

Πολυτεχνική Σχολή
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής



CEID

Project Βάσεων Δεδομένων 2020-2021

ΑΒΡΑΜΟΠΟΥΛΟΣ ΜΙΧΑΗΛ

ΑΜ: 1067451

ΠΑΝΑΪΚΑΣ ΣΩΤΗΡΙΟΣ

ΑΜ: 1067412

Contents

Μέρος Α: ΒΔ και SQL	4
ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΠΡΟΕΚΤΑΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	5
Αναφορικά με το πρώτο ερώτημα.	5
Αναφορικά με το δεύτερο ερώτημα.....	6
Αναφορικά με το τρίτο ερώτημα.....	7
ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΚΑΤΑΣΚΕΥΗ ΤΩΝ PROCEDURES	8
Για το πρώτο ερώτημα.....	8
Για το δεύτερο ερώτημα.....	9
Για το τρίτο ερώτημα.....	9
EXTRA PROCEDURE.....	11
ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΚΑΤΑΣΚΕΥΗ ΤΩΝ TRIGGERS.....	11
Για το πρώτο ερώτημα.....	11
Για το δεύτερο ερώτημα.....	11
Για το τρίτο ερώτημα.....	12
EXTRA TRIGGER	13
Μέρος Β: GUI	14
Log In	14
Manager.....	15
Αλλαγή πληροφοριών εταιρείας.....	16
Αλλαγή μισθού θέσης εργασίας.....	17
Μέσος όρος αξιολογητών	18
Αξιολογήσεις υπαλλήλου.....	19
Αλλαγή email και κωδικού.....	20
Αξιολογήσεις εταιρείας	21
Φάκελος υπαλλήλου	22
Χρόνια εμπειρίας.....	26
Evaluator	27
Επεξεργασία θέσης εργασίας	28
Επεξεργασία Αξιολόγησης	29
Θέσεις εργασίας εταιρείας	32
Προσθήκη νέας εργασίας	33
Αιτήσεις υπαλλήλων και δημιουργία αξιολογήσεων.....	35

Εμφάνιση αξιολογήσεων για μία θέση εργασίας	36
Employee	38
Εμφάνιση φακέλου.....	40
Εμφάνιση αιτήσεων.....	41
Admin.....	42
Δημιουργία λογαριασμού.....	43
Δημιουργία νέας εταιρείας.....	44
Εμφάνιση Log	45
SQL.....	47
CREATE	47
INSERTS	54
STORED PROCEDURES	62
EXTRA STORED PROCEDURE	69
TRIGGERS	70
EXTRA TRIGGER	80
Σχεσιακό	82

Μέρος Α: ΒΔ και SQL

Αρχικά να τονίσουμε πως η επέκταση της βάσης δεδομένων έγινε με βάση το ήδη δοσμένο σχεσιακό διάγραμμα χωρίς αυτό να αλλάξει στην μορφή και την δομή του παρά μόνο με τις παραπάνω εισαγωγές οντοτήτων(πινάκων) και ορισμάτων σε αυτούς που προσθέσαμε. Να ξεκινήσουμε επισημαίνοντας την παραδοχή πως ο πίνακας LANGUAGES δεν αξιοποιήθηκε σαν αυτόνομος πίνακας κατά την διάρκεια της διαδικασίας της CREATE αλλά σαν πλειότιμο όρισμα στον πίνακα EMPLOYEE καθώς παρατηρήθηκε πως δεν επηρεάζει κάπως τον αριθμό των εισαγωγών στον πίνακα λόγω της SET οπότε για συντομία χρόνου και κόπου δεν αποτελεί ξεχωριστό πίνακα όπως παρουσιάζεται στο σχεσιακό διάγραμμα της εκφώνησης. Τέλος είναι απαραίτητη η δημιουργία χρηστών με την CREATE USER έτσι ώστε να ελέγχουμε ποιοι χρήστες κάνουν αλλαγές στους πίνακες (πίνακας LOG). Έχουμε συμπεριλάβει την δημιουργία των χρηστών στο αρχείο των INSERTS.

```
CREATE TABLE IF NOT EXISTS EMPLOYEE(  
  USERNAME VARCHAR (12) DEFAULT 'unknown' NOT NULL,  
  BIO TEXT DEFAULT 'unknown' NOT NULL,  
  SISTATIKES VARCHAR (35) DEFAULT 'unknown' NOT NULL,  
  CERTIFICATES VARCHAR (35) DEFAULT 'unknown' NOT NULL,  
  AWARDS VARCHAR (35) DEFAULT 'unknown' NOT NULL,  
  LANG SET('EN','FR','SP','GR'),  
  PRIMARY KEY(USERNAME),  
  CONSTRAINT EMPL_USER FOREIGN KEY (USERNAME) REFERENCES  
  USER(USERNAME)  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```

ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΠΡΟΕΚΤΑΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.

Αναφορικά με το πρώτο ερώτημα.

Δημιουργήθηκε ένας επιπλέον πίνακας με το όνομα EVALUATION ο οποίος είναι υπεύθυνος για την καταγραφή των βαθμολογιών από τον EVALUATOR για την θέση εργασίας. Γι' αυτό παρατηρούμε την ύπαρξη του ξένου κλειδιού JOB_ID προκειμένου να επιτευχθεί η σύνδεση των πινάκων. Παράλληλα προσθέτουμε και δεύτερο ξένο κλειδί στον πίνακα EVALUATION αυτό του πίνακα EVALUATIONRESULT (EVALUATION_ID) προκειμένου να μεταφέρονται οι βαθμολογίες στον ήδη υπάρχων πίνακα EVALUATIONRESULT όπως ζητείται από την εκφώνηση.

```
CREATE TABLE IF NOT EXISTS EVALUATION(  
    EVALUATION_ID INT(4) NOT NULL AUTO_INCREMENT,  
    EMPL_USERNAME VARCHAR(12) NOT NULL,  
    JOB_ID INT(4) NOT NULL,  
    GRADE_A INT(4) DEFAULT NULL,  
    COMMENTS VARCHAR(255) DEFAULT 'unknown' NOT NULL,  
    GRADE_B INT(4) DEFAULT NULL,  
    GRADE_C INT(4) DEFAULT NULL,  
    PRIMARY KEY (EVALUATION_ID),  
    CONSTRAINT EV_JOBID FOREIGN KEY (JOB_ID) REFERENCES JOB(ID)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT EV_EMPL FOREIGN KEY (EMPL_USERNAME)  
    REFERENCES EMPLOYEE(USERNAME)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Αναφορικά με το δεύτερο ερώτημα.

Η εκφώνηση ζητάει μια διαδικασία υποβολής αιτήσεων γι' αυτό κ' εμείς δημιουργήσαμε έναν πίνακα με το όνομα `submit_application`, όπου περιέχει την επιλογή για τον υποψήφιο να εκδηλώσει το ενδιαφέρον του με την εισαγωγή μίας νέας πλειάδας στον πίνακα ή την διαγραφή σε περίπτωση που αλλάξει γνώμη.

Όπως και ο πρώτος πίνακας που δημιουργήσαμε έτσι και αυτός περιέχει δυο ξένα κλειδιά προκειμένου αφενός να συνδέεται με τον πίνακα `EMPLOYEE` μέσω του `EMPL_USERNAME`, αλλά και με τον πίνακα `JOB` μέσω του `JOB_ID`.

Τέλος ο συγκεκριμένος πίνακας αντικατέστησε τον πίνακα `REQUEST EVALUATION` καθώς θεωρούμε ότι η λειτουργία τους είναι παρόμοια.

```
CREATE TABLE IF NOT EXISTS SUBMIT_APPLICATION(  
  ID INT(4) NOT NULL AUTO_INCREMENT,  
  EMPL_USERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,  
  JOB_ID INT(4) DEFAULT '0' NOT NULL,  
  APPLICATION_DATETIME DATETIME DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY(ID),  
  CONSTRAINT SU_EMPL FOREIGN KEY (EMPL_USERNAME) REFERENCES EMPLOYEE(USERNAME)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT SU_JOBID FOREIGN KEY (JOB_ID) REFERENCES JOB(ID)  
  ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Αναφορικά με το τρίτο ερώτημα.

Μας ζητήθηκε να δημιουργήσουμε έναν πίνακα καταγράφων με το όνομα LOG. Αυτός περιέχει ως ορίσματα την ακριβή ημερομηνία και ώρα που διενεργήθηκε κάποια διαδικασία (INSERT,UPDATE,DELETE), από ποιον διενεργήθηκε ,αν είχε επιτυχία η απόπειρα αυτή και τέλος την μορφή που είχε και ποιον πίνακα επηρέαζε. Το USERNAME που καταχωρείται είναι αυτό που είναι αποθηκευμένο στη mysql.user και χρησιμοποιούμε την συνάρτηση **USER()** επιστρέφει τον συνδεδεμένο χρήστη στη μορφή '**USERNAME**'@localhost (περισσότερες πληροφορίες στην ενότητα των Triggers)

```
CREATE TABLE IF NOT EXISTS LOG(  
  ID INT(4) NOT NULL AUTO_INCREMENT,  
  USERNAME VARCHAR(40) DEFAULT 0 NOT NULL  
  PROCESS_DATE DATETIME NOT NULL,  
  SUCCESS BOOLEAN DEFAULT '0' NOT NULL,  
  PROCESS_TYPE ENUM('UPDATE','INSERT','DELETE'),  
  TABLE_NAME ENUM('JOB','EMPLOYEE','SUBMIT APPLICATION'),  
  PRIMARY KEY (ID)  
);
```

ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΚΑΤΑΣΚΕΥΗ ΤΩΝ PROCEDURES

Αρχικά να επισημάνουμε πως εκτός από τα ζητούμενα STORED PROCEDURES δημιουργήθηκε άλλο ένα προκειμένου να υπολογίζεται ο μέσος ορός των βαθμών του κάθε αξιολογητή.

Για τα ερωτήματα της εκφώνησης:

Για το πρώτο ερώτημα

Η PROCEDURE δέχεται ως είσοδο το ονοματεπώνυμο του EMPLOYEE (IN in_name VARCHAR(25), IN in_surname VARCHAR(35)) και χρησιμοποιώντας την SELECT INTO βρίσκουμε το USERNAME του που είναι το κλειδί για να βρούμε όλα τα υπόλοιπα στοιχεία . Αρχικά θέλουμε να εκτυπώνουμε τις αιτήσεις του. Έτσι στο πρώτο SELECT θέλουμε τα στοιχεία του αρχικά απ' τον πίνακα SUBMIT_APPLICATION χρησιμοποιώντας ορισμένα INNER JOIN ώστε να έχουμε πρόσβαση σε όλα τα στοιχεία τα οποία βρίσκονται σε διάφορους πίνακες(EMPLOYEE, USERS, JOB)με τα αντίστοιχα κλειδιά τους(EMPL_USERNAME = USERNAME, EMPL_USERNAME=USERS.USERNAME, SUBMIT_APPLICATION.JOB_ID = JOB.ID) και επιλέγουμε αυτά όπου η ημερομηνία καταχώρησης είναι μεγαλύτερη της σημερινής προκειμένου να έχει ήδη υποβληθεί.

WHERE EMPL_USERNAME=emp_username AND SUBMISSION_DATE > CURDATE()

Έπειτα θέλουμε τις τελικές αξιολογήσεις, γι' αυτό κάνουμε SELECT στα στοιχεία από τον πίνακα EVALUATIONRESULT με την χρήση INNER JOIN στους πίνακες(EMPLOYEE, JOB, EVALUATOR, USERS) με τα αντίστοιχα κλειδιά(EMPL_USERNAME = USERNAME, EVALUATIONRESULT.JOB_ID = JOB.ID, EVALUATOR.USERNAME = JOB.EVALUATOR, EVALUATIONRESULT.EMPL_USERNAME=USERS.USERNAME) όπου υπάρχει το USERNAME του συγκεκριμένου EMPLOYEE(EMPL_USERNAME=emp_username)

Στην συνέχεια θέλουμε τα στοιχεία του αξιολογητή. Συγκεκριμένα το ID, το όνομα και το επίθετο από τον πίνακα USERS με INNER JOIN στους πίνακες EVALUATOR, JOB και EVALUATIONRESULT

Τέλος, θέλουμε τις αξιολογήσεις που δεν έχουν ολοκληρωθεί ακόμα.

Ακολουθείται παρόμοια διαδικασία με τα δυο προηγούμενα SELECT όσων αφορά

τα INNER JOIN με την διαφορά ότι χρησιμοποιείται και LEFT JOIN στην ένωση των πινάκων EVALUATION και EVALUATIONRESULT καθώς θέλουμε μόνο τις πλειάδες που βρίσκονται στον πίνακα EVALUATION και όχι στον πίνακα EVALUATIONRESULT (δηλαδή τις αξιολογήσεις που βρίσκονται σε εξέλιξη).

```
LEFT JOIN EVALUATIONRESULT ON EVALUATION.EVALUATION_ID = EVALUATIONRESULT.EVID
```

```
WHERE EVALUATION.EMPL_USERNAME=emp_username AND EVALUATIONRESULT.EVID IS NULL;
```

Για το δεύτερο ερώτημα

Στην συγκεκριμένη περίπτωση έχουμε ως είσοδο το ID του JOB και του EVALUATOR και αφού γίνεται η δήλωση των τοπικών μεταβλητών οι περισσότερες από τις οποίες χρησιμεύουν στην αποθήκευση των βαθμών και στον υπολογισμό του τελικού αποτελέσματος, γίνεται χρήση CURSOR έτσι ώστε να συμπεριλάβουμε περισσότερες από μια εγγραφές. Αρχικά κάνουμε SELECT το ID, το USERNAME, τους βαθμούς και τα σχόλια από τον πίνακα EVALUATION και ενώνουμε τον πίνακα με την χρήση του INNER JOIN με τους πίνακες JOB, EVALUATOR προκειμένου να έχουμε πρόσβαση σε όλες τις εγγραφές. Έπειτα μέσα στο FETCH γίνεται αρχικά η αρχικοποίηση του grade_sum ίση με το 0 και έπειτα εάν και οι τρεις βαθμοί έχουν καταχωρηθεί (δηλαδή δεν είναι NULL) τότε το grade_sum ισούται με το άθροισμά τους. Τέλος αυτές οι τιμές εισάγονται στον πίνακα EVALUATIONRESULT, η INSERT IGNORE χρησιμοποιείται για να γίνει προσπάθεια εισαγωγής αξιολογήσεων που έχουν γίνει ήδη αποφεύγοντας την πιθανότητα εισαγωγής ενός κλειδιού 2 φορές.

Για το τρίτο ερώτημα

Η STORED PROCEDURE δέχεται ως είσοδο το ID μίας θέσης εργασίας. Χρησιμοποιούμε 2 CURSORS για να εκτυπώσουν τις ολοκληρωμένες και μη αξιολογήσεις στην περίπτωση που αυτές δεν έχουν οριστικοποιηθεί. Στη συνέχεια, χρησιμοποιούμε 2 SELECT INTO. Η 1^η μας επιστρέφει τον αριθμό των αξιολογήσεων που εκκρεμούν σε μία θέση εργασίας και η 2^η την καταμέτρηση

όλων των αξιολογήσεων μίας θέσης εργασίας. Η PROCEDURE μας διενεργεί κάποιους ελέγχους, αρχικά ελέγχει αν υπάρχουν αξιολογήσεις σε μία θέση εργασίας. Έπειτα ελέγχει αν όλες οι αξιολογήσεις έχουν οριστικοποιηθεί, εάν έχουν τότε εμφανίζει τα στοιχεία του εκάστοτε αξιολογούμενου λαμβάνοντας στοιχεία απ' τον πίνακα USERS και EVALUATIONRESULT σε φθίνουσα σειρά(

```
SELECT EVID AS ID, NAME, SURNAME, GRADE  
FROM EVALUATIONRESULT  
INNER JOIN USERS ON EMPL_USERNAME=USERS.USERNAME  
WHERE EMPL_USERNAME=USERS.USERNAME AND JOB_ID=in_id  
ORDER BY GRADE DESC; )
```

Στην περίπτωση που εκκρεμούν ακόμα ορισμένες αξιολογήσεις, χρησιμοποιούμε τον πρώτο CURSOR και εκτυπώνουμε τις αξιολογήσεις που έχουν ολοκληρωθεί προς το παρόν, με μια συνθήκη IF που σαν όρισμα παίρνει τους βαθμούς συνδεδεμένους με AND και ελέγχει εάν οι βαθμοί είναι καταχωρημένοι (δηλαδή ότι δεν είναι NULL). Τέλος κάνουμε SELECT τα στοιχεία του αξιολογούμενου με την βαθμολογία του.

```
SELECT ev_id AS 'ID', user_name AS 'NAME', user_surname AS 'SURNAME', gr_a +  
gr_b + gr_c AS 'GRADE';
```

Και στην συνέχεια επαναφέρουμε στην αρχική του τιμή το flag έτσι ώστε να μπορέσει να δρομολογηθεί ο δεύτερος CURSOR κατά σειρά. Όπου χρησιμοποιούμε την ίδια μεθοδολογία με την συνθήκη IF μόνο που αυτή τη φορά τους βαθμούς τους ενώνει η λογική πράξη OR καθώς θέλουμε να εκτυπώσουμε μόνο τις βαθμολογίες που εκκρεμούν.

