

Επαναληπτικές Ασκήσεις

HY360

Σχεσιακό Μοντέλο

- Θεωρείστε το ακόλουθο σχεσιακό σχήμα (υπογραμμίζονται τα κλειδιά):

Student (StudentID, Name, Department, Year¹)

Course (CourseID, Title, Department², InstructorID)

Enrollment (StudentID, CourseID, Grade³)

Instructor (InstructorID, Name, Department⁴)

¹: παίρνει τιμές 1,2,3,4,..

²: το Τμήμα που προσφέρει το μάθημα

³: παίρνει τιμές 0,1,...,10

⁴: το Τμήμα στο οποίο ανήκει ο διδάσκων

Εκφράστε τις ακόλουθες ερωτήσεις στη γλώσσα ή τις γλώσσες που προσδιορίζονται

Σχεσιακό Μοντέλο

Student (StudentID, Name, Department, Year)

Course (CourseID, Title, Department, InstructorID)

Enrollment (StudentID, CourseID, Grade)

Q1: Βρείτε τα ονόματα των φοιτητών που εγγράφονται μόνο σε μαθήματα που προσφέρονται από το CSD (σε Σχεσιακή Άλγεβρα και SQL)

Σχεσιακή Άλγεβρα

$\pi_{\text{Name}} (\text{Student} \text{ JOIN } (\pi_{\text{StudentID}} (\text{Student}) - \pi_{\text{StudentID}} (\sigma_{\text{Department} \neq \text{CSD}} (\text{Course JOIN Enrollment}))))$

SQL: select Name from Student where StudentID not in
(select StudentID from Enrollment, Course where Course.CourseID = Enrollment.CourseID
and Department $\neq$ CSD)

Σχεσιακό Μοντέλο

Enrollment (StudentID, CourseID, Grade)

Q2: Βρείτε τα IDs των φοιτητών που πήραν τον υψηλότερο βαθμό στο μάθημα HY360 (σε SQL, Σχεσιακό Λογισμό Πεδίων και Σχεσιακή Άλγεβρα)

SQL: select StudentID from Enrollment where Grade > all
(select Grade from Enrollment where CourseID = HY360)

Σχεσιακός Λογισμός Πεδίων

$\{s \mid \exists g (\text{Enrollment}(s, \text{HY360}, g) \wedge \forall s1, g1 (\text{Enrollment}(s1, \text{HY360}, g1) \wedge (s \neq s1) \rightarrow (g > g1))) \}$

Σχεσιακή Άλγεβρα

T1: $\sigma_{\text{CourseID} = \text{HY360}}(\text{Enrollment})$; T2:=T1; T3:= $\pi_{\text{StudentID}} (\sigma_{T1.\text{Grade} < T2.\text{Grade}} (T1 \times T2))$;

T4: $\pi_{\text{StudentID}} T1 - T3$

Σχεσιακό Μοντέλο

Enrollment (StudentID, CourseID, Grade)

Q3: Βρείτε τα IDs των μαθημάτων και τον αριθμό των εγγεγραμμένων σε αυτά φοιτητών (σε SQL, Σχεσιακό Λογισμό Πεδίων και Σχεσιακή Άλγεβρα)

```
SQL:  SELECT CourseID, COUNT(StudentID)
      FROM Enrollment
      GROUP BY CourseID;
```

Σχεσιακός Λογισμός Πεδίων: δε μπορεί να εκφραστεί

Σχεσιακή Άλγεβρα: δε μπορεί να εκφραστεί

Σχεσιακό Μοντέλο

Student (StudentID, Name, Department, Year)

Course (CourseID, Title, Department, InstructorID)

Enrollment (StudentID, CourseID, Grade)

Q4: Βρείτε τα ονόματα των φοιτητών που εγγράφονται σε όλα τα μαθήματα που προσφέρονται από το CSD (σε Σχεσιακή Άλγεβρα και SQL)

Σχεσιακή Άλγεβρα

$T1 := \pi_{CourseID} (\sigma_{Department = CSD} (Course)); T2 := \pi_{StudentID, CourseID} (Enrollment) \div T1;$

$T3 := \pi_{Name} (Student \text{ JOIN } T2)$

SQL: select Name from Student where not exists

(select CourseID from Course where Department = CSD and not exists

(select StudentID, CourseID from Enrollment where

Course.CourseID = CourseID and Student.StudentID = StudentID))

Σχεσιακό Μοντέλο

Course (CourseID, Title, Department, InstructorID)

Enrollment (StudentID, CourseID, Grade)

Instructor (InstructorID, Name, Department)

Q5: Βρείτε τα ονόματα των διδασκόντων που διδάσκουν τουλάχιστον ένα μάθημα με περισσότερους από 10 εγγεγραμμένους φοιτητές (σε SQL)

```
SQL: select Name from Instructor I, Enrollment E, Course C
      where I.InstructorID = C.InstructorID and C.CourseID=E.CourseID
      group by InstructorID, CourseID
      having count(StudentID) > 10
```

Σχεσιακό Μοντέλο

Student (StudentID, Name, Department, Year) Enrollment (StudentID, CourseID, Grade)

Q6: Βρείτε το μέσο βαθμό ανά μάθημα για τους φοιτητές του τμήματος CSD (σε SQL)

SQL: select CourseID, Avg(Grade) from Enrollment E, Student S
 where E.StudentID = S.StudentID and S.Department = CSD
 group by E.CourseID

Q7: Βρείτε ζεύγη φοιτητών που ανήκουν σε διαφορετικά τμήματα και παίρνουν το ίδιο μάθημα (σε Σχεσιακή Άλγεβρα και Σχεσιακό Λογισμό Πεδίων)

S1:=Student JOIN Enrollment; S2:= Student JOIN Enrollment

$\pi_{S1.StudentID, S2.StudentID} (\sigma_{(S1.Department \neq S2.Department) \wedge (S1.StudentID < S2.StudentID) \wedge (S1.CourseID = S2.CourseID)} (S1 \times S2))$

$\{(s1, s2) \mid \exists n1, d1, y1, n2, d2, y2, c, g1, g2 \text{ Student}(s1, n1, d1, y1) \wedge \text{Student}(s2, n2, d2, y2) \wedge (s1 < s2) \wedge \text{Enrollment}(s1, c, g1) \wedge \text{Enrollment}(s2, c, g2)\}$