

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ ΣΤΟ ΕΡΓΑΣΤΗΡΙΟ ΤΟΥ ΜΑΘΗΜΑΤΟΣ “ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ”

ΚΟΛΛΙΑΣ ΙΩΑΝΝΗΣ

1064886

Ακολουθεί η περιγραφή του κώδικα. Οι προσθέσεις και αλλαγές της βάσης φαίνονται παρακάτω.

Αρχικά καλείται το Procedure LoginAccount που φαίνεται τι δικαιώματα έχει ο κάθε user στη βάση δεδομένων. Με τα current\_username και current\_userkind στο login αυτόματα βρίσκω πού κατατάσσεται ο εκάστοτε χρήστης. Ο Administrator έχει ήδη τοποθετηθεί στη βάση και έχει τα περισσότερα δικαιώματα. Το current\_username χρησιμοποιείται κυρίως στη Log ενώ το current\_userkind περιορίζει την πρόσβαση σε όποια είδη χρηστών δεν έχουν τα δικαιώματα να αλλάξουν στοιχεία της βάσης, μέσω trigger. Συγκεκριμένα τα trigger Username Change και UnchangeableColumns είναι αυτά που δημιουργούν τους περιορισμούς αλλαγής στοιχείων. Οι δύο παραπάνω μεταβλητές είναι session variable, δηλαδή παραμένουν ενεργές έως ότου κλείσουμε τη σύνδεση ή ξανακαλέσουμε την “LoginAccount”.

Στον πίνακα user πρόσθεσα τη στήλη userkind έτσι ώστε να είναι πιο εύκολο να βρούμε κάποιο username και να μην ψάχνουμε στον κάθε πίνακα ξεχωριστά (employee, evaluator κ.λ.π).

Για τη στήλη edra χρησιμοποίησα την συνάρτηση CONCAT έτσι ώστε να προστεθούν τα διάφορα strings σε μία στήλη (country, city, street, num). Τέλος αναφορικά με το table company μέσω του trigger UnchangeableColumns κρατά αναλλοίωτες τις αλλαγές σε AFM,DOY και compname απ’ όποιον προσπαθεί να τις αλλάξει και δεν είναι administrator. Δυστυχώς η στήλη edra δεν λειτούργησε ως FK παρά μόνο στο xampp, ακριβώς επειδή έχω προσπαθήσει και το gui βρίσκεται στον κώδικά μου αλλά σε σχόλιο.

Μέσω του Procedure Average φαίνονται στη στήλη avr\_grade οι βαθμολογίες που οριστικοποιήθηκαν από αναφορές του συγκεκριμένου αξιολογητή. Το procedure αυτό παίρνει το username του evaluator απ’ όλα τα request\_evaluation σε job υπό την εποπτεία του και αποθηκεύεται στο avr\_grade.

**ΣΗΜΕΙΩΣΗ:** Πριν δείτε στο Select, τρέξτε τη Average ώστε να φανεί ο καινούριος μέσος όρος.

Στο table has\_degree περιέχει ένα πτυχίο που μπορεί να έχει ένας χρήστης. Ο κάθε employee μπορεί να έχει πολλά πτυχία. Απ’ τη στιγμή που το primary key είναι το empl\_username, titlos, idryma δεν τίθεται ζήτημα μοναδικότητας.

Απ’ την άλλη στο degree φαίνεται τι πτυχία έχουν οι χρήστες που εξετάζονται και πόσοι έχουν το ίδιο πτυχίο. Έτσι σε αυτό το table προστέθηκε και to column numgraduates.

Θα παρατηρήσετε πως δεν υπάρχουν τα constraint για τα δύο tables και αυτό γιατί δημιουργούνταν πολλές συγκρούσεις αναφορικά με τη μοναδικότητα των κλειδιών. Για αυτό χρησιμοποιώ triggers insert, delete, update. Το degree παίρνει από το insert του has\_degree, οπότε δεν ενημερώνεται απ’ το χρήστη. Αν υπάρχει ήδη τιμή στο degree με ίδιο titlos και idryma, τότε numgraduates +. Αν γίνει delete ένα στοιχείο του has\_degree, το numgraduates θα μειωθεί κατά 1. Αν γίνει update θα αφαιρεθεί ένα στα OLD titlos και idryma και θα προστεθεί ένα στα NEW titlos και idryma.

Στο table job, η στήλη job\_id έχει οριστεί με auto\_increment για να παίρνει αυτόματα κάποιον αριθμό ως id της δουλειάς. Το ίδιο ισχύει και με το announcedate. Υπάρχει Trigger για την επιτυχή ενημέρωση στο log.

Το SendAvailability trigger χρησιμοποιείται έτσι ώστε όταν προστεθεί καινούρια δουλειά, προσθέτει για κάθε εργαζόμενο που ανήκει στην εταιρία όπου δημιουργήθηκε η δουλειά, ένα insert στον πίνακα request. Απ’ την άλλη μεριά στο SAUpdatedDate αν αλλάξει το SubmissionDate ενημερώνει κάθε στοιχείο του requestevaluation από το συγκεκριμένο id\_job.

Στο requestevaluation έχω προσθέσει μία Boolean στήλη ( empl\_interest ) για τη δήλωση ενδιαφέροντος. Καλώντας το procedure RequestEvaluation από τον ενδιαφερόμενο υπάλληλο ( αν δεν έχει λήξει η ημερομηνία αιτήσεων ), δημιουργείται αυτόματα insert στο table evaluationresult και το Boolean empl\_interested γίνεται true.

Στο table evaluationresult ο κάθε εργαζόμενος μπορείς να στείλει μόνο μία αναφορά για μία δουλειά. Τα triggers evaluationresult(Insert/Update/Delete)Log είναι για να ενημερώσουν το table logs.

Το grade στο table evaluationresult δεν συμπληρώνεται τη στιγμή που παίρνουν τιμές τα διάφορα F1, F2, F3, αλλά ο evaluator καλεί την FinEvaluations. Αν έχει τιμή το grade, δεν μπορεί να αλλαχτεί χάρη στο Trigger UnchangeableGrade το οποίο πρέπει να καλεστεί απ’ το χρήστη.

Το procedure FinishedEvaluations δείχνει τον αριθμό όσων αιτημάτων για δουλειές που έχουν απαντηθεί δείχνοντας και τους υπαλλήλους που την έχουν ζητήσει και την βαθμολογία που τους δόθηκε, αλλά και τον αριθμό όσων δεν έχουν απαντηθεί.

Το τελευταίο procedure ParticularEmployeeRequests είναι μια μορφή αναζήτησης με βάση το ονοματεπώνυμο του υπαλλήλου έτσι ώστε να βρίσκουμε όλες τις δουλειές που έχει ζητήσει για αξιολόγηση, και για όσες υπάρχουν, παρουσιάζεται και η βαθμολογία αλλά και ο evaluator για τη δουλειά αυτή.

Τέλος αξίζει να σημειώσω τη χρήση των Trigger Success και Failure log για όπου για “YES” ή “NO” ενημερώνονται οι αντίστοιχες στήλες, και από τα διάφορα triggers που έχουν τη λέξη log ενημερωνόμαστε για την επιτυχία της ενημέρωσης.

**Παρατήρηση:** Λόγω των @current\_username και @current\_userkind δεν χρειάζεται να γράφουμε κάθε φορα τα στοιχεία του user αλλά και η order\_of\_action είναι αυτόματη.