```
SELECT ev_id AS 'ID', user_name AS 'NAME', user_surname AS 'SURNAME', gr_a +  
gr_b + gr_c AS 'GRADE'
```

Τέλος, εκτυπώνουμε τον αριθμό των αξιολογήσεων που εκκρεμούν.

```
SELECT COUNT(*) AS 'NUMBER OF EVALUATIONS IN PROCESS' FROM  
EVALUATION WHERE GRADE_A + GRADE_B + GRADE_C IS NULL;
```

Εάν καμιά απ' τις συνθήκες δεν ικανοποιηθεί τότε η STORED PROCEDURE εμφανίζει το αντίστοιχο μήνυμα : NO CANDIDATES FOR THIS POSITION

EXTRA PROCEDURE

Procedure που χρησιμοποιείται στο GUI. Υπολογίζει και επιστρέφει τον μέσο όρο των βαθμών ενός αξιολογητή από τον πίνακα EVALUATIONRESULT. Δέχεται σαν είσοδο το USERNAME του αξιολογητή και επιστρέφει τον Μ.Ο. των βαθμών του.

ΣΧΕΤΙΚΑ ΜΕ ΤΗΝ ΚΑΤΑΣΚΕΥΗ ΤΩΝ TRIGGERS

Για το πρώτο ερώτημα

Δίνονται τρεις ενέργειες για τρεις πίνακες συνεπώς έπρεπε να δημιουργηθούν 9 TRIGGER προκειμένου να καλυφθούν όλες οι περιπτώσεις. Ωστόσο για κάθε ένα ζευγάρι δεν δημιουργείται ένα TRIGGER αλλά δυο προκειμένου να καλυφθεί και η περίπτωση της ανεπιτυχούς ενέργειας. Αρχικά πριν γίνει κάποια ενέργεια (INSERT, DELETE, UPDATE) στους πίνακες EMPLOYEE, JOB ή SUBMIT_APPLICATION ενεργοποιείται ο TRIGGER ΕΝΕΡΓΕΙΑ_ΠΙΝΑΚΑΣ0 (π.χ. INSERT_EMPLOYEE0) όπου γίνεται εισαγωγή στον πίνακα LOG τον συνδεδεμένο χρήστη με την συνάρτηση USER(), την συγκεκριμένη ημερομηνία και ώρα με την χρήση της συνάρτησης CURRENT_TIMESTAMP(), το είδος της ενέργειας και τον πίνακα που πάει να γίνει η ενέργεια. Επίσης θέτουμε το SUCCESS = 0 επειδή μπορεί μετά η ενέργεια να μην ολοκληρωθεί. Μετά ενεργοποιείται ο TRIGGER ΕΝΕΡΓΕΙΑ_ΠΙΝΑΚΑΣ1 (π.χ. INSERT_EMPLOYEE1) εάν η ενέργεια όντως ολοκληρώθηκε με επιτυχία και θα θέσει το SUCCESS = 1.

Για το δεύτερο ερώτημα

Θέλουμε στον πίνακα να ανανεώνονται μερικά μόνο από τα γνωρίσματα του πίνακα COMPANY οπότε αναγκαστικά θα γίνει χρήση των μεθόδων NEW και OLD

έτσι ώστε οι τιμές των γνωρίσματος που θέλουμε να αλλάξουν να συνοδεύονται από το NEW ενώ οι υπόλοιπες με το OLD (AFM,DOY,NAME). Έτσι όπου το OLD στοιχείο είναι διαφορετικό από το NEW γίνεται SET στο NEW TO OLD προκειμένου να μην αλλάξει η κατάσταση του.

Π.χ. IF (OLD.AFM != NEW.AFM) THEN

SET NEW.AFM = OLD.AFM;

END IF;

Για το τρίτο ερώτημα

Το σκεπτικό για την υλοποίηση του συγκεκριμένου TRIGGER είναι το εξής: Εάν ο χρήστης δεν ανήκει στους πίνακες που έχουν ήδη δημιουργηθεί (EMPLOYEE, MANAGER, EVALUATOR) που κληρονομούν απ' τον πίνακα USERS τότε είναι ο διαχειριστής.

Μετά την δήλωση τοπικών μεταβλητών γίνεται χρήση της συνάρτησης USER() για τον εκάστοτε χρήστη και αυτό αποθηκεύεται στην μεταβλητή usr που έχει δηλωθεί λίγο νωρίτερα ως τοπική. Στην συνέχεια μέσω της συνάρτησης REPLACE αφαιρείται το @localhost, που μπαίνει αυτόματα από την συνάρτηση USER, και παραμένει μόνο το όνομα του χρήστη, αυτό αποθηκεύεται στη μεταβλητή usrnw. Έπειτα κάνουμε έναν έλεγχο σε όλους τους πίνακες που κληρονομούν από τον πίνακα USERS τους οποίους τους ενώνουμε με τα αντίστοιχα LEFT JOIN και τα κλειδιά τους για τον έλεγχο μας. Εάν το USERNAME υπάρχει σε έναν από τους πίνακες EMPLOYEE, MANAGER, EVALUATOR τότε αυτό αποθηκεύεται στην μεταβλητή usrcheck. Τέλος, γίνεται έλεγχος για το αν ο χρήστης προσπάθησε να αλλάξει το όνομα του και αν ο συγκεκριμένος χρήστης είναι ο διαχειριστής δηλαδή δεν ανήκει σε κάποια από τις παραπάνω κατηγορίες του SELECT. Εάν είναι όντως διαχειριστής τότε η μεταβλητή usrcheck θα είναι NULL, αλλιώς είναι κανονικός χρήστης.

Προφανώς η δημιουργία του TRIGGER είναι πριν το UPDATE για να αποφευχθούν τυχόν μη επιθυμητά αποτελέσματα.

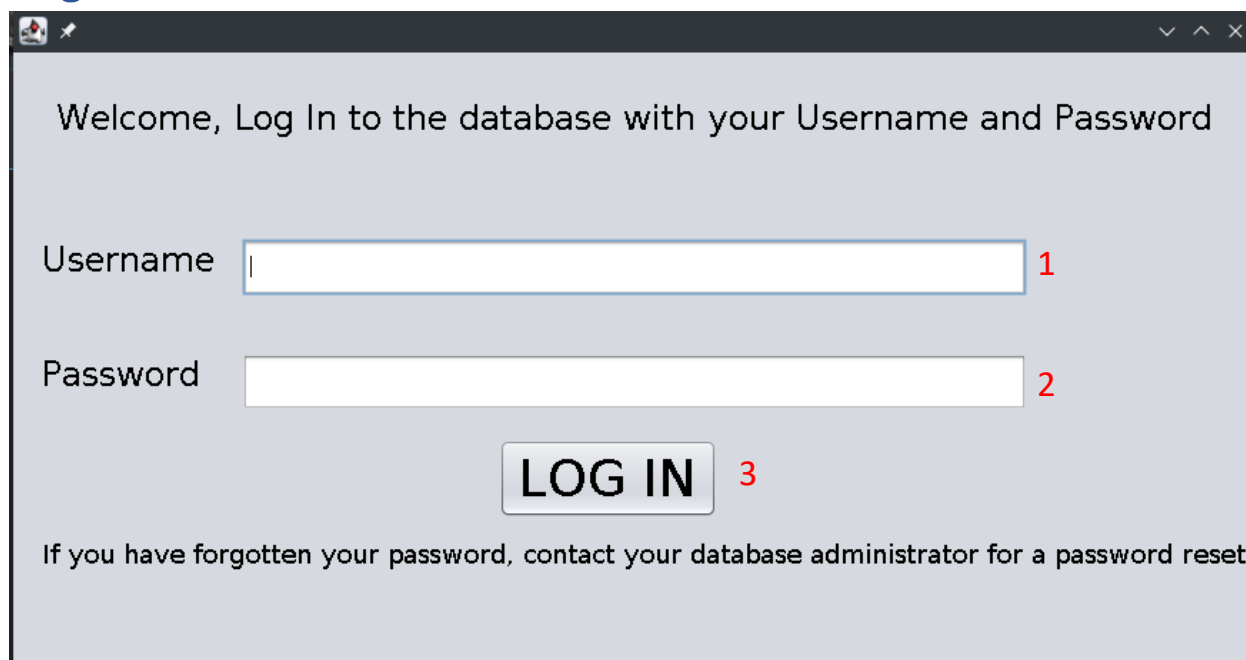
EXTRA TRIGGER

Δημιουργήθηκε ένας παραπάνω TRIGGER ο οποίος είναι υπεύθυνος για τον έλεγχο του email του εκάστοτε χρήστη. Ο οποίος ενεργοποιείται προφανώς πριν την όποια αλλαγή στον πίνακα USERS και ελέγχει αν το mail που εισάγει ο χρήστης δεν είναι στην απαιτούμενη μορφή τότε γίνεται ανάθεση του παλιού mail στο καινούριο προκειμένου να μην αλλάξει και εμφανίζει το αντίστοιχο μήνυμα (New email not in correct form!).

Μέρος B: GUI

Δημιουργήσαμε τις 4 διεπαφές που ζητήθηκαν χρησιμοποιώντας Java και ως IDE το NetBeans. Χρησιμοποιήσαμε JDK 11 διότι είναι από τα πιο σταθερά σε όλες τις πλατφόρμες. Παρακάτω δίνονται Screenshots των διεπαφών με οδηγίες για την χρήση αυτών. Θα θέλαμε να τονίσουμε ότι οποτεδήποτε ζητείται να γραφτεί ημερομηνία πρέπει αυτή να είναι της μορφής YYYY-MM-DD και αν ζητείται και ώρα τότε η μορφή είναι YYYY-MM-DD HH:MM:SS.

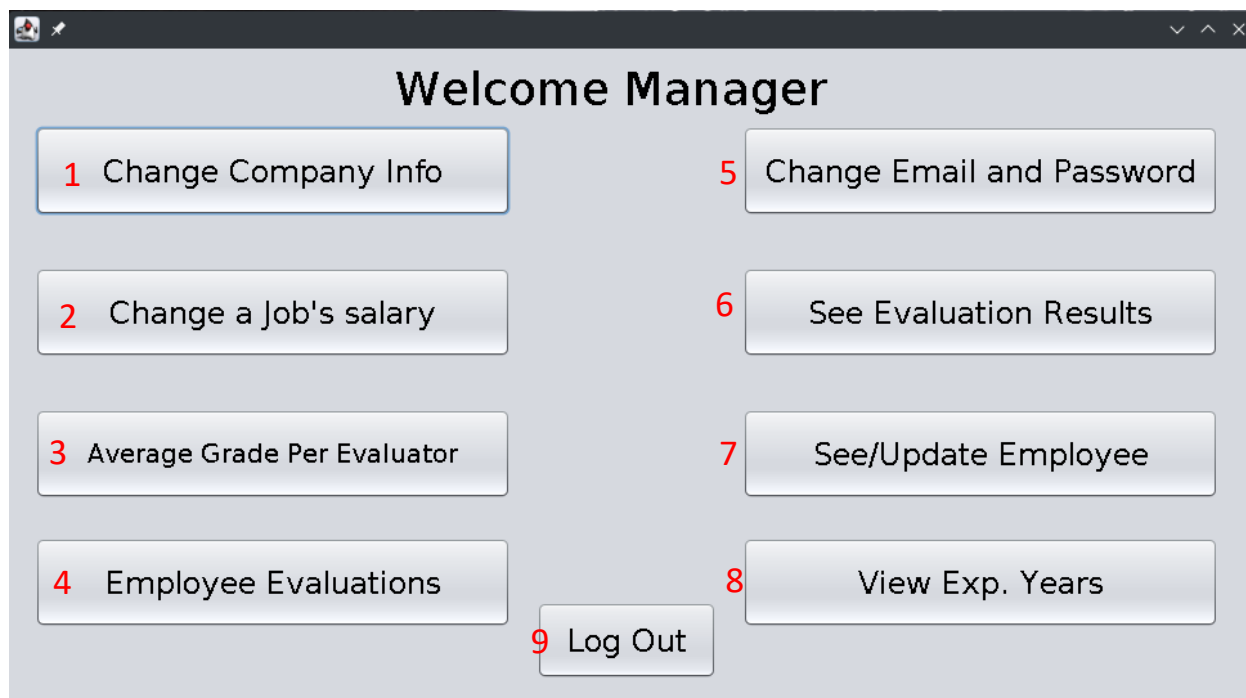
Log In



1. Πεδίο εισαγωγής USERNAME
2. Πεδίο εισαγωγής PASSWORD
3. Κουμπί Log In, ελέγχει εάν τα πεδία δεν είναι άδεια και αν ταιριάζουν με κάποιο χρήστη (και στις δύο περιπτώσεις εκτυπώνεται κατάλληλο μήνυμα). Εάν ο χρήστης υπάρχει στη ΒΔ τότε ανοίγει αυτόματα η κατάλληλη διεπαφή (πχ αν είναι Manager ανοίγει η διεπαφή του Manager)

Manager

Αυτό είναι το κύριο μενού του Manager.



1. Κουμπί για την εμφάνιση του μενού που ο Manager μπορεί να αλλάξει τις πληροφορίες της εταιρείας του.
2. Κουμπί για την εμφάνιση του μενού που ο Manager μπορεί να αλλάξει τον μισθό των θέσεων εργασίας της εταιρείας του.
3. Κουμπί για την εμφάνιση του μενού που ο Manager μπορεί να δει τον μέσο όρο των βαθμών των αξιολογητών της εταιρείας του.
4. Κουμπί για την εμφάνιση του μενού που ο Manager μπορεί να δει τις αξιολογήσεις κάποιου υπαλλήλου.
5. Κουμπί για την εμφάνιση του μενού που ο Manager μπορεί να αλλάξει το email και τον κωδικό του.
6. Κουμπί για την εμφάνιση του μενού που ο Manager μπορεί να δει τις αξιολογήσεις της εταιρείας του που έχουν οριστικοποιηθεί.
7. Κουμπί για την εμφάνιση του μενού που ο Manager μπορεί να δει ή και να αλλάξει τον φάκελο του υπαλλήλου.
8. Κουμπί που ενεργοποιεί popup μήνυμα για να δει ο Manager τα χρόνια εμπειρίας του και να τα αλλάξει
9. Κουμπί αποσύνδεσης, επιστρέφει στην διεπαφή Log In.

Στη συνέχεια ακολουθούν τα μενού των κουμπιών.

Αλλαγή πληροφοριών εταιρείας

AFM	DOY	NAME	PHONE	STREET	NUMBER	CITY	COUNTRY
264001429	A PATRWN	Vasilopoulos	2610222555	Ag. Nikolaou	2	Patra	Greece

1. Πίνακας που αναγράφει τα στοιχεία της εταιρείας του Manager, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Πεδίο εγγραφής, ο Manager πληκτρολογεί το κείμενο που θέλει να αλλάξει
3. Κουμπιά επιλογής, ενημερώνουν ένα συγκεκριμένο στοιχείο με το οτιδήποτε περιέχεται στο πεδίο εγγραφής. Δεν υπάρχει κουμπί ενημέρωσης ΑΦΜ, ΔΟΥ και Ονόματος διότι αυτά τα στοιχεία δεν γίνεται να ενημερωθούν. Με κάθε πάτημα των κουμπιών ενημερώνεται και ο πίνακας.
4. Κουμπί επιστροφής στο κύριο μενού του Manager.

Αλλαγή μισθού θέσης εργασίας

ID	POSITION	EDRA	SALARY	SUBMISSION DATE
5	Accountant	Patra	600.0	2021-03-20
6	Engineer	Patra	900.0	2021-03-20
7	Programmer	Patra	1000.0	2021-03-20
8	Chemical Engineer	Patra	800.0	2021-03-20
9	Assistant Manager	Patra	800.0	2021-03-20
10	Support	Patra	700.0	2021-03-20

Select Job ID: 5

Enter new Salary:

Change Salary

Back

1. Πίνακας που αναγράφει τις θέσεις εργασίας που δέχονται αιτήσεις, εάν μια θέση εργασίας δεν δέχεται αιτήσεις δεν υπάρχει κάποιο νόημα αλλαγής του μισθού, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box για την επιλογή του ID της θέσης εργασίας που επιθυμούμε να αλλάξουμε τον μισθό της.
3. Πεδίο εγγραφής που ο Manager πληκτρολογεί τον νέο μισθό.
4. Κουμπί ανανέωσης μισθού και ανανέωσης πίνακα.
5. Κουμπί επιστροφής στο κύριο μενού του Manager.

[illegible]

- ΑΒΡΑΜΟΠΟΥΛΟΣ ΜΙΧΑΗΛ - ΠΑΝΑΪΚΑΣ ΣΩΤΗΡΙΟΣ

Αξιολογήσεις υπαλλήλου

Employee Evaluations Menu

Applications of Employee

Job ID	Position	Edra

Evaluation Results of Employee and Evaluator

Ev. ID	Position	Edra	Grade	Ev. Surname	Ev. Name

Surname Name

1. Πίνακας που εμφανίζονται όλες οι αιτήσεις ενός υπαλλήλου.
2. Πίνακας που εμφανίζονται οι τελικές αξιολογήσεις ενός υπαλλήλου και το όνομα και επώνυμο του αντίστοιχου αξιολογητή .
3. Πεδίο εγγραφής του επωνύμου του επιθυμητού υπαλλήλου.
4. Πεδίο εγγραφής του ονόματος του επιθυμητού υπαλλήλου.
5. Κουμπί που καλεί το Stored Procedure 3.1 (employee_applications). Τα αποτελέσματα που επιστρέφονται εμφανίζονται στους 2 πίνακες
6. Κουμπί επιστροφής στο κύριο μενού του Manager.

Αλλαγή email και κωδικού

1. Πεδίο εγγραφής νέου email.
2. Κουμπί αλλαγής email, ανανεώνει την ΒΔ με οτιδήποτε γράφεται στο πεδίο 1. Στη περίπτωση που γίνει επιτυχώς η ανανέωση εμφανίζεται κατάλληλο μήνυμα. Στη περίπτωση που το πεδίο είναι κενό εμφανίζεται κατάλληλο μήνυμα
3. Πεδίο εγγραφής νέου κωδικού.
4. Πεδίο επανεγγραφής νέου κωδικού.
5. Κουμπί αλλαγής κωδικού. Ελέγχει εάν κάποιο από τα πεδία 3, 4 είναι κενά και εάν οι κωδικοί ταιριάζουν. Εάν οι κωδικοί ταιριάζουν τότε ανανεώνεται ο κωδικός του χρήστη στην ΒΔ μας και στον πίνακα mysql.user. Σε κάθε περίπτωση εμφανίζεται κατάλληλο μήνυμα για να ενημερώνεται ο χρήστης.
6. Κουμπί επιστροφής στο κύριο μενού του Manager.

Αξιολογήσεις εταιρείας

The screenshot shows a window titled "Evaluation Results". It contains a table with the following data:

ID	Position	Employee Surname	Employee Name	Grade
1	Chemist	Johnston	Mathilde	8
2	Chemist	Williams	Louis	2
3	Chemist	Cunningham	Fedora	10

Below the table, there is a section labeled "Select ID to show comments". It includes a dropdown menu with the value "1" selected, a "Show Comments" button, and a "Back" button.

1. Πίνακας που αναγράφει τις οριστικοποιημένες αξιολογήσεις της εταιρείας του Manager, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box για την επιλογή του ID αξιολόγησης από τον πίνακα 1.
3. Κουμπί που εμφανίζει τα σχόλια της αξιολόγησης που επιλέχθηκε στο dropdown box 2. Τα σχόλια εμφανίζονται σε ένα popur μήνυμα.
4. Κουμπί επιστροφής στο κύριο μενού του Manager.

Φάκελος υπαλλήλου

AM	Surname	Name	Languages	Exp. Years
1	Johnston	Mathilde	EN,GR	4
2	Williams	Louis	EN,SP,GR	2
3	Cunningham	Fedora	GR	1
4	Pierce	Calvin	GR	1
5	Parsons	Jimmie	EN,GR	3

English Greek
French Spanish

Select AM of Employee Select what to see - edit

1 2 3 4 5 6 7

See - Edit Set Languages Back

1. Πίνακας που εμφανίζονται όλοι οι υπάλληλοι της εταιρείας του Manager.
2. Dropdown box με τα AM των υπαλλήλων από τον πίνακα προς επιλογή.
3. Dropdown box που ο Manager επιλέγει τι θέλει να δει ή να αλλάξει (Συστατικές, Πιστοποιητικά, Βραβεία, Projects, Πτυχία, Χρόνια εμπειρίας).
4. Εμφάνιση μενού επεξεργασίας ανάλογα με το AM που επιλέχθηκε στο κουτί 2 και το είδος που πρόκειται για αλλαγή ή για να διαβαστεί. Εξάιρεση είναι τα χρόνια εμπειρίας που εμφανίζεται popup για την αλλαγή τους.

Exp Years

Enter Years

Cancel OK

5. Κουτιά επιλογής γλωσσών.
