2.

SQL:

* SELECT DISTINCT s.snum, s.name

FROM student AS s

JOIN enrolled AS e ON e.snum = s.snum

WHERE s.major = 'CS' AND e.class\_name = 'IT3292E Database 20221';

Relational Algebra:

* πs .*snum*, s .*name*  
   σs .*major*= "CS"*AND*e .*class*\_name = "IT3292E*Database*20221"  
    (ρs*student* ⋈e .*snum*= s .*snum*  
     ρe*enrolled*)

3.

SQL:

* SELECT DISTINCT s.snum, s.name

FROM student AS s

JOIN enrolled AS e ON e.snum = s.snum

JOIN class AS c ON c.name = e.class\_name

JOIN faculty\_member AS fm ON fm.fid = c.fid

WHERE c.room = 'D9-205'

OR (fm.name = 'Prof. Nguyen' AND fm.dept = 'Computer Science');

Relational Algebra:

* πs .*snum*, s .*name*  
   σc .*room*= "D9-205"*OR*(fm .*name*= "Prof.*Nguyen*"*AND**fm*.*dept*= "Computer*Science*")  
    (ρs*student* ⋈e .*snum*= s .*snum*  
     ρe*enrolled* ⋈c .*name*= e .*class*\_name  
      ρc*class* ⋈*fm*.*fid*= c .*fid*  
       ρ*fm**faculty*\_member)

4.

SQL:

* SELECT DISTINCT s1.snum, s1.name, s2.snum, s2.name

FROM enrolled AS e1

JOIN enrolled AS e2 ON e1.class\_name = e2.class\_name

JOIN student AS s1 ON s1.snum = e1.snum

JOIN student AS s2 ON s2.snum = e2.snum

WHERE e1.snum < e2.snum;

Relational Algebra:

* π*s1*.*snum*,*s1*.*name*,*s2*.*snum*,*s2*.*name*  
   σ*e1*.*snum*<*e2*.*snum*  
    (ρ*e1**enrolled* ⋈*e1*.*class*\_name =*e2*.*class*\_name  
     ρ*e2**enrolled* ⋈*s1*.*snum*=*e1*.*snum*  
      ρ*s1**student* ⋈*s2*.*snum*=*e2*.*snum*  
       ρ*s2**student*)

5.

SQL:

* SELECT DISTINCT s.snum, s.name, c1.name, c2.name

FROM class AS c1

JOIN class AS c2

ON c1.week\_day = c2.week\_day

AND c2.start\_time <= c1.start\_time

AND c1.start\_time <= c2.end\_time

JOIN enrolled AS e1 ON e1.class\_name = c1.name

JOIN enrolled AS e2 ON e2.class\_name = c2.name

JOIN student AS s ON s.snum = e1.snum

WHERE c1.name <> c2.name

AND e1.snum = e2.snum;

Relational Algebra:

* πs .*snum*, s .*name*,*c1*.*name*,*c2*.*name*  
   σ*c1*.*name*<>*c2*.*name**AND**e1*.*snum*=*e2*.*snum*  
    (ρ*c1**class* ⋈*c1*.*week*\_day =*c2*.*week*\_day*AND**c2*.*start*\_time **≤***c1*.*start*\_time*AND**c1*.*start*\_time **≤***c2*.*end*\_time  
     ρ*c2**class* ⋈*e1*.*class*\_name =*c1*.*name*  
      ρ*e1**enrolled* ⋈*e2*.*class*\_name =*c2*.*name*  
       ρ*e2**enrolled* ⋈s .*snum*=*e1*.*snum*  
        ρs*student*)

6.

SQL:

* SELECT DISTINCT fm.fid, fm.name

FROM faculty\_member AS fm

JOIN class AS c ON c.fid = fm.fid

WHERE c.semester = 20221

GROUP BY fm.fid

HAVING COUNT(DISTINCT c.week\_day) = 6;

Relational Algebra:

* π*fm*.*fid*,*fm*.*name*  
   σ*COUNT*(DISTINCT*week*\_day) = 6  
    γ*fid*,*COUNT*(DISTINCT*week*\_day)  
     σc .*semester*= 20221  
      (ρ*fm**faculty*\_member ⋈c .*fid*=*fm*.*fid*  
       ρc*class*)

7.

SQL:

* SELECT DISTINCT s.level, AVG(s.age) AS average\_age

FROM student AS s

GROUP BY s.level;

Relational Algebra:

* πs .*level*,*AVG*(age) →*average*\_age  
   γ*level*,*AVG*(age)  
    ρs*student*

8.

SQL:

* SELECT DISTINCT fm.fid, fm.name, COUNT(c.name) AS classes

FROM faculty\_member AS fm

JOIN class AS c ON c.fid = fm.fid

GROUP BY fm.fid, fm.name, c.semester

HAVING COUNT(c.name) > 4;

Relational Algebra:

* π*fm*.*fid*,*fm*.*name*,*COUNT*(name) →*classes*  
   σ*COUNT*(name) > 4  
    γ*fid*,*name*,*semester*,*COUNT*(name)  
     (ρ*fm**faculty*\_member ⋈c .*fid*=*fm*.*fid*  
      ρc*class*)

9.

SQL:

* WITH student\_classes AS (

SELECT s.snum, s.name, COUNT(e.class\_name) AS classes

FROM student AS s

LEFT JOIN enrolled AS e ON e.snum = s.snum

GROUP BY s.snum, s.name

)

SELECT DISTINCT sc.snum, sc.name

FROM student\_classes AS sc

WHERE sc.classes = (SELECT MIN(classes) FROM student\_classes);

Relational Algebra: …

10.

SQL:

* SELECT DISTINCT s.snum, s.name

FROM student AS s

EXCEPT

SELECT DISTINCT s.snum, s.name

FROM student AS s

JOIN enrolled AS e ON e.snum = s.snum

JOIN class AS c ON c.name = e.class\_name

JOIN faculty\_member AS fm ON fm.fid = c.fid

WHERE fm.dept = 'Computer Science';

Relational Algebra:

* πs .*snum*, s .*name*  
   ρs*student* ∖  
    πs .*snum*, s .*name*  
     σ*fm*.*dept*= "Computer*Science*"  
      (ρs*student* ⋈e .*snum*= s .*snum*  
       ρe*enrolled* ⋈c .*name*= e .*class*\_name  
        ρc*class* ⋈*fm*.*fid*= c .*fid*  
         ρ*fm**faculty*\_member)