

Q1. Για να ανακτηθούν τα ονόματα των τμημάτων και το πλήθος των εργαζομένων χρησιμοποιείται το SELECT dept\_name και COUNT(ssn) αντίστοιχα, ενώ επειδή υπάρχει το count μαζί με το dept\_name, χρησιμοποιείται ένα Group by dept\_name το οποίο πάρθηκε στο SELECT και το HAVING για να οριστεί ο μέγιστος μισθος MAX(salary)>42000.

Q2. Χρησιμοποιείται SELECT fname για να ανακτηθούν τα ονόματα των εργαζομένων. Οι πίνακες που θα χρησιμοποιηθούν είναι οι department, employee και συνδέονται με τα foreign και primary keys dnumber, dno τα οποία εξισώνονται στο WHERE. Παράλληλα, στο WHERE χρησιμοποιείται εμφωλευμένο SELECT για να παρθεί το dno στο οποίο θα βρίσκεται το μικρότερο salary που θα είναι MIN(salary). Οπότε χρησιμοποιείται πρώτα εμφωλευμένο SELECT MIN(salary) για να βρεθεί το μικρότερο salary και εν συνεχεία αυτό χρησιμοποιείται για να βρεθεί το dno όπου το salary είναι το παραπάνω.

Q3. SELECT fname ξανά για τα ονόματα των εργαζομένων. Αυτή τη φορά με distinct για να πάρουμε μία εγγραφή ανά όνομα. Στο WHERE, χρησιμοποιείται ξανά ένα εμφωλευμένο SELECT στο οποίο λαμβάνεται ο μέσος μισθός AVG(salary) από το Research Department. Καθώς αναζητάμε τα ονόματα εκείνων που κερδίζουν 5000 περισσότερα από τον μέσο μισθό του εμφωλευμένου SELECT, γράφεται στο WHERE: salary – 5000 < μέσο μισθό του εμφωλευμένου SELECT.

Q4. Για τη δημιουργία όψης χρησιμοποιείται το CREATE VIEW ... AS. Αναζητούμε όνομα τμήματος, διευθυντή και μισθό, οπότε στο SELECT λαμβάνονται dname, fname, salary ενώ τα keys που χρησιμοποιούνται στο where είναι ssn και mgrssn αυτή τη φορά καθώς αναζητούμε τους Managers των τμημάτων.

Q5. Χρησιμοποιώντας το WITH, δημιουργείται ένας temporary πίνακας η οποία δίνει τον αριθμό των εργαζομένων σε κάθε τμήμα. Έπειτα αυτός ο αριθμός χρησιμοποιείται στο SELECT μαζί με dname, fname, lname και τέλος με το COUNT(pname) για να βρεθεί ο αριθμός των projects ανά τμήμα. Έπειτα πραγματοποιούνται τα inner joins για την ένωση των πινάκων. Αντί για το WITH, μπορούσε ο temporary αυτός πίνακας να γίνει view και να χρησιμοποιηθεί το συγκεκριμένο view σαν πίνακας.

Q6. Ξανά, χρησιμοποιώντας το WITH, φτιάχνονται temporary πίνακες. Ο ένας δίνει τον αριθμό των εργαζομένων ανά project, ο επόμενος τις ώρες ανά project ενώ οι άλλοι δύο τον αριθμό males και females ανά προτζεκτ. Για τους δύο τελευταίους πίνακες, χρησιμοποιήθηκε ένωση του works\_on με τον employee όπου essn=ssn και sex='M' και F αντίστοιχα. Για τον πίνακα females συγκεκριμένα, χρησιμοποιήθηκε το FULL OUTER JOIN του works\_on με employee καθώς στο Project Z δεν υπάρχουν γυναίκες και το συγκεκριμένο προτζεκτ δεν θα έδινε εγγραφή στο τελικό αποτέλεσμα. Μετά τη δημιουργία των temporary πινάκων, έγινε SELECT των εγγραφών που ζητείται με ένωση των πινάκων που χρησιμοποιούνται με INNER JOIN.