6. Κουμπί που ανανεώνει τις γλώσσες του υπαλλήλου που επιλέχθηκε από το AM στο κουτί 2. **ΠΡΟΣΟΧΗ** θα εισαχθεί ότι γλώσσα επιλεγθεί και θα διαγραφούν οι παλιές κατά την ανανέωση (π.χ. αν επιλέξω μόνο Γαλλικά και χρήστη με AM = 1 τότε στην ανανέωση του πίνακα ο χρήστη με AM = 1 θα έχει σαν γλώσσα μόνο Γαλλικά, αν θέλουμε να κρατήσουμε και τις


παλιές πρέπει να επιλέξουμε και τα κουτιά των παλιών, δηλαδή Αγγλικά, Ελληνικά και Γαλλικά.

7. Κουμπί επιστροφής στο κύριο μενού του Manager.

Μενού που προκύπτουν από το κουμπί 4 του μενού

<<Φάκελος υπαλλήλου>>

- Τα μενού των συστατικών, βραβείων και πιστοποιητικών είναι παρόμοια οπότε θα δούμε ένα από αυτά



1. Αναλόγως με το τι επιλέξουμε να επεξεργαστούμε θα εμφανίζεται ο κατάλληλος τίτλος. Π.χ. εδώ εμφανίζεται ο τίτλος για τα βραβεία.
2. Πεδίο εγγραφής. Αρχικά εμφανίζεται το κείμενο που είναι αποθηκευμένο στη ΒΔ, μπορούμε να το επεξεργαστούμε, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
3. Κουμπί αποθήκευσης. Αποθηκεύει ότι είναι γραμμένο στο πεδίο εγγραφής.
2. Τέλος εμφανίζεται κατάλληλο μήνυμα αν η αποθήκευση ήταν επιτυχής.

4. Κουμπί επιστροφής στο κύριο μενού <<Φάκελος υπαλλήλου>>. **ΠΡΟΣΟΧΗ** εάν ο χρήστης δεν πατήσει το κουμπί αποθήκευσης τότε δεν θα αποθηκευτεί το νέο κείμενο.

- Projects

Number	URL
1	https://github.com/corwnet/github1s/issues/12
2	https://github.com/nemanjarogic/DesignPatternsLibrary

1

2 Add Project

Select Number

1 3

Edit/Read Description 4

Remove Project 5

Back 6

1. Πίνακας που εμφανίζονται τα Project του χρήστη με AM που επιλέξαμε στο κουτί του προηγούμενου μενού, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Κουμπί προσθήκης Project. Εμφανίζεται popup για να πληκτρολογήσει ο χρήστης το URL του Project (ο αριθμός προστίθεται αυτόματα). **ΠΡΟΣΟΧΗ** η περιγραφή του νέου Project είναι κατά DEFAULT “unknown”, ζητείτε από τον χρήστη να την συμπληρώσει.
3. Dropdown box που επιλέγουμε το νούμερο του Project που είναι προς επεξεργασία.

4. Κουμπί που μας μεταφέρει σε ένα μενού επεξεργασίας του επιλεγμένου Project από το κουτί 2. Το συγκεκριμένο μενού είναι πανομοιότυπο με αυτό των συστατικών, βραβείων και πιστοποιητικών.
5. Κουμπί που αφαιρεί το επιλεγμένο Project από το κουτί 2, γίνεται άμεση ανανέωση.
6. Κουμπί επιστροφής στο κύριο μενού <<Φάκελος υπαλλήλου>>.

- Πτυχία

Titlos	Idryma	Bathmida	Etos	Grade
Apolythrio Lykeiou	Kastritsi	LYKEIO	2010-01-01	19.0
Chemistry	UOP	UNIV	2015-01-01	8.0

1

2

3

4

5

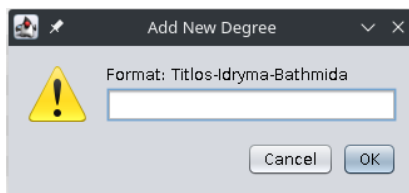
6

7

8

9

1. Πίνακας που εμφανίζει τα πτυχία του επιλεγμένου χρήστη από το προηγούμενο μενού, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box που περιέχει τους τίτλους, ιδρύματα και βαθμίδες όλων των πτυχίων που υπάρχουν στον πίνακα DEGREE τις ΒΔ.
3. Κουμπί που εμφανίζει popup για την εισαγωγή νέου πτυχίου στη ΒΔ.
ΠΡΟΣΟΧΗ η εισαγωγή έχει συγκεκριμένο format ('Τίτλος-Ίδρυμα-Βαθμίδα' χωρίς κενά). Μετά την επιτυχή προσθήκη τα πτυχία στο κουτί 2 ανανεώνονται.

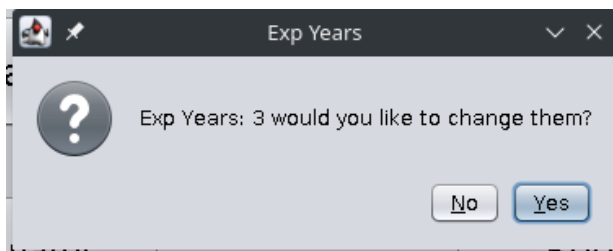


4. Πεδίο εγγραφής έτους πτυχίου.
5. Πεδίο εγγραφής βαθμού πτυχίου.
6. Κουμπί που προσθέτει νέο πτυχίο στον υπάλληλο. Ο τίτλος και το ίδρυμα επιλέγονται από το κουτί 2, το έτος και ο βαθμός από τα αντίστοιχα πεδία και γίνεται εισαγωγή στον πίνακα HAS_DEGREE της ΒΔ και ανανέωση του πίνακα 1.
7. Dropdown box με τους τίτλους των πτυχίων που έχει ο υπάλληλος και εμφανίζονται στον πίνακα 1.
8. Κουμπί που αφαιρεί κάποιο πτυχίο από τον πίνακα HAS_DEGREE του αντίστοιχου υπαλλήλου και τίτλου. Μετά την επιτυχή διαγραφή γίνεται ανανέωση του πίνακα 1.
9. Κουμπί επιστροφής στο κύριο μενού <<Φάκελος υπαλλήλου>>.

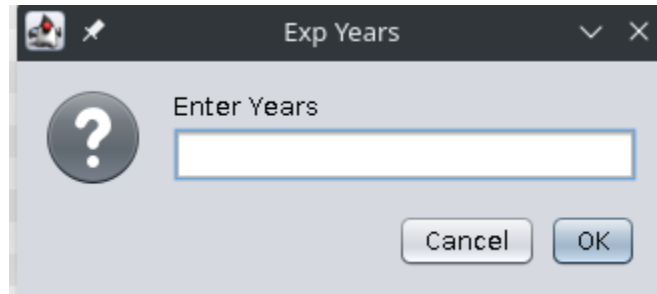
- Επιστροφή στα κουμπιά του κύριου μενού του Manager.

Χρόνια εμπειρίας

Αρχικά εμφανίζεται popup που δείχνει τα χρόνια εμπειρίας που είναι γραμμένα στη ΒΔ.

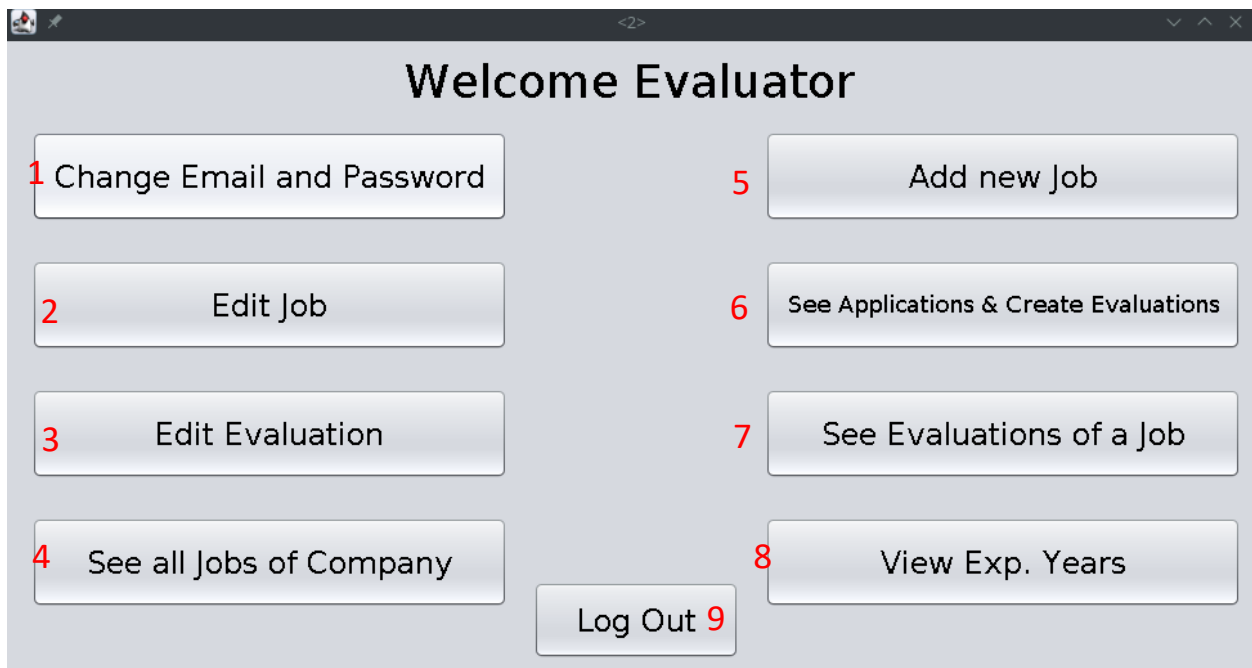


Εάν ο χρήστης επιθυμεί να τα αλλάξει πατάει το κουμπί "Yes"



Evaluator

Αυτό είναι το κύριο μενού του Evaluator



1. Κουμπί για την εμφάνιση του μενού που ο Evaluator μπορεί να αλλάξει το email και τον κωδικό του.
2. Κουμπί για την εμφάνιση του μενού που ο Evaluator μπορεί να επεξεργαστεί μία θέση εργασίας.
3. Κουμπί για την εμφάνιση του μενού που ο Evaluator μπορεί να επεξεργαστεί μία αξιολόγηση.
4. Κουμπί για την εμφάνιση του μενού που ο Evaluator μπορεί να δει όλες τις θέσεις εργασίας της εταιρείας του.
5. Κουμπί για την εμφάνιση του μενού που ο Evaluator μπορεί να δημιουργήσει μία νέα θέση εργασίας.

6. Κουμπί για την εμφάνιση του μενού που ο Evaluator μπορεί να δει τις αιτήσεις των υπαλλήλων και να δημιουργήσει νέα αξιολόγηση.
7. Κουμπί για την εμφάνιση του μενού που ο Evaluator μπορεί να δει τις αξιολογήσεις μίας θέσης εργασίας.
8. Κουμπί που ενεργοποιεί popup μήνυμα για να δει ο Evaluator τα χρόνια εμπειρίας του και να τα αλλάξει
9. Κουμπί αποσύνδεσης, επιστρέφει στην διεπαφή Log In.

Στη συνέχεια ακολουθούν τα μενού των κουμπιών.

- Το μενού αλλαγής email και κωδικού είναι το ίδιο με του Manager.

Επεξεργασία θέσης εργασίας

The screenshot shows a web application window titled "Edit Job Menu". It contains a table with the following data:

ID	Announce Date	Position	Edra	Salary	Submission Date	Start Date	Antikeimeno
5	2021-01-10 00:...	Accountant	Patra	600.0	2021-03-20	2021-03-15	Maths
6	2021-01-10 00:...	Engineer	Patra	900.0	2021-03-20	2021-03-15	Engineer
7	2021-01-10 00:...	Programmer	Patra	1000.0	2021-03-20	2021-03-15	Computer Engi...
8	2021-01-10 00:...	Chemical Engin...	Patra	800.0	2021-03-20	2021-03-15	Chemical Engin...

Below the table, there is a form for editing a job position. It includes:

- A dropdown menu for "ID" with the value "5" selected (labeled 2).
- A text input field for the "ID" value (labeled 3).
- A "Change" button (labeled 1).
- A dropdown menu for "Position" with the value "Chemistry" selected (labeled 4).
- A dropdown menu for "Antikeimeno" with the value "Chemistry" selected (labeled 5).
- An "Add Antikeimeno" button (labeled 6).
- An "Update" button (labeled 7).
- A "Back" button (labeled 8).

1. Πίνακας που εμφανίζει τις θέσεις εργασίας που έχει αναλάβει ο συγκεκριμένος αξιολογητής, εμφανίζονται οι θέσεις που το Start Date είναι μεγαλύτερα της τωρινής μέρας (δεν έχει νόημα να επεξεργαστεί μια θέση εργασίας που έχει τελειώσει η αξιολόγηση της), ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box που περιέχει τα ID των θέσεων εργασίας που εμφανίζονται στον πίνακα.
3. Πεδίο εγγραφής για την αλλαγή που θα γίνει.
4. Dropdown box που ο χρήστης διαλέγει την στήλη προς αλλαγή.

5. Διαθέσιμα αντικείμενα που έχει η ΒΔ.
6. Το κουμπί προσθέτει το αντικείμενο που έχει επιλεγθεί στον κουτί 5 στο πεδίο εγγραφής 3 με την μορφή: Αντικείμενο1, Αντικείμενο2, ..., Αντικείμενο N. Η μορφή αυτή είναι απαραίτητη για την λειτουργία, ο χρήστης μπορεί να διαγράψει αντικείμενα από το πεδίο διαγραφής αλλά θέλει ιδιαίτερη προσοχή.
7. Ανανεώνει τη θέση εργασίας που επιλέχθηκε από το κουτί 2 και τη στήλη από το κουτί 4 (στη συγκεκριμένη θέση θα μπει οτιδήποτε γράφει το πεδίο εγγραφής ανεξαρτήτως τι είχε πριν).
8. Κουμπί επιστροφής στο κύριο μενού του Evaluator.

Επεξεργασία Αξιολόγησης

Ev. ID	Job ID	Surname	Name	Grade A	Grade B	Grade C
4	4	Pierce	Calvin	4		2

1

4 2 Grade A 4 Edit Grade 5 See/Edit Comments 6 See Folder 7

Ev. ID Select Job ID 8 9 Finalize Evaluations 10 Back

1. Πίνακας που εμφανίζει τις αξιολογήσεις του συνδεδεμένου αξιολογητή, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box με τα ID του πίνακα 1.
3. Πεδίο εγγραφής για την επεξεργασία βαθμού.
4. Dropdown box για την επιλογή βαθμού προς επεξεργασία (A, B, C).

5. Κουμπί που ανανεώνει τον βαθμό που επιλέχθηκε από το κουτί 4 και της αξιολόγησης που επιλέχθηκε στο κουτί 2. Ανανεώνεται η ΒΔ όπως και ο πίνακας.
6. Κουμπί που μας μεταφέρει στο μενού επεξεργασίας σχολίων της επιλεγμένης αξιολόγησης από το κουτί 2 (η δομή του μενού είναι ίδια με αυτή της επεξεργασίας Συστατικών, βραβείων, πιστοποιητικών του Manager)
7. Κουμπί που μας μεταφέρει σε μενού που μπορεί ο αξιολογητής να δει τον φάκελο του υπαλλήλου της επιλεγμένης αξιολόγησης από το κουτί 2.
8. Πεδίο εγγραφής ID θέσης εργασίας.
9. Κουμπί που καλεί την Stored Procedure 3.2 (evaluation_checker) για να οριστικοποιήσει τις αξιολογήσεις της θέσης εργασίας που είναι γραμμένη στο πεδίο εγγραφής 8.
10. Κουμπί επιστροφής στο κύριο μενού του Evaluator.

Μενού του φακέλου που βλέπει ο αξιολογητής

See Folder Menu

Projects

Number	URL
1	https://github.com/goldbergonyi/nodebestpractices
2	https://github.com/streamich/react-use
3	https://github.com/ryanmcdermott/clean-code-javascript

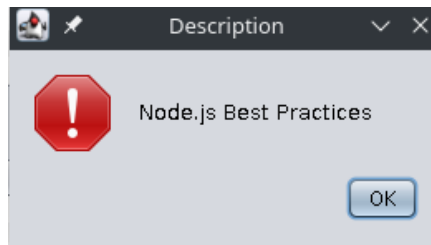
Degrees

Title	Idryma	Bathmida	Etos	Grade
Apolythrio Lykeiou Chemistry	Kastritsi UOP	LYKEIO UNIV	2011-01-01 2015-01-01	16.0 8.0

Number: 3 1 4 See Description Select what you would like to see: Bio 5 6 See 7 Back

1. Πίνακας που εμφανίζει τα Project του επιλεγμένου υπαλλήλου από το προηγούμενο μενού, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.

2. Πίνακας που εμφανίζει τα Πτυχία του επιλεγμένου υπαλλήλου από το προηγούμενο μενού, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
3. Dropdown box για την επιλογή αριθμού Project.
4. Κουμπί που εμφανίζει την περιγραφή του επιλεγμένου Project από το κουτί σε μορφή popup.



5. Dropdown box για να επιλέξουμε τα υπόλοιπα στοιχεία του φακέλου του επιλεγμένου υπαλλήλου (Βιογραφικό, Συστατικές, Πιστοποιητικά, Βραβεία, Γλώσσες).
6. Κουμπί που εμφανίζει popup ανάλογα με το τι επιλέξαμε στο κουτί 5.



7. Κουμπί επιστροφής στο μενού «Επεξεργασία Αξιολόγησης».

- Επιστροφή στα κουμπιά του κύριου μενού του Evaluator

Θέσεις εργασίας εταιρείας

All Jobs of your Company								
ID	Announce Da...	Position	Edra	Salary	Ev. Surname	Ev. Name	Submission D...	Start Date
1	2020-09-10 0...	Chemist	Patra	800.0	Thompson	Claudette	2020-09-20	2020-10-10
4	2020-09-15 0...	Cleaner	Patra	500.0	Thompson	Claudette	2020-09-25	2020-10-15
5	2021-01-10 0...	Accountant	Patra	600.0	Thompson	Claudette	2021-03-20	2021-03-15
6	2021-01-10 0...	Engineer	Patra	900.0	Thompson	Claudette	2021-03-20	2021-03-15
7	2021-01-10 0...	Programmer	Patra	1000.0	Thompson	Claudette	2021-03-20	2021-03-15
8	2021-01-10 0...	Chemical En...	Patra	800.0	Thompson	Claudette	2021-03-20	2021-03-15
9	2021-01-10 0...	Assistant Ma...	Patra	800.0	Harris	Nicholas	2021-03-20	2021-03-15
10	2021-01-10 0...	Support	Patra	700.0	Harris	Nicholas	2021-03-20	2021-03-15

