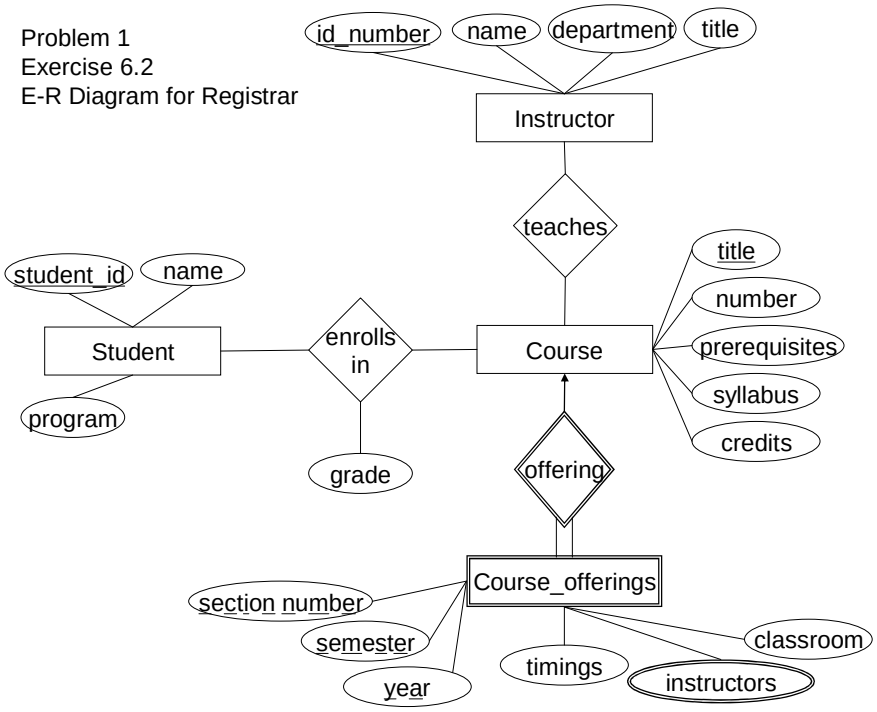
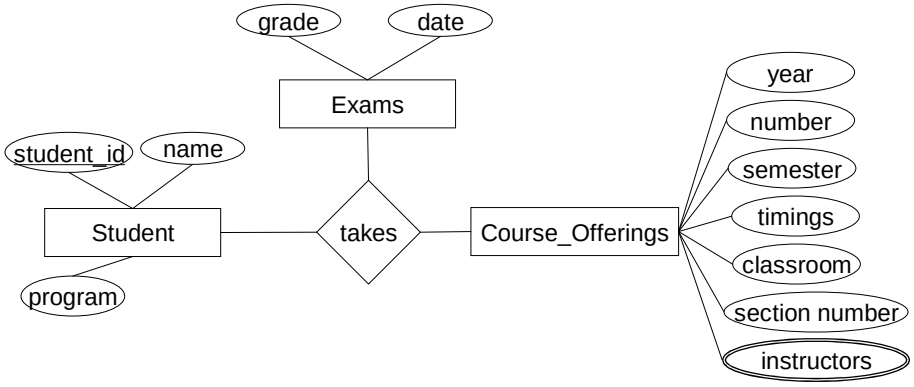
Patrick Trinkle
CMSC461 SU06
Homework #1

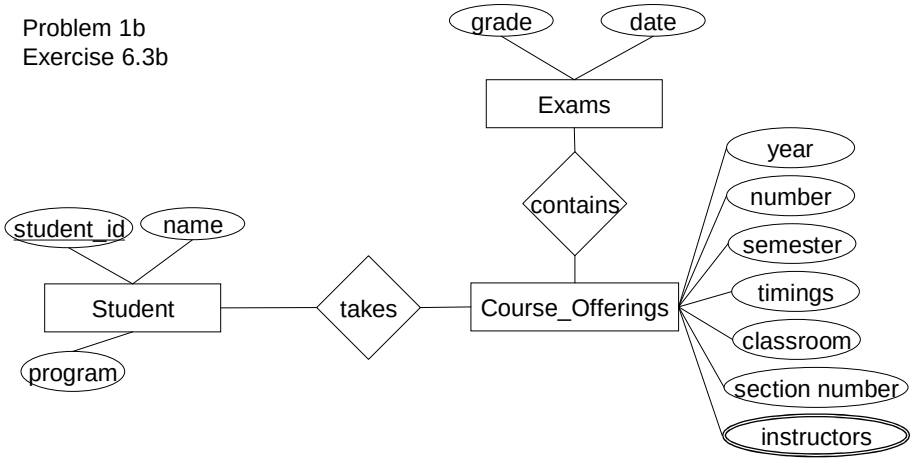
**Problem 1. (20 points)**
Exercises 6.2 and 6.3 on page 256 of the text.



