

CS 386/586 Exercise 3

Work through the exercises

Undergraduate students, please turn in your paper (with papers from all other exercises) at the end of the term.

Teacher table:			
Number	Name	Office	E-mail
1	Lois	O12	imd@whatever.edu
2	James	O14	jterwill@whatever.edu
3	Sally	O55	sally@whatever.edu

Course table:		
Number	Name	Description
386	Intro to Databases	Introduction to relational database management systems
161	Intro to Computer Science 1	Introduction to problem-solving, algorithm and program design

In the following instance of the Class-offering table, only two of the class offerings for the Fall 2007 Quarter are listed. We assume that CRN is the key for this table. We also assume that each class offering has only one teacher, just to keep things simple.

Class-offering table:			
CRN	Course	Section	Teacher
701	386	1	1
702	161	1	3

Student table:	
Id	Name
101	Joe
102	Mary
103	Bill

Scores table			
CRN	Id	Assign	Grade
701	101	1	95
701	102	1	90
701	101	2	100
701	102	2	95
701	101	3	95

Show the query answer for each of the following queries. (Note: the UNION, INTERSECT, and EXCEPT clauses requires union-compatible input tables. If the input tables are not union-compatible, then the query is not well-formed and does not have a query answer. The same is true for the \cap , \cup , and $-$ operators in relational algebra.)

Show the query answer for each of the following queries.

Explain what the query computes in English.

Write an equivalent query in relational algebra.

1. (SELECT * FROM Teacher)
UNION
(SELECT * FROM Student);

2. (SELECT Name FROM Teacher)
UNION
(SELECT Name FROM Student);

3. (SELECT Grade FROM Scores WHERE Id = 101)
INTERSECT
(SELECT Grade FROM Scores WHERE Id = 102);

4. (SELECT Assign FROM Scores WHERE Id = 101)
EXCEPT
(SELECT Assign FROM Scores WHERE Id = 102);

For each of the following queries, show the query answer and explain what it computes in English. Use the data shown on the first page of the exercise, above.

5.

```
SELECT    Id, Name, Grade
FROM      Student FULL OUTER JOIN Scores USING (Id);
```

6.

```
SELECT    Id, Name, Grade
FROM      Student RIGHT OUTER JOIN Scores USING (Id);
```

7.

```
SELECT    DISTINCT Student.Id, Name, Assign, Grade
FROM      Student, Scores
WHERE     Student.Id = Scores.Id and
         Scores.Grade >= ALL (SELECT Grade FROM Scores);
```

8.

```
SELECT    DISTINCT Student.Id, Name, Assign, Grade
FROM      Student, Scores
WHERE     Student.Id = Scores.Id and
         Scores.Grade = (SELECT MAX(Grade) FROM Scores);
```

9.

```
SELECT    DISTINCT Student.Id, Name, S1.Assign, S1.Grade
FROM      Student, Scores S1
WHERE     Student.Id = S1.Id and
         (SELECT MAX(Grade) FROM Scores S2
          WHERE S1.Id = S2.Id ) = S1.Grade;
```

For questions 7 through 9, which of the subqueries, if any, are correlated?

Consider the following table. It is the same as the table shown above except that we have added one new row, listed as the last row shown here.

Scores table			
<u>CRN</u>	<u>Id</u>	Assign	Grade
701	101	1	95
701	102	1	90
701	101	2	100
701	102	2	95
701	101	3	95
701	102	3	<i>null</i>

For each of the following queries, indicate whether this new, 6th row in the Scores table would satisfy the WHERE clause.

10. SELECT *
FROM Scores
WHERE Grade > 90;

11. SELECT *
FROM Scores
WHERE Grade = 95 OR Grade < 95 OR Grade > 95;

12. SELECT *
FROM Scores
WHERE Grade IS NULL;

13. SELECT *
FROM Scores
WHERE Grade IS NOT NULL;