1. Πίνακας που εμφανίζει όλες τις θέσεις εργασίας της εταιρείας του συνδεδεμένου Evaluator, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Κουμπί επιστροφής στο κύριο μενού του Evaluator.

Προσθήκη νέας εργασίας

The screenshot shows a web application window titled "Add new Job Menu". The form contains the following elements:

- Position**: A text input field with a red number 1 below it.
- Edra**: A text input field with a red number 2 below it.
- Submission Date**: A date input field with a red number 3 below it.
- Start Date**: A date input field with a red number 4 below it.
- Salary**: A text input field with a red number 5 below it.
- Antikeimeno**: A dropdown menu showing "Chemistry" with a red number 6 below it.
- Create Antikeimeno**: A button with a red number 7 above it.
- Antikeimeno Input**: A text input field with a red number 9 below it.
- Add Antikeimeno**: A button with a red number 8 above it.
- Add Job**: A button with a red number 10 above it.
- Back**: A button with a red number 11 above it.

1. Πεδίο εγγραφής Θέσης.
2. Πεδίο εγγραφής Έδρας.
3. Πεδίο εγγραφής Submission Date.
4. Πεδίο εγγραφής Start Date.
5. Πεδίο εγγραφής Μισθού.
6. Dropdown box που περιέχει όλα τα αντικείμενα που υπάρχουν στην ΒΔ.
7. Κουμπί που μας μεταφέρει στο μενού δημιουργίας νέου αντικειμένου.
8. Κουμπί που προσθέτει νέο αντικείμενο στο πεδίο εγγραφής 9 με την μορφή Αντικείμενο1, Αντικείμενο2, ..., ΑντικείμενοN.
9. Πεδίο εγγραφής Αντικειμένων (Πανομοιότυπο με αυτό που εξηγήσαμε πιο πάνω).
10. Κουμπί που δημιουργεί νέα εγγραφή στον πίνακα JOB της ΒΔ με τα στοιχεία από όλα τα πεδία εγγραφής.
11. Κουμπί επιστροφής στο κύριο μενού του Evaluator.

Μενού δημιουργίας νέου αντικειμένου

Create Antikeimeno Menu

Title 1

Belongs to 2 None

Description 3

4 Create Antikeimeno 5 Back

1. Πεδίο εγγραφής τίτλου.
2. Dropdown box που περιέχει όλα τα αντικείμενα που υπάρχουν στη ΒΔ για την επιλογή γονέα αντικείμενο (επίσης περιέχει την λέξη "None" αν το αντικείμενο που προσθέσουμε δεν έχει γονέα).
3. Πεδίο εγγραφής περιγραφής αντικειμένου.
4. Κουμπί που δημιουργεί νέα εγγραφή στον πίνακα ANTIKEIM της ΒΔ με τα στοιχεία που δώσαμε.
5. Κουμπί επιστροφής στο μενού «Προσθήκη νέας εργασίας».

- Επιστροφή στα κουμπιά του κύριου μενού του Evaluator

Αιτήσεις υπαλλήλων και δημιουργία αξιολογήσεων

ID	Job ID	Surname	Name	Application Date	Submission Date
4	5	Pierce	Calvin	2021-02-20 00:00:00.0	2021-03-20

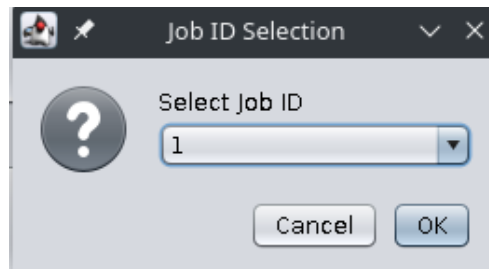
1

2 ID 4 Create Evaluation 3 4 Back

1. Πίνακας που γράφει όλες τις αιτήσεις των υπαλλήλων της εταιρείας που το Submission date δεν έχει περάσει την προθεσμία, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box που περιέχει τα ID των αιτήσεων που δεν έχει δημιουργηθεί αξιολόγηση και είναι στον πίνακα 1.
3. Κουμπί που δημιουργεί αξιολόγηση για την αίτηση που επιλέχθηκε στο κουτί 2. Αφού δημιουργηθεί αφαιρείται από το κουτί 2. Για να εισάγει βαθμούς ο αξιολογητής πρέπει να πάει στο μενού «Επεξεργασία Αξιολόγησης».
4. Κουμπί επιστροφής στο κύριο μενού του Evaluator.

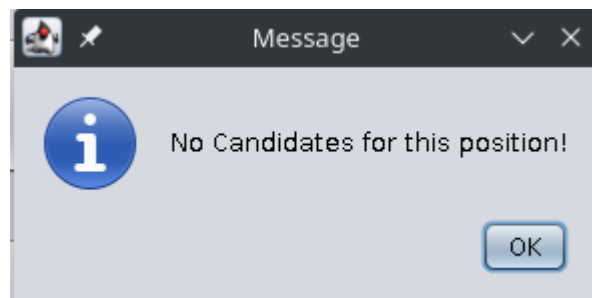
Εμφάνιση αξιολογήσεων για μία θέση εργασίας

Αρχικά εμφανίζεται popup μήνυμα που ζητάει από τον χρήστη να διαλέξει το ID μίας θέσης εργασίας.



Ο χρήστης διαλέγει το ID από το dropdown box και πατάει OK.

Μετά καλείται το Stored Procedure 3.3 (job_checker). Αν δεν υπάρχουν αξιολογήσεις για την επιλεγμένη θέση εργασίας εμφανίζεται το εξής μήνυμα σε μορφή popup:



Αν υπάρχουν αξιολογήσεις σε εξέλιξη τότε εμφανίζεται το εξής μενού:

Ev. ID	Surname	Name	Grade

Ev. ID	Surname	Name	Grade
4	Pierce	Calvin	

Number of Evaluations in process: 1

3 Back

1. Πίνακας που εμφανίζει τις ολοκληρωμένες θέσεις εργασίας.
2. Πίνακας που εμφανίζει τις που εκκρεμούν
3. Κουμπί επιστροφής στο κύριο μενού του Evaluator.

Τέλος αν έχουν οριστικοποιηθεί όλες οι αξιολογήσεις τότε εμφανίζεται το εξής μενού:

Ev. ID	Surname	Name	Grade
3	Cunningham	Fedora	10
1	Johnston	Mathilde	8
2	Williams	Louis	2

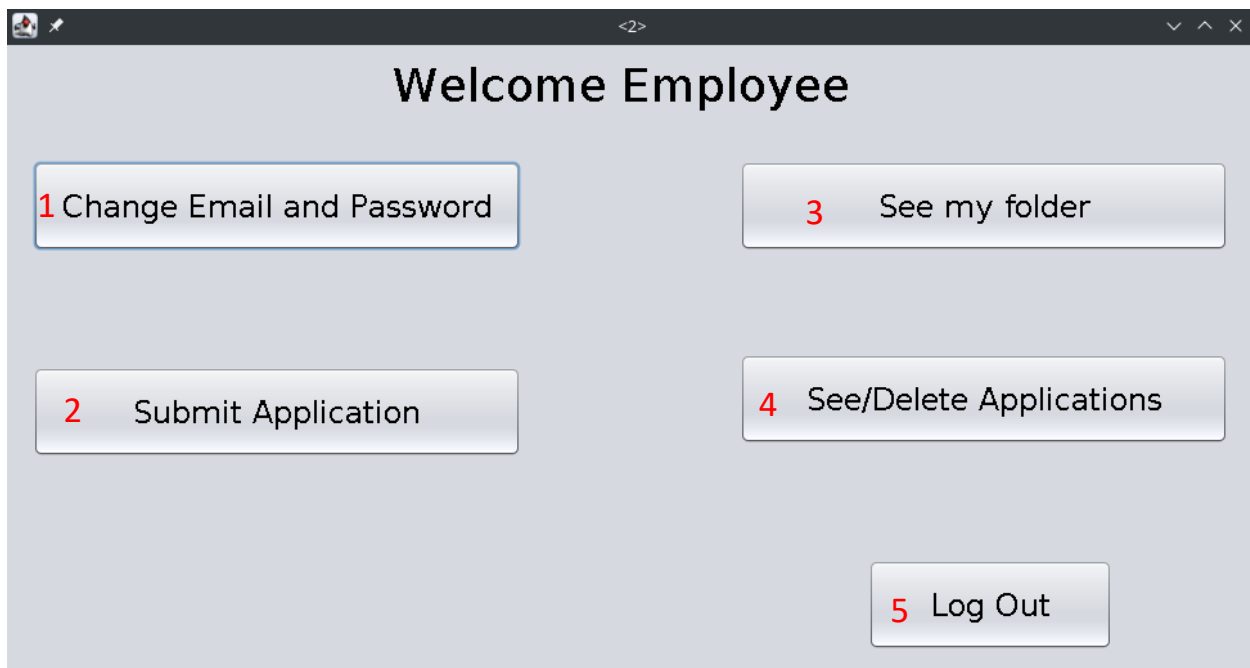
1

2 Back

1. Πίνακας που εμφανίζει τις οριστικοποιημένες αξιολογήσεις με φθίνουσα σειρά με βάση τον βαθμό.
 2. Κουμπί επιστροφής στο κύριο μενού του Evaluator.
- Τα κουμπιά «Εμφάνιση Χρόνων Εμπειρίας» και «Αποσύνδεσης» έχουν την ίδια λειτουργία με αυτή του Manager

Employee

Αυτό είναι το κύριο μενού του υπαλλήλου



1. Κουμπί για την εμφάνιση του μενού που ο Employee μπορεί να αλλάξει το email και τον κωδικό του.
2. Κουμπί για την εμφάνιση του μενού που ο Employee μπορεί να κάνει αίτηση για μία θέση εργασίας.
3. Κουμπί για την εμφάνιση του μενού που ο Employee μπορεί να δει τον φάκελο του.
4. Κουμπί για την εμφάνιση του μενού που ο Employee μπορεί να δει τις αιτήσεις του
5. Κουμπί αποσύνδεσης, επιστρέφει στην διεπαφή Log In.

- Το μενού αλλαγής email και κωδικού είναι ίδιο με τα προηγούμενα.
- Αίτηση για θέση εργασίας

Submit Application Menu

ID	Announce Date	Position	Edra	Salary	Submission Date	Start Date
6	2021-01-10 00:00...	Engineer	Patra	900.0	2021-03-20	2021-03-15
7	2021-01-10 00:00...	Programmer	Patra	1000.0	2021-03-20	2021-03-15
8	2021-01-10 00:00...	Chemical Engineer	Patra	800.0	2021-03-20	2021-03-15
10	2021-01-10 00:00...	Support	Patra	700.0	2021-03-20	2021-03-15

1

ID

2 6

Submit Application 3

Back 4

1. Πίνακας που εμφανίζει τις διαθέσιμες θέσεις εργασίες που ο υπάλληλος δεν έχει κάνει αίτηση. Εμφανίζονται μόνο όσες δεν έχουν περάσει την προθεσμία του Submission date. Ο πίνακας ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box που περιέχει τα ID των διαθέσιμων θέσεων εργασίας.
3. Κουμπί που δημιουργεί εγγραφή στον πίνακα SUBMIT_APPLICATION της ΒΔ. Παράλληλα ανανεώνει τον πίνακα 1.
4. Κουμπί επιστροφής στο κύριο μενού του Employee.

Εμφάνιση φακέλου

See Folder Menu

My Projects

Number	Url
1	https://github.com/conwnet/github1s/issues/12
2	https://github.com/nemanjarogic/DesignPatternsLibrary

My Degrees

Title	Idryma	Bathmida	Etos	Grade
Apolythrio Lykeiou	Kastritsi	LYKEIO	2010-01-01	19.0
Apolythrio Lykeiou	Peiramatiko	LYKEIO	2010-01-01	19.0
Chemistry	UOA	UNIV	2015-01-01	8.0
Chemistry	UOP	UNIV	2015-01-01	8.0

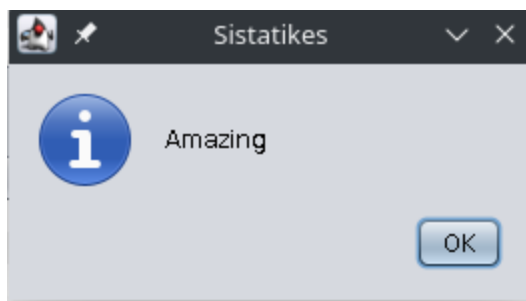
Languages EN,GR 1 6

Select 3 Sistatikes

See 4 5 See/Edit Bio 8 Back

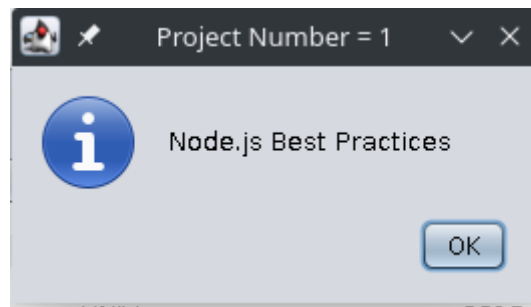
See Project Descr 7

1. Πίνακας που εμφανίζει τα Projects του συνδεδεμένου χρήστη, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Πίνακας που εμφανίζει τα Πτυχία του συνδεδεμένου χρήστη, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
3. Dropdown box για την επιλογή των Συστατικών, Πιστοποιητικών και Βραβείων.
4. Κουμπί για την εμφάνιση της επιλογής από το κουτί 3 σε μορφή popup.



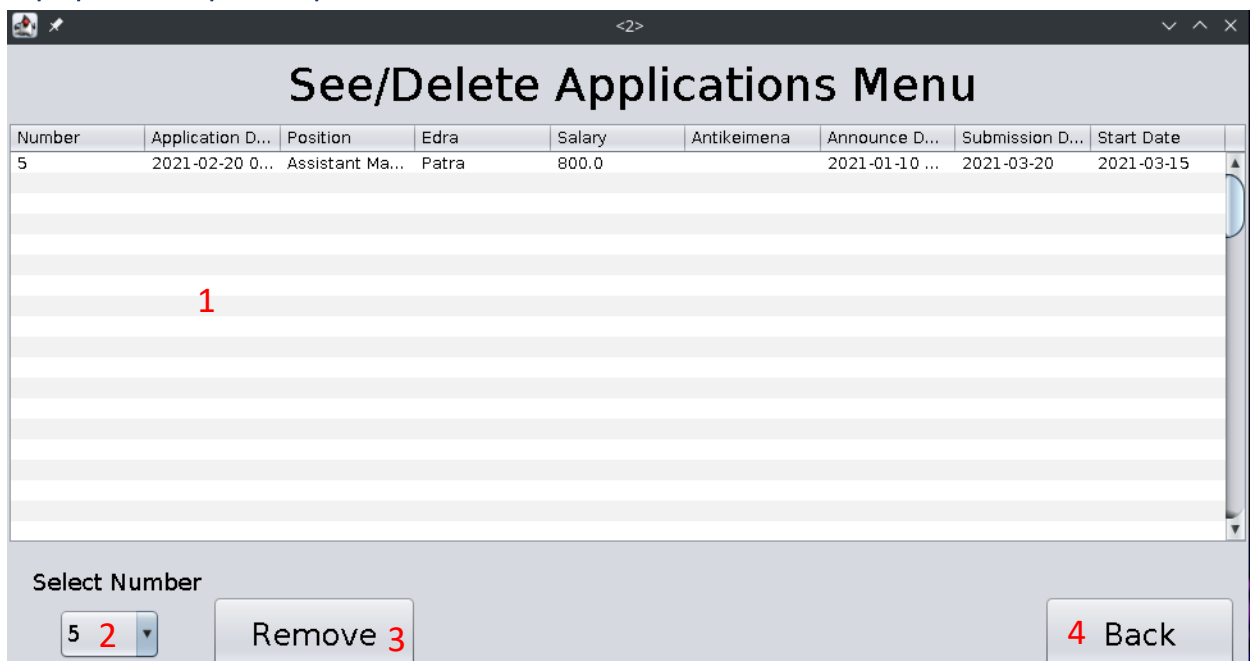
5. Κουμπί για την μετάβαση στο μενού επεξεργασίας βιογραφικού του χρήστη. Το μενού είναι ίδιο με αυτό του Manager για την επεξεργασία των Συστατικών, Πιστοποιητικών και Βραβείων.
6. Dropdown box με τους αριθμούς των Project.

7. Κουμπί για την εμφάνιση της περιγραφής του επιλεγμένου Project από το κουτί 6 σε μορφή popup.



8. Κουμπί επιστροφής στο κύριο μενού του Employee.

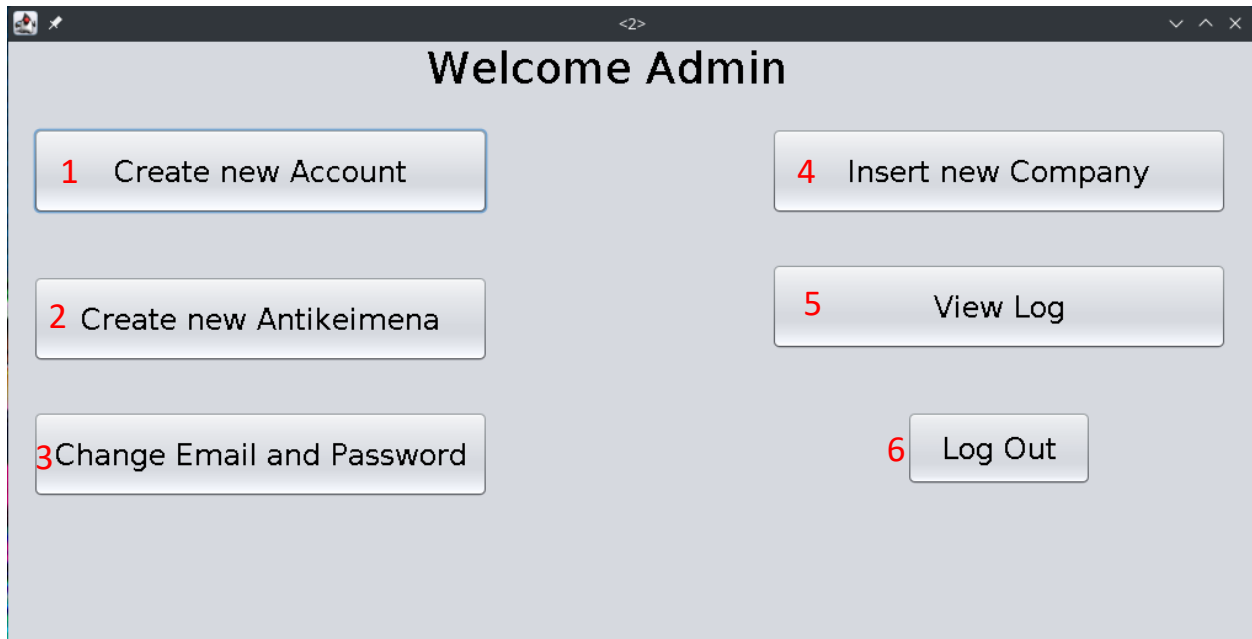
Εμφάνιση αιτήσεων



1. Πίνακας που εμφανίζει τις αιτήσεις του συνδεδεμένου χρήστη, ανανεώνεται με την εισαγωγή του χρήστη στο συγκεκριμένο μενού.
2. Dropdown box με τα ID των αιτήσεων.
3. Κουμπί που αφαιρεί την αίτηση (διαγράφει την εγγραφή στη ΒΔ) που επιλέχθηκε στο κουτί 2.
4. Κουμπί επιστροφής στο κύριο μενού του Employee.

Admin

Αυτό είναι το κύριο μενού του Admin



1. Κουμπί για την εμφάνιση του μενού που ο Admin μπορεί να δημιουργήσει έναν νέο λογαριασμό.
2. Κουμπί για την εμφάνιση του μενού που ο Admin μπορεί να δημιουργήσει νέα Αντικείμενα.
3. Κουμπί για την εμφάνιση του μενού που ο Admin μπορεί να αλλάξει τον κωδικό και το email του.
4. Κουμπί για την εμφάνιση του μενού που ο Admin μπορεί να δημιουργήσει μία νέα εταιρεία.
5. Κουμπί για την εμφάνιση του μενού που ο Admin μπορεί να δει τα δεδομένα του πίνακα LOG.
6. Κουμπί αποσύνδεσης, επιστρέφει στην διεπαφή Log In.

- Τα μενού αλλαγής email και κωδικού και δημιουργία αντικειμένων είναι ίδια με τα προηγούμενα.

Δημιουργία λογαριασμού

Create New Account Menu

Username 1

Name 4

Account Type 7 Employee

Password 2

Surname 5

Firm 8 187771638

Retype Password 3

Email 6

Exp Years 9

10 Create Account 11 Back

1. Πεδίο εγγραφής Username.
2. Πεδίο εγγραφής κωδικού.
3. Πεδίο επανεγγραφής κωδικού.
4. Πεδίο εγγραφής Ονόματος.
5. Πεδίο εγγραφής Επωνύματος.
6. Πεδίο εγγραφής email.
7. Dropdown box που περιέχει τα είδη των λογαριασμών (Employee, Manager, Evaluator).
8. Dropdown box που περιέχει τα ΑΦΜ των εταιρειών.
9. Πεδίο εγγραφής χρόνων εμπειρίας.
10. Κουμπί που δημιουργεί εγγραφές στους πίνακες:
 - α) USERS με τα στοιχεία που δόθηκαν.
 - β) EMPLOYEE, MANAGER ή EVALUATOR ανάλογα με την επιλογή στο κουτί 7 και συμπληρώνονται τα υπόλοιπα στοιχεία.
 - γ) mysql.user.
11. Κουμπί επιστροφής στο κύριο μενού του Admin.

Δημιουργία νέας εταιρείας

Insert New Company Menu

AFM 1

Phone 4

City 7

DOY 2

Street 5

Country 8

Name 3

Number 6

Insert 9

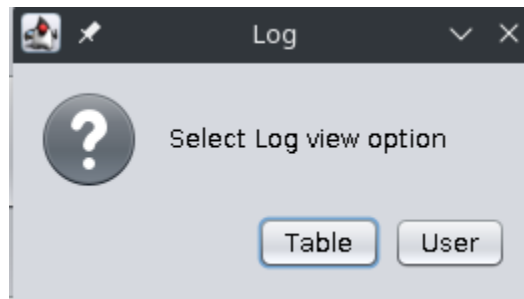
Caution: AFM, DOY and Name cannot be changed after insertion!

10 Back

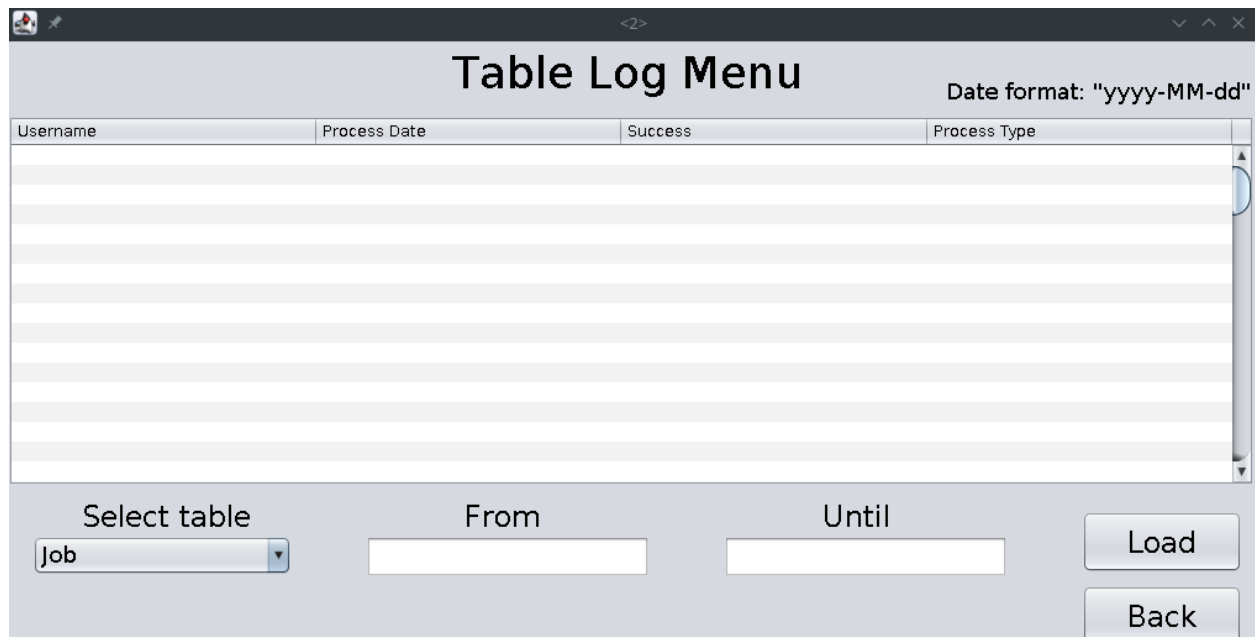
1. Πεδίο εγγραφής ΑΦΜ.
2. Πεδίο εγγραφής ΔΟΥ.
3. Πεδίο εγγραφής Ονόματος.
4. Πεδίο εγγραφής Τηλεφώνου.
5. Πεδίο εγγραφής Οδού.
6. Πεδίο εγγραφής Αριθμού.
7. Πεδίο εγγραφής Πόλης.
8. Πεδίο εγγραφής Χώρας.
9. Κουμπί που δημιουργεί εγγραφή στον πίνακα COMPANYY της ΒΔ.
10. Κουμπί επιστροφής στο κύριο μενού του Admin.

Εμφάνιση Log

Αρχικά όταν πατιέται το κουμπί εμφανίζεται popup με τις επιλογές Table και User. Ανάλογα με την επιλογή εμφανίζεται το μενού Log που αφορά πίνακα ή χρήστη της ΒΔ.



Log που αφορά πίνακα:



Username	Process Date	Success	Process Type

Select table: Job From: Until: Load Back

Log που αφορά χρήστη:

1. Πίνακας που εμφανίζει τα δεδομένα του πίνακα LOG της ΒΔ.
2. Dropdown box που επιλέγουμε τον πίνακα ή τον χρήστη (ανάλογα σε ποιο μενού είμαστε).
3. Πεδίο εγγραφής ημερομηνίας «Από».
4. Πεδίο εγγραφής ημερομηνίας «Έως».
5. Κουμπί που φορτώνει τα δεδομένα ανάλογα με τις επιλογές του χρήστη στον πίνακα 1. Εδώ να σημειώσουμε πως ένα τα πεδία 3, 4 είναι κενά τότε και πάλι θα εμφανιστούν δεδομένα στον πίνακα (με λίγα λόγια τα πεδία λειτουργούν σαν φίλτρα παρά σαν όρια).
6. Κουμπί επιστροφής στο κύριο μενού του Admin.

SQL

CREATE