Q7. Δημιουργείται με WITH ένας πίνακας με τα ssn των 3 εργαζομένων που δούλεψαν τις περισσότερες ώρες. Για να γίνει αυτό, λαμβάνεται το essn και SUM(hours) από το works\_on όπου το hours δεν είναι null (IS NOT null) καθώς υπάρχει εγγραφή με null ώρες, έπειτα γίνεται order by ώρες που δούλεψαν από τις περισσότερες στις λιγότερες για να είναι πάνω πάνω οι 3 περισσότερες και έπειτα με τη χρήση LIMIT 3 κρατούνται μόνο οι 3 πάνω. Στη συνέχεια χρησιμοποιώντας αυτά, ανακτούνται dname, dname, pname, hours με NATURAL JOIN των πινάκων employee, department, works\_on, project και του temporary ο οποίος δημιουργήθηκε. Χρησιμοποιώντας INNER JOIN, χάνονται 2 εγγραφές του Franklin, ενώ με το Natural join παίρνονται όλες έπρεπε.

Q8. Στο SELECT, παίρνουμε name για το όνομα του Instructor και έπειτα χρησιμοποιώντας το CASE WHEN, για κάθε βαθμό χρησιμοποιείται LIKE grade% για να παρθεί ο βαθμός με Regular Expression διότι οι βαθμοί έχουν + και -. Στο τέλος του κάθε case, αναγράφεται το then 1 end για να κλείσει το statement. Οι πίνακες teaches, takes, instructor που χρησιμοποιήθηκαν ενώθηκαν με INNER JOIN. Στο τέλος γίνεται ORDER BY grade DESC και χρησιμοποιούνται όλοι οι βαθμοί για την ταξινόμηση ούτως ώστε να διευκρινιστεί τι θα γίνεται εφόσον για παράδειγμα οι βαθμοί A είναι ίσοι ανάμεσα σε δύο καθηγητές.

Q9. Δημιουργείται ένας temporary πίνακας ο οποίος περιέχει το σύνολο των ωρών στο κάθε room\_number των κτιρίων και ένας ακόμα πίνακας ο οποίος περιέχει τον συνολικό αριθμό των μαθητών σε κάθε αίθουσα. Στον temporary πίνακα ωρών, για τον υπολογισμό των συνολικών ωρών, μετατρέπονται σε λεπτά τα πάντα, γίνονται οι πράξεις και εν τέλει διαιρείται με το 60 για τη μετατροπή σε ώρες και για να μην βγει στην εγγραφή ένας αριθμός με πολλά δεκαδικά, γίνεται round σε ένα δεκαδικό. Έπειτα, ενώνοντας τους παραπάνω πίνακες με τον πίνακα takes, λαμβάνονται οι εγγραφές που ζητούνται. Παρότι το year δεν ζητείται στον πίνακα, προστίθεται ούτως ώστε να γίνεται καλύτερα ο διαχωρισμός των semesters. Επίσης για να μην βγαίνουν έξτρα εγγραφές, πρέπει στα INNER JOINS να προστεθούν πολλά conditions για να αποφευχθούν διπλοεγγραφές.

Q10. Για τη δημιουργία ωρολόγιου προγράμματος, ανακτούνται ο τίτλος μαθήματος, το εξάμηνο, το όνομα καθηγητή, η αίθουσα, το κτίριο, η μέρα και οι ώρες έναρξης και λήξης των μαθημάτων. Όλοι οι πίνακες που χρησιμοποιήθηκαν ενώνονται με INNER JOIN. Χρησιμοποιείται πρώτα το τμήμα καθώς ζητάται το ωρολόγιο πρόγραμμα για κάθε τμήμα. Επίσης χρησιμοποιείται πρώτα το τμήμα στο ORDER BY για τον ίδιο λόγο και έπειτα το start\_hr το οποίο ζητάται στην άσκηση.