

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 $\text{MAX}(\text{salary}) > 42000$.

	dname character varying (20)	count bigint
1	Headquarters	1
2	Administration	3

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 $\text{MIN}(\text{salary})$. Οπότε χρησιμοποιείται πρώτα εμφωλευμένο SELECT $\text{MIN}(\text{salary})$ για να βρεθεί το μικρότερο salary και εν συνεχεία αυτό χρησιμοποιείται για να βρεθεί το dno όπου το salary είναι το παραπάνω.

	fname character varying (20)
1	Franklin
2	John
3	Jennifer
4	Alicia
5	Ramesh
6	Joyce
7	Ahmad

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 $\text{AVG}(\text{salary})$ από το Research Department. Καθώς αναζητάμε τα ονόματα εκείνων που κερδίζουν 5000 περισσότερα από τον μέσο μισθό του εμφωλευμένου SELECT, γράφεται στο WHERE: $\text{salary} - 5000 < \text{μέσο μισθό του εμφωλευμένου SELECT}$.

	fname character varying (20)
1	Jennifer
2	Franklin
3	James

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.

	fname character varying (20)	dname character varying (20)	salary numeric (6)
1	Franklin	Research	40000
2	Jennifer	Administration	43000
3	James	Headquarters	55000

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.

	dname character varying (20)	fname character varying (20)	lname character varying (20)	number_of_empl bigint	number_of_projects bigint
1	Research	Franklin	Wong	4	3
2	Administration	Jennifer	Wallace	3	2
3	Headquarters	James	Bong	1	1

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.

	pname character varying (20)	dname character varying (20)	employees_per_project bigint	count_males bigint	count_females bigint	hours_per_project numeric
1	Reorganization	Headquarters	3	2	1	
2	ProductY	Research	3	2	1	
3	ProductZ	Research	2	2	0	
4	ProductX	Research	2	1	1	
5	Computerization	Administration	3	2	1	
6	Newbenefits	Administration	3	1	2	

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.

	fname character varying (20)	dname character varying (20)	pname character varying (20)	hours numeric (2)
1	Ahmad	Administration	Newbenefits	5
2	John	Research	ProductY	8
3	Franklin	Research	Computerization	10
4	Franklin	Research	Reorganization	10
5	Franklin	Research	ProductY	10
6	Franklin	Research	ProductZ	10
7	John	Research	ProductX	33
8	Ahmad	Administration	Computerization	35

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 RE 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 βαθμοι για την ταξινόμηση ούτως ώστε να διευκρινιστεί τι θα γίνεται εφόσον για παράδειγμα οι βαθμοί A είναι ίσοι ανάμεσα σε δύο καθηγητές.

	name character varying (20)	a bigint	b bigint	c bigint	d bigint	e bigint	f bigint
1	Srinivasan	6	1	2	0	0	1
2	Brandt	2	1	0	0	0	0
3	Crick	1	0	0	0	0	0
4	Mozart	1	0	0	0	0	0
5	Katz	0	2	0	0	0	0
6	El Said	0	1	0	0	0	0
7	Einstein	0	1	0	0	0	0
8	Kim	0	0	1	0	0	0
9	Wu	0	0	1	0	0	0

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.

	building character varying (15)	room_number character varying (7)	semester character varying (6)	year numeric (4)	tot_hours numeric	tot_student_no bigint
1	Packard	101	Fall	2017	15.0	
2	Packard	101	Spring	2018	2.5	
3	Painter	514	Spring	2018	2.5	
4	Painter	514	Summer	2017	2.5	
5	Painter	514	Summer	2018	2.5	
6	Taylor	3128	Fall	2017	5.0	
7	Taylor	3128	Spring	2017	2.5	
8	Taylor	3128	Spring	2018	2.5	
9	Taylor	3128	Spring	2017	5.0	
10	Watson	100	Fall	2017	2.5	
11	Watson	100	Spring	2018	2.5	
12	Watson	120	Spring	2018	5.0	

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.

	dept_name character varying (20)	title character varying (50)	semester character varying (6)	year numeric (4)	name character varying (20)
1	Biology	Genetics	Summer	2018	Crick
2	Biology	Genetics	Summer	2018	Crick
3	Biology	Genetics	Summer	2018	Crick
4	Biology	Intro. to Biology	Summer	2017	Crick
5	Biology	Intro. to Biology	Summer	2017	Crick
6	Biology	Intro. to Biology	Summer	2017	Crick
7	Comp. Sci.	Database System Concepts	Fall	2017	Srinivasan
8	Comp. Sci.	Database System Concepts	Fall	2017	Srinivasan