```
DROP DATABASE if exists PROJECT_BASEIS;
```

```
CREATE DATABASE PROJECT_BASEIS;
```

```
USE PROJECT_BASEIS;
```

```
CREATE TABLE IF NOT EXISTS COMPANY(  
    AFM CHAR(9) DEFAULT 'unknown' NOT NULL,  
    DOY VARCHAR(15) DEFAULT 'unknown' NOT NULL,  
    NAME VARCHAR(35) DEFAULT 'unknown' NOT NULL,  
    PHONE BIGINT(16) DEFAULT '0' NOT NULL,  
    STREET VARCHAR(15) DEFAULT 'unknown' NOT NULL,  
    NUM TINYINT(16) DEFAULT '0' NOT NULL,  
    CITY VARCHAR(15) DEFAULT 'unknown' NOT NULL,  
    COUNTRY VARCHAR(15) DEFAULT 'unknown' NOT NULL,  
    PRIMARY KEY(AFM)  
);
```

```
CREATE TABLE IF NOT EXISTS USERS(  
    USERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,  
    PASSWORD VARCHAR(30) DEFAULT 'unknown' NOT NULL,  
    NAME VARCHAR(25) DEFAULT 'unknown' NOT NULL,  
    SURNAME VARCHAR(35) DEFAULT 'unknown' NOT NULL,  
    REG_DATE DATE NOT NULL DEFAULT CURDATE(),  
    EMAIL VARCHAR(35) DEFAULT 'unknown' NOT NULL,  
    PRIMARY KEY(USERNAME)  
);
```

