόπο να ελέγχει αν υπάρχει καταχώρηση κράτησης. Για να το καταφέρουμε αυτό θα έπρεπε να γίνεται έλεγχος αν το tr\_id = res\_tr\_id. Ίσως με την χρήση μίας IF statement.

DELIMITER $

CREATE TRIGGER denyChange

BEFORE UPDATE ON trip

FOR EACH ROW

BEGIN

IF NEW.tr\_id = OLD.tr\_id THEN

IF NEW.tr\_departure <> OLD.tr\_departure THEN

SIGNAL SQLSTATE VALUE ‘45000’

SET MESSAGE\_TEXT = ‘You cannot change your departure date’;

END IF;

IF NEW.tr\_return <> OLD.tr\_return THEN

SIGNAL SQLSTATE VALUE ‘45000’

SET MESSAGE\_TEXT = ‘You cannot change your return date’;

END IF;

IF NEW.tr\_cost <> OLD.tr\_cost THEN

SIGNAL SQLSTATE VALUE ‘45000’

SET MESSAGE\_TEXT = ‘You cannot change your trip cost’;

END IF;

END IF;

END $

DELIMITER ;

Έπειτα παραθέτουμε στιγμιότυπο για παραδείγματα όπου η αλλαγή κόστους, ημερομηνίας αναχώρησης και επιστροφής δεν είναι εφικτή :

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Trigger changeSalary

Το παρακάτω κομμάτι είναι λειτουργικό και αποτρέπει την μείωση του μισθού ενός υπαλλήλου.

DELIMITER $

CREATE TRIGGER changeSalary

BEFORE UPDATE ON worker

FOR EACH ROW

BEGIN

IF NEW.wrk\_salary < OLD.wrk\_salary THEN

SIGNAL SQLSTATE VALUE ‘45000’

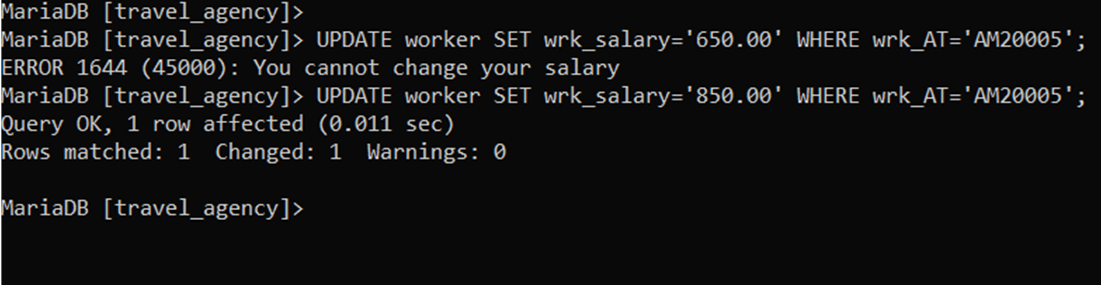
SET MESSAGE\_TEXT = ‘You cannot change your salary’;

END IF;

END $

DELIMITER ;

Παραθέτουμε και στιγμιότυπα από την αύξηση του μισθού, που είναι επιτρεπτή, και για την μείωση που δεν είναι επιτρεπτή. Πιο συγκεκριμένα, με την εντολή κλήσης UPDATE :



Εκτελέσαμε ένα παράδειγμα για τον υπάλληλο με Α.Τ: ΑΜ20005, όπου μείωνοντας τον μισθό του εμφανίζεται κατάλληλο μήνυμα απότρεψης, ενώ στη συνέχεια η αύξηση του είναι εφικτή.