```
CREATE TABLE IF NOT EXISTS MANAGER(  

```

```

MANAGERUSERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,
EXP_YEARS TINYINT(4) DEFAULT '0' NOT NULL,
FIRM CHAR(9) DEFAULT 'unknown' NOT NULL,
PRIMARY KEY(MANAGERUSERNAME),
CONSTRAINT FIRM_COMPANY FOREIGN KEY(FIRM) REFERENCES
COMPANY(AFM)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT MAN_USER FOREIGN KEY(MANAGERUSERNAME) REFERENCES
USERS(USERNAME)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS EMPLOYEE(
USERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,
AM INT(10) AUTO_INCREMENT NOT NULL, /* AM εταιρείας */
BIO TEXT DEFAULT 'unknown' NOT NULL,
SISTATIKES VARCHAR(255) DEFAULT 'unknown' NOT NULL,
CERTIFICATES VARCHAR(255) DEFAULT 'unknown' NOT NULL,
AWARDS VARCHAR(255) DEFAULT 'unknown' NOT NULL,
LANG SET('EN','FR','SP','GR'), /* Επιλογή πολλαπλών γλωσσών σαν μεταβλητή */
FIRM CHAR(9) DEFAULT 'unknown' NOT NULL, /* ΑΦΜ εταιρείας εργασίας */
EXP_YEARS TINYINT(4) DEFAULT '0' NOT NULL, /* Χρόνια εμπειρίας υπαλλήλου */
PRIMARY KEY(USERNAME),
UNIQUE KEY (AM), /* Κάθε χρήστης έχει δικό του AM αλλά δεν είναι το κύριο γνωρισμά
του */
CONSTRAINT EMPL_USER FOREIGN KEY (USERNAME) REFERENCES
USERS(USERNAME)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT FIRM_EMPLOYEE FOREIGN KEY(FIRM) REFERENCES
COMPANY(AFM)

```


ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS PROJECT(

EMPL VARCHAR(12) DEFAULT 'unknown' NOT NULL,

NUM TINYINT(4) NOT NULL AUTO_INCREMENT,

DESCR TEXT DEFAULT 'unknown' NOT NULL,

URL VARCHAR(60) DEFAULT 'unknown' NOT NULL,

PRIMARY KEY (NUM, EMPL), /* Ξεχωριστός συνδυασμός έτσι ώστε κάθε χρήστης να έχει
δικιά του απαρίθμηση των project του */

CONSTRAINT EMPL_PROJECT FOREIGN KEY (EMPL) REFERENCES
EMPLOYEE(USERNAME)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE IF NOT EXISTS EVALUATOR(

USERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,

ID TINYINT(4) NOT NULL AUTO_INCREMENT,

EXP_YEARS TINYINT(4) DEFAULT '0' NOT NULL,

FIRM CHAR(9) DEFAULT 'unknown' NOT NULL,

PRIMARY KEY (USERNAME),

UNIQUE KEY (ID), /* Κάθε αξιολογητής έχει δικό του αριθμό αλλά δεν είναι το κύριο
γνωρισμά του */

CONSTRAINT EV_USER FOREIGN KEY (USERNAME) REFERENCES
USERS(USERNAME)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT EV_COMPANY FOREIGN KEY (FIRM) REFERENCES COMPANY(AFM)

ON DELETE CASCADE ON UPDATE CASCADE

);

```

CREATE TABLE IF NOT EXISTS JOB(
ID INT(4) NOT NULL AUTO_INCREMENT,
START_DATE DATE NOT NULL,
SALARY FLOAT(6,1) DEFAULT '0' NOT NULL,
POSITION VARCHAR(40) DEFAULT 'unknown' NOT NULL,
EDRA VARCHAR(45) DEFAULT 'unknown' NOT NULL,
EVALUATOR VARCHAR(12) DEFAULT 'unknown' NOT NULL,
ANNOUNCE_DATE DATETIME DEFAULT CURRENT_TIMESTAMP,
SUBMISSION_DATE DATE NOT NULL,
PRIMARY KEY(ID),
CONSTRAINT EV_JOB FOREIGN KEY (EVALUATOR) REFERENCES
EVALUATOR(USERNAME)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS DEGREE(
TITLE VARCHAR(50) DEFAULT 'unknown' NOT NULL,
IDRYMA VARCHAR(40) DEFAULT 'unknown' NOT NULL,
BATHMIDA ENUM('LYKEIO','UNIV','MASTER','PHD'),
PRIMARY KEY(TITLE, IDRYMA)
);

```

```

CREATE TABLE IF NOT EXISTS HAS_DEGREE(
DEGR_TITLE VARCHAR(50) DEFAULT 'unknown' NOT NULL,
DEGR_IDRYMA VARCHAR(40) DEFAULT 'unknown' NOT NULL,
EMPL_USERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,
ETOS YEAR(4) DEFAULT 0 NOT NULL,
GRADE FLOAT(3,1) DEFAULT 0 NOT NULL,
PRIMARY KEY (DEGR_TITLE,DEGR_IDRYMA,EMPL_USERNAME),

```

```

CONSTRAINT DEG_TITLE_IDR FOREIGN KEY (DEGR_TITLE, DEGR_IDRYMA)
REFERENCES DEGREE(TITLE, IDRYMA)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT EMPL_USR FOREIGN KEY (EMPL_USERNAME) REFERENCES
EMPLOYEE(USERNAME)

ON DELETE CASCADE ON UPDATE CASCADE

);

```

```

DROP TABLE IF EXISTS ANTIKEIM;

CREATE TABLE IF NOT EXISTS ANTIKEIM(
TITLE VARCHAR(36) DEFAULT NULL,
DESCR TINYTEXT DEFAULT 'unknown' NOT NULL,
BELONGS_TO VARCHAR(36) DEFAULT NULL,
PRIMARY KEY (TITLE),
CONSTRAINT ANT_BELONGS FOREIGN KEY(BELONGS_TO) REFERENCES
ANTIKEIM(TITLE)

ON DELETE CASCADE ON UPDATE CASCADE

);

```

```

CREATE TABLE IF NOT EXISTS NEEDS(
JOB_ID INT(4) DEFAULT '0' NOT NULL,
ANTIKEIM_TITLE VARCHAR(36) DEFAULT 'unknown' NOT NULL,
PRIMARY KEY (JOB_ID, ANTIKEIM_TITLE),
CONSTRAINT NEED_ID FOREIGN KEY (JOB_ID) REFERENCES JOB(ID)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT NEED_TITLE FOREIGN KEY (ANTIKEIM_TITLE) REFERENCES
ANTIKEIM(TITLE)

ON DELETE CASCADE ON UPDATE CASCADE

);

```

```

CREATE TABLE IF NOT EXISTS LOG(
ID INT(4) NOT NULL AUTO_INCREMENT,
USERNAME VARCHAR(40) DEFAULT 0 NOT NULL, /*Το username που καταχωρείται
είναι αυτό που είναι αποθηκευμένο στη mysql.user*/
PROCESS_DATE DATETIME NOT NULL,
SUCCESS BOOLEAN DEFAULT '0' NOT NULL,
PROCESS_TYPE ENUM('UPDATE','INSERT','DELETE'),
TABLE_NAME ENUM('JOB','EMPLOYEE','SUBMIT APPLICATION'),
PRIMARY KEY (ID)
);

```

```

CREATE TABLE IF NOT EXISTS EVALUATION( /*Πίνακας που αποθηκεύονται όλες οι
αξιολογήσεις ανεξαρτήτως κατάστασης*/
EVALUATION_ID INT(4) NOT NULL AUTO_INCREMENT,
EMPL_USERNAME VARCHAR(12) NOT NULL,
JOB_ID INT(4) NOT NULL,
GRADE_A INT(4) DEFAULT NULL,
COMMENTS VARCHAR(255) DEFAULT 'unknown' NOT NULL,
GRADE_B INT(4) DEFAULT NULL,
GRADE_C INT(4) DEFAULT NULL,
PRIMARY KEY (EVALUATION_ID),
CONSTRAINT EV_JOBID FOREIGN KEY (JOB_ID) REFERENCES JOB(ID)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT EV_EMPL FOREIGN KEY (EMPL_USERNAME) REFERENCES
EMPLOYEE(USERNAME)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE IF NOT EXISTS EVALUATIONRESULT(/*Πίνακας που
αποθηκεύονται οι αξιολογήσεις που έχουν οριστικοποιηθεί*/

```

```

EVID INT(4) DEFAULT '0' NOT NULL,
EMPL_USERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,
JOB_ID INT(4) DEFAULT '0' NOT NULL,
GRADE INT(4) DEFAULT NULL,
COMMENTS VARCHAR(255) DEFAULT 'unknown' NOT NULL,
PRIMARY KEY (EVID),
CONSTRAINT EVA_ID FOREIGN KEY (EVID) REFERENCES
EVALUATION(EVALUATION_ID)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT EVA_EMPL FOREIGN KEY (EMPL_USERNAME) REFERENCES
EMPLOYEE(USERNAME)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT EVA_JOB FOREIGN KEY (JOB_ID) REFERENCES JOB(ID)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

CREATE TABLE IF NOT EXISTS SUBMIT_APPLICATION(/* Αντικατάσταση της
 REQUEST EVALUATION, ο εργαζόμενος κάνει αίτηση για να αξιολογηθεί, αν αλλάξει
 γνώμη διαγράφεται η αίτηση του από τον πίνακα */

```

ID INT(4) NOT NULL AUTO_INCREMENT,
EMPL_USERNAME VARCHAR(12) DEFAULT 'unknown' NOT NULL,
JOB_ID INT(4) DEFAULT '0' NOT NULL,
APPLICATION_DATETIME DATETIME DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY(ID),
CONSTRAINT SU_EMPL FOREIGN KEY (EMPL_USERNAME) REFERENCES
EMPLOYEE(USERNAME)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT SU_JOBID FOREIGN KEY (JOB_ID) REFERENCES JOB(ID)

```

ON DELETE CASCADE ON UPDATE CASCADE

);

INSERTS

INSERT INTO COMPANY VALUES

(264001429, 'A PATRWN', 'Vasilopoulos', '2610222555', 'Ag. Nikolaou', 2, 'Patra', 'Greece'),
(548781409, 'B Athinwn', 'Liakopoulos A.E.', '2104440002', '25hs Martiou', 10, 'Athina', 'Greece'),
(187771638, 'Tripolis', 'Anastasopoulos', '2610333000', 'Ypapanths', 7, 'Tripoli', 'Greece');

INSERT INTO USERS VALUES

('X_Panagiot', 'melenepano', 'Panagioths', 'Xristopoulos', '2020-10-10', 'xpanagiot@gmail.com'),
('mavramos', 'doomiseteral', 'Michalis', 'Avramopoulos', '2020-10-10', 'mixas27@gmail.com'),
('pansot', 'madclip', 'Sotiris', 'Panaikas', '2020-12-15', 'sotpant@outlook.com'),
('anag_xri', '123456789', 'Xristos', 'Anagnostopoulos', '2020-10-15', 'anag_xri@yahoo.com'),
('zaxmar', 'qweasdzxc', 'Maria', 'Zaxariou', '2021-01-10', 'zaxmar@gmail.com'),
('neomar', 'qazwsxedc', 'Maria', 'Neofytou', '2020-11-10', 'neofytou@gmail.com'),
('liakos008', 'grantlembarxos', 'Ilias', 'Mamalakis', '2020-10-20', 'liakos008@gmail.com'),
('gian_gian', 'kappa', 'Giannis', 'Gianakopoulos', '2020-10-10', 'gian@gmail.com'),
('konpit', 'kwdikos', 'Konstantinos', 'Pitakis', '2020-10-25', 'konpit@gmail.com'),
('claud_thom', 'Keitha', 'Claudette', 'Thompson', '2020-11-25', 'claud_thom@hotmail.com'),
('nic_har', 'Danita', 'Nicholas', 'Harris', '2020-10-15', 'nic_har@gmail.com'),
('neil_burns', 'qwertyuiop', 'Neil', 'Burns', '2020-12-10', 'neil_burns@yahoo.com'),
('jak_web', 'kippo', 'Jakob', 'Webb', '2020-11-10', 'jak_web@yahoo.com'),
('bob_jim', 'password', 'Bob', 'Jimenez', '2020-10-12', 'bob_jim@gmail.com'),
('dean_arn', 'krusty', 'Dean', 'Arnold', '2020-10-15', 'dean_arn@gmail.com'),
('mat_john', 'testtest', 'Mathilde', 'Johnston', '2020-12-10', 'mat_john@yahoo.com'),
('loui_will', 'ouioui', 'Louis', 'Williams', '2020-10-10', 'loui_will@gmail.com'),
('fed_cun', 'rockenroll', 'Fedora', 'Cunningham', '2021-01-05', 'fed_cun@hotmail.com'),

```

('cal_pie', 'applepie', 'Calvin', 'Pierce', '2020-10-15', 'cal_pie@yahoo.com'),
('jim_par', 'parpar', 'Jimmie', 'Parsons', '2020-11-25', 'jim_par@gmail.com'),
('cin_wal', 'hm1234', 'Cindy', 'Walters', '2020-12-13', 'cin_wal@webmail.com'),
('ber_pow', 'powpow', 'Bertha', 'Powers', '2020-10-23', 'ber_pow@hotmail.gr'),
('tim_gon', 'gongon', 'Timothy', 'Gonzales', '2020-10-12', 'tim_gon@outlook.com'),
('pat_mar', 'marmar', 'Patrick', 'Martinez', '2020-10-15', 'pat_mar@outlook.com'),
('mel_bai', 'baibai', 'Melissa', 'Bailey', '2020-11-10', 'mel_bai@gmail.com'),
('mat_san', 'sansan', 'Matthew', 'Sanders', '2020-10-15', 'mat_san@gmail.com'),
('dio_sama', 'wryyyy', 'Dio', 'Brando', '2020-10-10', 'dio_sama@gmail.com'),
('jon_joe', 'pluck', 'Jonathan', 'Joestar', '2020-10-10', 'jon_joe@gmail.com'),
('jojo', 'starplantinum', 'Jotaro', 'Kujo', '2020-10-15', 'jojo@gmail.com'),
('ben_pat', 'benben', 'Benjamin', 'Patterson', '2020-10-23', 'ben_pat@gmail.com');

```

INSERT INTO MANAGER VALUES

```

('liakos008', 3, 264001429),
('anag_xri', 4, 264001429),
('zaxmar', 5, 548781409),
('neomar', 4, 548781409),
('gian_gian', 2, 187771638),
('konpit', 3, 187771638);

```

INSERT INTO EVALUATOR VALUES

```

('claud_thom', DEFAULT, 4, 264001429),
('nic_har', DEFAULT, 5, 264001429),
('neil_burns', DEFAULT, 4, 548781409),
('jak_web', DEFAULT, 3, 548781409),
('bob_jim', DEFAULT, 2, 187771638),
('dean_arn', DEFAULT, 4, 187771638);

```

INSERT INTO EMPLOYEE VALUES

('mat_john', DEFAULT, 'Worked for google', 'Amazing', 'No certificates', 'Employee of the month', 'GR,EN', 264001429, 4),

('loui_will', DEFAULT, 'Worked for yahoo', 'Good', '5 certificates', 'Employee of the year', 'GR,EN,SP', 264001429, 2),

('fed_cun', DEFAULT, 'Looking for a good job', 'Newbie', 'No certificates', 'None', 'GR', 264001429, 1),

('cal_pie', DEFAULT, 'Looking for a good job', 'Newbie', 'No certificates', 'None', 'GR', 264001429, 1),

('jim_par', DEFAULT, '1st place in hackathon', 'Amazing programmer', '2 certificates', 'None', 'GR,EN', 264001429, 3),

('cin_wal', DEFAULT, "", "", 'No certificates', 'None', 'GR,EN', 548781409, 1),

('ber_pow', DEFAULT, "", "", '2 certificates', 'None', 'GR,EN,FR', 548781409, 3),

('tim_gon', DEFAULT, 'Trying new goals', "", '2 certificates', 'Employee of the month', 'GR,EN,FR', 548781409, 4),

('pat_mar', DEFAULT, 'Aiming high', 'Top tier', '2 certificates', 'Employee of the year', 'GR,EN,FR', 548781409, 4),

('mel_bai', DEFAULT, '1st job', "", '3 certificates', "", 'GR,EN', 548781409, 1),

('mat_san', DEFAULT, "", "", 'No certificates', 'None', 'GR,EN', 187771638, 1),

('dio_sama', DEFAULT, "", "", '2 certificates', 'None', 'GR,EN,FR', 187771638, 3),

('jon_joe', DEFAULT, 'Trying new goals', "", '2 certificates', 'Employee of the month', 'GR,EN,FR', 187771638, 4),

('jojo', DEFAULT, 'Aiming high', 'Top tier', '2 certificates', 'Employee of the year', 'GR,EN,FR', 187771638, 4),

('ben_pat', DEFAULT, '1st job', "", '3 certificates', "", 'GR,EN', 187771638, 1);

INSERT INTO ANTIKEIM VALUES

('Maths', 'Mathematics', DEFAULT),

('Applied Mathematics', 'Branch of Mathematics', 'Maths'),

('Chemistry', "", DEFAULT),

('Chemical Engineer', 'Chemistry in Industrial space', 'Chemistry'),
('Engineer', DEFAULT, DEFAULT),
('Computer Engineer', 'He is an engineer, for computers', 'Engineer');

INSERT INTO DEGREE VALUES

('Apolythrio Lykeiou', 'Kastritsi', 'LYKEIO'),
('Apolythrio Lykeiou', 'Peiramatiko', 'LYKEIO'),
('Chemistry', 'UOP', 'UNIV'),
('Chemistry', 'UOA', 'UNIV'),
('Chemistry PHD', 'UOP', 'PHD'),
('Chemistry PHD', 'UOA', 'PHD'),
('CEID', 'UOP', 'UNIV');

INSERT INTO HAS_DEGREE VALUES

('Apolythrio Lykeiou', 'Kastritsi', 'mat_john', '2010', 19),
('Apolythrio Lykeiou', 'Kastritsi', 'loui_will', '2012', 20),
('Apolythrio Lykeiou', 'Kastritsi', 'fed_cun', '2013', 15),
('Apolythrio Lykeiou', 'Kastritsi', 'cal_pie', '2011', 16),
('Apolythrio Lykeiou', 'Kastritsi', 'jim_par', '2009', 13.2),
('Apolythrio Lykeiou', 'Kastritsi', 'cin_wal', '2011', 17),
('Apolythrio Lykeiou', 'Peiramatiko', 'ber_pow', '2010', 16),
('Apolythrio Lykeiou', 'Peiramatiko', 'tim_gon', '2011', 15),
('Apolythrio Lykeiou', 'Peiramatiko', 'pat_mar', '2009', 12),
('Apolythrio Lykeiou', 'Peiramatiko', 'mel_bai', '2008', 18),
('Apolythrio Lykeiou', 'Peiramatiko', 'mat_san', '2000', 19),
('Apolythrio Lykeiou', 'Peiramatiko', 'dio_sama', '2001', 20),
('Apolythrio Lykeiou', 'Peiramatiko', 'jon_joe', '2004', 15),
('Apolythrio Lykeiou', 'Peiramatiko', 'jojo', '2000', 13),

('Apolythrio Lykeiou', 'Peiramatiko', 'ben_pat', '2011', 17.5),
 ('Chemistry', 'UOP', 'mat_john', '2015', 8),
 ('Chemistry', 'UOA', 'loui_will', '2014', 6),
 ('CEID', 'UOP', 'fed_cun', '2019', 8),
 ('Chemistry', 'UOP', 'cal_pie', '2015', 8),
 ('Chemistry', 'UOA', 'cin_wal', '2014', 6),
 ('CEID', 'UOP', 'tim_gon', '2019', 8),
 ('Chemistry', 'UOP', 'jim_par', '2015', 8),
 ('Chemistry', 'UOA', 'ber_pow', '2014', 6),
 ('CEID', 'UOP', 'mat_san', '2019', 8);

INSERT INTO PROJECT VALUES

('mat_john', 1, 'Use GitHub GraphQL query to get the file content',
 'https://github.com/conwnet/github1s/issues/12'),
 ('mat_john', 2, 'Design Patterns Library', 'https://github.com/nemanjarogic/DesignPatternsLibrary'),
 ('loui_will', 1, 'Amazon SageMaker Examples', 'https://github.com/aws/amazon-sagemaker-examples'),
 ('fed_cun', 1, 'AI-on-the-edge-device', 'https://github.com/jonjio/ai-on-the-edge-device'),
 ('cal_pie', 1, 'Node.js Best Practices', 'https://github.com/goldbergonyi/nodebestpractices'),
 ('cal_pie', 2, 'react-use', 'https://github.com/streamich/react-use'),
 ('cal_pie', 3, 'clean-code-javascript', 'https://github.com/ryanmcdermott/clean-code-javascript'),
 ('jim_par', 1, 'gaze-detection', 'https://github.com/charliegerard/gaze-detection'),
 ('cin_wal', 1, 'firefox-ios', 'https://github.com/mozilla-mobile/firefox-ios');

INSERT INTO JOB VALUES

(1, '2020-10-10', 800, 'Chemist', 'Patra', 'claud_thom', '2020-9-10', '2020-9-20'),
 (2, '2020-10-10', 800, 'Chemist', 'Athina', 'neil_burns', '2020-9-10', '2020-9-20'),
 (3, '2020-10-10', 800, 'Chemist', 'Tripoli', 'bob_jim', '2020-9-10', '2020-9-20'),

(4, '2020-10-15', 500, 'Cleaner', 'Patra', 'claud_thom', '2020-9-15', '2020-9-25'),
(5, '2021-03-15', 600, 'Accountant', 'Patra', 'claud_thom', '2021-1-10', '2021-3-20'),
(6, '2021-03-15', 900, 'Engineer', 'Patra', 'claud_thom', '2021-1-10', '2021-3-20'),
(7, '2021-03-15', 1000, 'Programmer', 'Patra', 'claud_thom', '2021-1-10', '2021-3-20'),
(8, '2021-03-15', 800, 'Chemical Engineer', 'Patra', 'claud_thom', '2021-1-10', '2021-3-20'),
(9, '2021-03-15', 800, 'Assistant Manager', 'Patra', 'nic_har', '2021-1-10', '2021-3-20'),
(10, '2021-03-15', 700, 'Support', 'Patra', 'nic_har', '2021-1-10', '2021-3-20');

INSERT INTO NEEDS VALUES

(1, 'Chemistry'),
(2, 'Chemistry'),
(3, 'Chemistry'),
(1, 'Maths'),
(2, 'Maths'),
(3, 'Maths'),
(5, 'Maths'),
(6, 'Engineer'),
(7, 'Computer Engineer'),
(8, 'Chemical Engineer');

INSERT INTO SUBMIT_APPLICATION VALUES

(1, 'mat_john', 1, '2020-9-15'),
(2, 'loui_will', 1, '2020-9-15'),
(3, 'fed_cun', 1, '2020-9-15'),
(4, 'cal_pie', 5, '2021-2-20'),
(5, 'mat_john', 9, '2021-2-20');

INSERT INTO EVALUATION VALUES

```
(1, 'mat_john', 1, 3, 'Very good', 3, 2),  
(2, 'loui_will', 1, 1, 'Not good', 1, 0),  
(3, 'fed_cun', 1, 4, 'Excellent', 4, 2),  
(4, 'cal_pie', 4, 4, DEFAULT, DEFAULT, 2);
```

INSERT INTO EVALUATIONRESULT VALUES

```
(1, 'mat_john', 1, 8, 'Very good'),  
(2, 'loui_will', 1, 2, 'Not good'),  
(3, 'fed_cun', 1, 10, 'Excellent');
```

```
CREATE USER 'X_Panagiot'@'localhost' IDENTIFIED BY 'melenepano';  
CREATE USER 'mavramos'@'localhost' IDENTIFIED BY 'doomiseternal';  
CREATE USER 'pansot'@'localhost' IDENTIFIED BY 'madclip';  
CREATE USER 'anag_xri'@'localhost' IDENTIFIED BY '123456789';  
CREATE USER 'zaxmar'@'localhost' IDENTIFIED BY 'qweasdzxc';  
CREATE USER 'neomar'@'localhost' IDENTIFIED BY 'qazwsxedc';  
CREATE USER 'liakos008'@'localhost' IDENTIFIED BY 'grantlembarxos';  
CREATE USER 'gian_gian'@'localhost' IDENTIFIED BY 'kappa';  
CREATE USER 'konpit'@'localhost' IDENTIFIED BY 'kwdikos';  
CREATE USER 'claud_thom'@'localhost' IDENTIFIED BY 'Keitha';  
CREATE USER 'nic_har'@'localhost' IDENTIFIED BY 'Danita';  
CREATE USER 'neil_burns'@'localhost' IDENTIFIED BY 'qwertyuiop';  
CREATE USER 'jak_web'@'localhost' IDENTIFIED BY 'kippo';  
CREATE USER 'bob_jim'@'localhost' IDENTIFIED BY 'password';  
CREATE USER 'dean_arn'@'localhost' IDENTIFIED BY 'krusty';  
CREATE USER 'mat_john'@'localhost' IDENTIFIED BY 'testtest';  
CREATE USER 'loui_will'@'localhost' IDENTIFIED BY 'ouioui';
```

```

CREATE USER 'fed_cun'@'localhost' IDENTIFIED BY 'rockenroll';
CREATE USER 'cal_pie'@'localhost' IDENTIFIED BY 'applepie';
CREATE USER 'jim_par'@'localhost' IDENTIFIED BY 'parpar';
CREATE USER 'cin_wal'@'localhost' IDENTIFIED BY 'hm1234';
CREATE USER 'ber_pow'@'localhost' IDENTIFIED BY 'powpow';
CREATE USER 'tim_gon'@'localhost' IDENTIFIED BY 'gongon';
CREATE USER 'pat_mar'@'localhost' IDENTIFIED BY 'marmar';
CREATE USER 'mel_bai'@'localhost' IDENTIFIED BY 'baibai';
CREATE USER 'mat_san'@'localhost' IDENTIFIED BY 'sansan';
CREATE USER 'dio_sama'@'localhost' IDENTIFIED BY 'wryyyy';
CREATE USER 'jon_joe'@'localhost' IDENTIFIED BY 'pluck';
CREATE USER 'jojo'@'localhost' IDENTIFIED BY 'starplantinum';
CREATE USER 'ben_pat'@'localhost' IDENTIFIED BY 'benben';

```

```

GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'X_Panagiot'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'mavramos'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'pansot'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'anag_xri'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'zaxmar'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'neomar'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'liakos008'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'gian_gian'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'konpit'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'claud_thom'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'nic_har'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'neil_burns'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'jak_web'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS.* to 'bob_jim'@'localhost';

```

```
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'dean_arn'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'mat_john'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'loui_will'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'fed_cun'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'cal_pie'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'jim_par'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'cin_wal'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'ber_pow'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'tim_gon'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'pat_mar'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'mel_bai'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'mat_san'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'dio_sama'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'jon_joe'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'jojo'@'localhost';
GRANT ALL PRIVILEGES ON PROJECT_BASEIS. * to 'ben_pat'@'localhost';
```

```
FLUSH PRIVILEGES;
```

STORED PROCEDURES

```
DELIMITER $
```

```
DROP PROCEDURE IF EXISTS employee_applications$
```

```
CREATE PROCEDURE employee_applications(IN in_name VARCHAR(25), IN in_surname  
VARCHAR(35))
```

```
BEGIN
```

```
DECLARE emp_username VARCHAR(12);
```

```
SELECT USERNAME INTO emp_username FROM USERS WHERE NAME=in_name  
AND SURNAME=in_surname;
```

```
SELECT NAME, SURNAME, JOB_ID AS 'ID', POSITION, EDRA /* Εκτύπωση αιτήσεων  
εργαζομένου */
```

```
FROM SUBMIT_APPLICATION
```

```
INNER JOIN EMPLOYEE ON EMPL_USERNAME = USERNAME
```

```
INNER JOIN USERS ON EMPL_USERNAME=USERS.USERNAME
```

```
INNER JOIN JOB ON SUBMIT_APPLICATION.JOB_ID = JOB.ID
```

```
WHERE EMPL_USERNAME=emp_username AND SUBMISSION_DATE > CURDATE();
```

```
SELECT EVID, USERS.NAME, USERS.SURNAME, JOB_ID, POSITION, EDRA, GRADE  
/* Εκτύπωση τελικών αξιολογήσεων εργαζομένου */
```

```
FROM EVALUATIONRESULT
```

```
INNER JOIN EMPLOYEE ON EMPL_USERNAME = USERNAME
```

```
INNER JOIN JOB ON EVALUATIONRESULT.JOB_ID = JOB.ID
```

```
INNER JOIN EVALUATOR ON EVALUATOR.USERNAME = JOB.EVALUATOR
```

```
INNER JOIN USERS ON
```

```
EVALUATIONRESULT.EMPL_USERNAME=USERS.USERNAME
```

```
WHERE EMPL_USERNAME=emp_username;
```

```
SELECT 'EVALUATOR';
```

```
SELECT EVID, NAME, SURNAME
```

```
FROM USERS
```

```
INNER JOIN EVALUATOR ON EVALUATOR.USERNAME = USERS.USERNAME
```

```
/* Εκτύπωση αντίστοιχων αξιολογητών */
```

```
INNER JOIN JOB ON JOB.EVALUATOR = EVALUATOR.USERNAME
```

```
INNER JOIN EVALUATIONRESULT ON JOB_ID = JOB.ID
```

```
WHERE EMPL_USERNAME = emp_username;
```

```
SELECT 'EVALUATION IN PROCESS';
```

```
SELECT EVALUATION_ID, USERS.NAME, USERS.SURNAME /* Εκτύπωση των
αξιολογήσεων που βρίσκονται σε εξέλιξη */
FROM EVALUATION
INNER JOIN EMPLOYEE ON EMPL_USERNAME = EMPLOYEE.USERNAME
INNER JOIN JOB ON EVALUATION.JOB_ID = JOB.ID
INNER JOIN EVALUATOR ON EVALUATOR.USERNAME = JOB.EVALUATOR
INNER JOIN USERS ON EVALUATION.EMPL_USERNAME=USERS.USERNAME
LEFT JOIN EVALUATIONRESULT ON EVALUATION.EVALUATION_ID =
EVALUATIONRESULT.EVID
WHERE EVALUATION.EMPL_USERNAME=emp_username AND
EVALUATIONRESULT.EVID IS NULL;
```

```
END$
```

```
DELIMITER ;
```

```
DELIMITER $
```

```
DROP PROCEDURE IF EXISTS evaluation_checker$ /*Έλεγχος αξιολογήσεων και
οριστικοποίηση τους*/
CREATE PROCEDURE evaluation_checker(IN in_job_id INT(4), IN in_evaluator_id INT(4))
BEGIN
DECLARE ev_id INT(4);
DECLARE emp_username VARCHAR(12);
```



```

DECLARE grade_sum INT(4);
DECLARE comm VARCHAR(255);
DECLARE g_a INT(4);
DECLARE g_b INT(4);
DECLARE g_c INT(4);
DECLARE evflag INT;
DECLARE evaluationCursor CURSOR FOR
SELECT EVALUATION_ID, EMPL_USERNAME, GRADE_A, GRADE_B, GRADE_C,
COMMENTS FROM EVALUATION /* Επιλέγουμε όλες τις αξιολογήσεις του αξιολογητή
στην δοσμένη θέση εργασίας */
INNER JOIN JOB ON EVALUATION.JOB_ID=JOB.ID
INNER JOIN EVALUATOR ON JOB.EVALUATOR=EVALUATOR.USERNAME
WHERE in_job_id=EVALUATION.JOB_ID AND in_evaluator_id=EVALUATOR.ID;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET evflag=1;
OPEN evaluationCursor;
SET evflag=0;
WHILE (evflag=0) DO
FETCH evaluationCursor INTO ev_id, emp_username, g_a, g_b, g_c, comm;
SET grade_sum=0;
IF(g_a IS NOT NULL AND g_b IS NOT NULL AND g_c IS NOT NULL) THEN /* Εάν ένας
βαθμός είναι null σημαίνει πως η αξιολόγηση βρίσκεται σε εξέλιξη άρα ελέγχουμε τις
αξιολογήσεις που έχουν βαθμούς */
SET grade_sum = g_a + g_b + g_c; /* Υπολογισμός αθροίσματος */
INSERT IGNORE INTO EVALUATIONRESULT VALUES(ev_id, emp_username,
in_job_id, grade_sum, comm); /* Οριστικοποιούμε τις αξιολογήσεις κάνοντας Insert στον
πίνακα EVALUATIONRESULT, αν υπάρχει ήδη κάποια αξιολόγηση που έχει
οριστικοποιηθεί την αγνοούμε με την εντολή IGNORE */
END IF;
END WHILE;
CLOSE evaluationCursor;

```

END\$

DELIMITER ;

DELIMITER \$

DROP PROCEDURE IF EXISTS job_checker\$

CREATE PROCEDURE job_checker(IN in_id INT(4))

BEGIN

DECLARE gr_a INT(4);

DECLARE gr_b INT(4);

DECLARE gr_c INT(4);

DECLARE gr_count INT;

DECLARE eva_count INT(4);

DECLARE gradeflag INT;

DECLARE user_name VARCHAR(25);

DECLARE user_surname VARCHAR(35);

DECLARE ev_id INT(4);

DECLARE gradeCursor1 CURSOR FOR

SELECT EVALUATION_ID, NAME, SURNAME, GRADE_A, GRADE_B, GRADE_C
FROM EVALUATION INNER JOIN USERS ON
EMPL_USERNAME=USERS.USERNAME WHERE JOB_ID=in_id;

/* Σε περίπτωση που έχουμε αξιολογήσεις που δεν έχουν οριστικοποιηθεί χρησιμοποιείται
ο gradeCursor1 για να μας εμφανίσει τις ολοκληρωμένες αξιολογήσεις */

```
DECLARE gradeCursor2 CURSOR FOR
```

```
SELECT EVALUATION_ID, NAME, SURNAME, GRADE_A, GRADE_B, GRADE_C  
FROM EVALUATION INNER JOIN USERS ON  
EMPL_USERNAME=USERS.USERNAME WHERE JOB_ID=in_id;
```

```
/* Σε περίπτωση που έχουμε αξιολογήσεις που δεν έχουν οριστικοποιηθεί χρησιμοποιείται  
ο gradeCursor2 για να μας εμφανίσει τις αξιολογήσεις που εκκρεμούν */
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET gradeflag=1;
```

```
SET gradeflag=0;
```

```
SELECT COUNT(*) INTO gr_count FROM EVALUATION WHERE JOB_ID=in_id AND  
(GRADE_A IS NULL OR GRADE_B IS NULL OR GRADE_C IS NULL);
```

```
/* Καταμέτρηση αξιολογήσεων που εκκρεμούν σε μία θέση εργασίας */
```

```
SELECT COUNT(*) INTO eva_count FROM EVALUATION WHERE JOB_ID=in_id;
```

```
/* Καταμέτρηση όλων των αξιολογήσεων μίας θέσης εργασίας */
```

```
IF(eva_count!=0) THEN /* Έλεγχος αν υπάρχουν αξιολογήσεις σε μία θέση εργασίας  
(Οριστικοποιημένες και μη) */
```

```
IF(gr_count=0) THEN /* Έλεγχος αν υπάρχουν αξιολογήσεις που δεν είναι  
οριστικοποιημένες */
```

```
/* Περίπτωση που όλες οι αξιολογήσεις είναι οριστικοποιημένες */
```

```
SELECT 'FINALIZED TABLES';
```

```
SELECT EVID AS ID, NAME, SURNAME, GRADE
```

```
FROM EVALUATIONRESULT
```

```
INNER JOIN USERS ON EMPL_USERNAME=USERS.USERNAME
```

```
WHERE EMPL_USERNAME=USERS.USERNAME AND JOB_ID=in_id
```

```
ORDER BY GRADE DESC;
```

```
ELSE
```

```

/* Περίπτωση που υπάρχουν αξιολογήσεις που εκκρεμούν */
SELECT 'COMPLETED EVALUATIONS';

OPEN gradeCursor1;

FETCH gradeCursor1 INTO ev_id, user_name, user_surname, gr_a, gr_b, gr_c;

WHILE(gradeFlag=0) DO

    IF(gr_a IS NOT NULL AND gr_b IS NOT NULL AND gr_c IS NOT NULL) THEN

        SELECT ev_id AS 'ID', user_name AS 'NAME', user_surname AS 'SURNAME', gr_a +
gr_b + gr_c AS 'GRADE'; /* Εκτύπωση ολοκληρωμένων αξιολογήσεων */

        END IF;

        FETCH gradeCursor1 INTO ev_id, user_name, user_surname, gr_a, gr_b, gr_c;

    END WHILE;

CLOSE gradeCursor1;

SET gradeFlag=0; /* Επαναφορά του flag για να χρησιμοποιηθεί στον gradeCursor2 */

SELECT 'EVALUATIONS IN PROCESS';

OPEN gradeCursor2;

FETCH gradeCursor2 INTO ev_id, user_name, user_surname, gr_a, gr_b, gr_c;

WHILE(gradeFlag=0) DO

    IF (gr_a IS NULL OR gr_b IS NULL OR gr_c IS NULL) THEN

        SELECT ev_id AS 'ID', user_name AS 'NAME', user_surname AS 'SURNAME', gr_a +
gr_b + gr_c AS 'GRADE' /* Εκτύπωση αξιολογήσεων που εκκρεμούν */

        ORDER BY ev_id ASC;

        END IF;

        FETCH gradeCursor2 INTO ev_id, user_name, user_surname, gr_a, gr_b, gr_c;

    END WHILE;

CLOSE gradeCursor2;

SELECT COUNT(*) AS 'NUMBER OF EVALUATIONS IN PROCESS' FROM
EVALUATION WHERE GRADE_A + GRADE_B + GRADE_C IS NULL; /* Εκτύπωση
αριθμού αξιολογήσεων που εκκρεμούν */

END IF;

```

ELSE

SELECT 'NO CANDIDATES FOR THIS POSITION'; /* Εκτύπωση σε περίπτωση που δεν υπάρχουν αξιολογήσεις για μία θέση εργασίας (eva_count=0) */

END IF;

END\$

DELIMITER ;

EXTRA STORED PROCEDURE

/* Procedures που είναι απαραίτητα για την λειτουργία του gui */

DELIMITER \$

DROP PROCEDURE IF EXISTS average_grade_evaluator\$ /* Υπολογισμός μέσου όρου βαθμολόγησης ενός αξιολογητή */

CREATE PROCEDURE average_grade_evaluator(IN in_evaluator VARCHAR(12), OUT result FLOAT(5,3))

BEGIN

DECLARE ev_grade FLOAT(5,3);

DECLARE grade_sum FLOAT(5,3);

DECLARE i FLOAT(5,3);

DECLARE flagdone INT;

DECLARE averageCursor CURSOR FOR

SELECT GRADE + 0.0 FROM EVALUATIONRESULT INNER JOIN JOB ON
EVALUATIONRESULT.JOB_ID = JOB.ID INNER JOIN EVALUATOR ON
JOB.EVALUATOR = EVALUATOR.USERNAME WHERE in_evaluator = USERNAME;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET flagdone = 1;

OPEN averageCursor;

SET grade_sum = 0;

SET i = 0;

SET flagdone = 0;

```

FETCH averageCursor INTO ev_grade;

WHILE(flagdone = 0) DO
    SET grade_sum = grade_sum + ev_grade;
    SET i = i + 1;
    SELECT ev_grade, grade_sum, i;
    FETCH averageCursor INTO ev_grade;
END WHILE;

CLOSE averageCursor;

SET result = grade_sum / i;

END$

```

```

DELIMITER ;

```

TRIGGERS

/* Οι Triggers με όνομα X_YZ είναι για τον πίνακα log, όπου X το είδος επεξεργασίας, Y ο πίνακας που υπέστη επεξεργασία

και Z το success που γράφεται. Πριν γίνει η επεξεργασία σε κάποιο πίνακα αποθηκεύουμε το όνομα του χρήστη μέσω της

συνάρτησης USER() που επιστρέφει τον συνδεδεμένο χρήστη στην Βάση δεδομένων μας από τον πίνακα mysql.USER.

Έπειτα αποθηκεύουμε την ώρα και ημερομηνία, το είδος επεξεργασίας και τον πίνακα που υπέστη επεξεργασία.

Θέτουμε το SUCCESS false γιατί η εισαγωγή στο log γίνεται πριν γίνει κάποια επεξεργασία, με λίγα λόγια αποθηκεύουμε

την απόπειρα. Αν γίνει όντως η επεξεργασία ενεργοποιείται ο trigger με Z = 1 που είναι τύπου λ.χ. after insert και θέτουμε το SUCCESS true. */

```

DROP TRIGGER IF EXISTS INSERT_EMPLOYEE0;

DELIMITER $

```

```
CREATE TRIGGER INSERT_EMPLOYEE0 BEFORE INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
INSERT INTO LOG VALUES(DEFAULT,
USER(),CURRENT_TIMESTAMP(),0,'INSERT','EMPLOYEE');
END$
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS INSERT_EMPLOYEE1;
DELIMITER $
CREATE TRIGGER INSERT_EMPLOYEE1 AFTER INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
DECLARE idmax INT(4);
SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();
UPDATE LOG
SET SUCCESS = 1
WHERE USERNAME=USER() AND ID = idmax;
END $
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS INSERT_JOB0;
DELIMITER $
CREATE TRIGGER INSERT_JOB0 BEFORE INSERT ON JOB
FOR EACH ROW
BEGIN
INSERT INTO LOG
VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'INSERT','JOB');
```

```
END $  
DELIMITER ;  
  
DROP TRIGGER IF EXISTS INSERT_JOB1;  
DELIMITER $  
CREATE TRIGGER INSERT_JOB1 AFTER INSERT ON JOB  
FOR EACH ROW  
BEGIN  
DECLARE idmax INT(4);  
SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();  
UPDATE LOG  
SET SUCCESS ='1'  
WHERE USERNAME=USER() AND ID = idmax;  
END $  
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS INSERT_SUBMIT_APPLICATION0;  
DELIMITER $  
CREATE TRIGGER INSERT_SUBMIT_APPLICATION0 BEFORE INSERT ON  
SUBMIT_APPLICATION  
FOR EACH ROW  
BEGIN  
INSERT INTO LOG  
VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'INSERT','SUBMIT  
APPLICATION');  
END $  
DELIMITER ;
```



```

DROP TRIGGER IF EXISTS INSERT_SUBMIT_APPLICATION1;

DELIMITER $

CREATE TRIGGER INSERT_SUBMIT_APPLICATION1 AFTER INSERT ON
SUBMIT_APPLICATION

FOR EACH ROW

BEGIN

DECLARE idmax INT(4);

SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();

UPDATE LOG

SET SUCCESS ='1'

WHERE USERNAME=USER() AND ID = idmax;

END $

DELIMITER ;

```

```

DROP TRIGGER IF EXISTS DELETE_SUBMIT_APPLICATION0;

DELIMITER $

CREATE TRIGGER DELETE_SUBMIT_APPLICATION0 BEFORE DELETE ON
SUBMIT_APPLICATION

FOR EACH ROW

BEGIN

INSERT INTO LOG

VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'DELETE','SUBMIT
APPLICATION');

END $

DELIMITER ;

```

```

DROP TRIGGER IF EXISTS DELETE_SUBMIT_APPLICATION1;

```

DELIMITER \$

CREATE TRIGGER DELETE_SUBMIT_APPLICATION1 AFTER DELETE ON
SUBMIT_APPLICATION

FOR EACH ROW

BEGIN

DECLARE idmax INT(4);

SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();

UPDATE LOG

SET SUCCESS ='1'

WHERE USERNAME=USER() AND ID = idmax;

END \$

DELIMITER ;

DROP TRIGGER IF EXISTS DELETE_EMPLOYEE0;

DELIMITER \$

CREATE TRIGGER DELETE_EMPLOYEE0 BEFORE DELETE ON EMPLOYEE

FOR EACH ROW

BEGIN

INSERT INTO LOG

VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'DELETE','EMPLOYEE');

END \$

DELIMITER ;

DROP TRIGGER IF EXISTS DELETE_EMPLOYEE1;

DELIMITER \$

CREATE TRIGGER DELETE_EMPLOYEE1 AFTER DELETE ON EMPLOYEE

FOR EACH ROW

BEGIN

```
DECLARE idmax INT(4);  
SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();  
UPDATE LOG  
SET SUCCESS ='1'  
WHERE USERNAME=USER() AND ID = idmax;  
END $  
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS DELETE_JOB0;  
DELIMITER $  
CREATE TRIGGER DELETE_JOB0 BEFORE DELETE ON JOB  
FOR EACH ROW  
BEGIN  
INSERT INTO LOG  
VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'DELETE',JOB);  
END $  
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS DELETE_JOB1;  
DELIMITER $  
CREATE TRIGGER DELETE_JOB1 AFTER DELETE ON JOB  
FOR EACH ROW  
BEGIN  
DECLARE idmax INT(4);  
SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();  
UPDATE LOG
```

```
SET SUCCESS ='1'  
WHERE USERNAME=USER() AND ID = idmax;  
END $  
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS UPDATE_JOB0;  
DELIMITER $  
CREATE TRIGGER UPDATE_JOB0 BEFORE UPDATE ON JOB  
FOR EACH ROW  
BEGIN  
INSERT INTO LOG  
VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'UPDATE',JOB);  
END $  
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS UPDATE_JOB1;  
DELIMITER $  
CREATE TRIGGER UPDATE_JOB1 AFTER UPDATE ON JOB  
FOR EACH ROW  
BEGIN  
DECLARE idmax INT(4);  
SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();  
UPDATE LOG  
SET SUCCESS ='1'  
WHERE USERNAME=USER() AND ID = idmax;
```

END \$

DELIMITER ;

DROP TRIGGER IF EXISTS UPDATE_EMPLOYEE0;

DELIMITER \$

CREATE TRIGGER UPDATE_EMPLOYEE0 BEFORE UPDATE ON EMPLOYEE

FOR EACH ROW

BEGIN

INSERT INTO LOG

VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'UPDATE','EMPLOYEE');

END \$

DELIMITER ;

DROP TRIGGER IF EXISTS UPDATE_EMPLOYEE1;

DELIMITER \$

CREATE TRIGGER UPDATE_EMPLOYEE1 AFTER UPDATE ON EMPLOYEE

FOR EACH ROW

BEGIN

DECLARE idmax INT(4);

SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();

UPDATE LOG

SET SUCCESS ='1'

WHERE USERNAME=USER() AND ID = idmax;

END \$

DELIMITER ;

```
DROP TRIGGER IF EXISTS UPDATE_SUBMIT_APPLICATION0;
DELIMITER $
CREATE TRIGGER UPDATE_SUBMIT_APPLICATION0 BEFORE UPDATE ON
SUBMIT_APPLICATION
FOR EACH ROW
BEGIN
INSERT INTO LOG
VALUES(DEFAULT,USER(),CURRENT_TIMESTAMP(),0,'UPDATE','SUBMIT
APPLICATION');
END $
DELIMITER ;
```

```
DROP TRIGGER IF EXISTS UPDATE_SUBMIT_APPLICATION1;
DELIMITER $
CREATE TRIGGER UPDATE_SUBMIT_APPLICATION1 AFTER UPDATE ON
SUBMIT_APPLICATION
FOR EACH ROW
BEGIN
DECLARE idmax INT(4);
SELECT MAX(ID) INTO idmax FROM LOG WHERE USERNAME=USER();
UPDATE LOG
SET SUCCESS ='1'
WHERE USERNAME=USER() AND ID = idmax;
END $
DELIMITER ;
```

```

DROP TRIGGER IF EXISTS UPDATE_COMPANY; /* Έλεγχος αν γίνεται απόπειρα
αλλαγής AFM, DOY, NAME */

DELIMITER $

CREATE TRIGGER UPDATE_COMPANY BEFORE UPDATE ON COMPANY
FOR EACH ROW
BEGIN
IF (OLD.AFM != NEW.AFM) THEN
SET NEW.AFM = OLD.AFM; /* Αν το AFM διαφέρει, βάλε το παλιό στο UPDATE */
END IF;
IF (OLD.DOY != NEW.DOY) THEN
SET NEW.DOY = OLD.DOY; /* Αν η DOY διαφέρει, βάλε το παλιό στο UPDATE */
END IF;
IF (NEW.NAME != OLD.NAME) THEN
SET NEW.NAME = OLD.NAME; /* Αν το NAME διαφέρει, βάλε το παλιό στο UPDATE */
END IF;
END $

DELIMITER ;

```

```

DROP TRIGGER IF EXISTS ADMIN_CHECK; /* Έλεγχος αν ο χρήστης είναι
διαχειριστής */

DELIMITER $

CREATE TRIGGER ADMIN_CHECK BEFORE UPDATE ON USERS
FOR EACH ROW
BEGIN
DECLARE usr VARCHAR(30);
DECLARE usrn timer VARCHAR(30);
DECLARE usrcheck VARCHAR(30);

```

```

SELECT USER() INTO usr; /* Αποθηκεύουμε τον συνδεδεμένο χρήστη, είναι της μορφής
USERNAME@localhost */

SET usrnw = REPLACE (usr, '@localhost', ''); /* Αφαιρούμε το κομμάτι '@localhost' και
αποθηκεύουμε στη μεταβλητή usrnw μόνο το USERNAME */

SELECT USERS.USERNAME INTO usrcheck

FROM USERS

LEFT JOIN MANAGER ON USERS.USERNAME = MANAGER.USERNAME

LEFT JOIN EMPLOYEE ON EMPLOYEE.USERNAME = USERS.USERNAME

LEFT JOIN EVALUATOR ON EVALUATOR.USERNAME = USERS.USERNAME

WHERE MANAGER.USERNAME IS NULL AND EMPLOYEE.USERNAME IS NULL
AND EVALUATOR.USERNAME IS NULL AND USERS.USERNAME = usrnw;

/* Αν το USERNAME υπάρχει σε έναν από τους πίνακες MANAGER, EMPLOYEE ή
EVALUATOR τότε ο χρήστης δεν είναι διαχειριστής */

IF (NEW.USERNAME != OLD.USERNAME) THEN /* Ελέγχουμε αν έγινε απόπειρα
αλλαγής του USERNAME */

    IF(usrcheck IS NULL) THEN /* Ελέγχουμε αν τελικά ο χρήστης είναι διαχειριστής */

        SIGNAL SQLSTATE VALUE '45000'

        SET MESSAGE_TEXT = 'ONLY ADMIN CAN UPDATE USERNAME'; /* Εκτύπωση
μηνύματος στην περίπτωση που δεν είναι */

    END IF;

END IF;

END $

DELIMITER ;

```

EXTRA TRIGGER

--Trigger που ελέγχουμε αν το email είναι σε κατάλληλη μορφή

DELIMITER \$

CREATE TRIGGER emailCheck


```
BEFORE UPDATE ON USERS
FOR EACH ROW
BEGIN
IF NEW.EMAIL NOT LIKE '%@%.%' THEN
    SET NEW.email=OLD.email;
    SIGNAL SQLSTATE VALUE '45000'
    SET MESSAGE_TEXT = 'New email not in correct form!';
END IF;
END$
DELIMITER ;
```

Σχεσιακό