Problem 1b
Exercise 6.3a




Problem 1b
Exercise 6.3b




**Problem 2. (20 points)**
Exercise 6.16 on page 258 of the text.

a)
customer = (customer_id, name, primary_telephone, monthly_rate)
car = (VIN, make, model, year)
accident = (VIN, claim_number, cost, type, date)

insures = (customer_id, VIN)
b)
student = (student_id, name, program)
instructor = (id_number, name, department, title)
course = (title, number, prerequisites, syllabus, credits)
course_offering = (title, section_number, semester, year, timings, classroom)
instructors = (title, instructor) *
enrolls-in = (student_id, title, grade)
teaches = (id_number, title)
offering = (title, section_number, semester, year)
*title must reference valid instructor tuple
**Also, does course_offering still need an instructors attribute?**

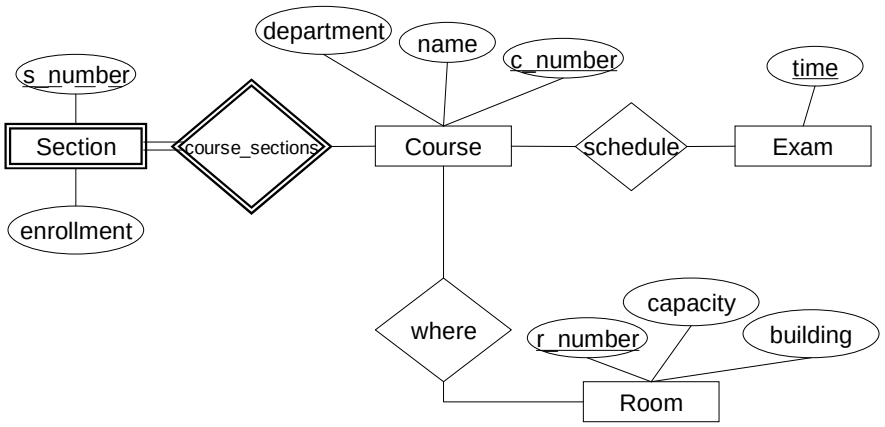
**Problem 3. (10 points)**
Exercise 6.19 on page 259 of the text.

**6.19** We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then do we have weak entity sets?

If we added the appropriate attributes to the weak entity we would be giving it the primary key attributes of its owner entity. This would make to strong entities with essentially the same primary key, but also two strong entities which basically belong to one another, wherein you cannot have one of them (the original weak entity) without the strong entity in a total relationship. In this case the weak entity is essentially redundant and inappropriately independent and therefore should be modeled differently.

**Problem 4. (10 points)**
Exercise 6.6 on page 256 of the text.

Problem 4a
Exercise 6.6a




b)
Scheduling courses to rooms based on the needs of a specific section of a course could more easily be conducted with the additional entities. Queries could be constructed such that large sections could have larger rooms.

**Problem 5. (60 points)**
Exercises 2.1, 2.5, 2.8, and 2.11 on pages 71-73 of the text.

$\Pi \neg \sigma \cup \cap \wedge \vee \leftarrow$

2.1
a)
b) $\Pi_{\text{person\_name}} (\sigma_{\text{company\_name} \neg \text{"First Bank Corporation"}} (\text{works}))$
c) maxsalary $\leftarrow$ **Gmax** $(\Pi_{\text{salary}} (\sigma_{\text{company\_name} = \text{"Small Bank Corporation"}} (\text{works})))$
result $\leftarrow \Pi_{\text{person\_name}} (\sigma_{\text{company\_name} \neg \text{"Small Bank Corporation"}} (\text{works}) \wedge \text{salary} > \text{maxsalary})$

2.5
a) $\Pi_{\text{person\_name}} (\sigma_{\text{company\_name} = \text{"First Bank Corporation"}} (\text{works}))$
b) $\Pi_{\text{person\_name, city}} (\sigma_{\text{company\_name} = \text{"First Bank Corporation"}} (\text{employee} \mid X \mid \text{works}))$
c) $\Pi_{\text{person\_name, street, city}} (\sigma_{\text{company\_name} = \text{"First Bank Corporation"} \wedge \text{salary} > 10{,}000} (\text{employee} \mid X \mid \text{works}))$
d) $\Pi_{\text{person\_name}} (\text{employee} \mid X \mid \text{company})$
e)

2.8

2.11

**Problem 6. (80 points)**
Exercise 3.2, 3.3, 3.6, 3.9, 3.11, and 3.21 on pages 117-120 of the text.

3.2
a) select employee_name, city from employee, works where company_name = 'First Bank Corporation'
b) select employee_name, city, street from employee, works where company_name = 'First Bank Corporation' and salary > 10000
c) select employee_name from works where company_name <> 'First Bank Corporation'
d) select employee_name from works where salary > max(select salary from works where company_name = 'Small Bank Corporation')
e)
f)
g) select company_name from works where company_name <> 'First Bank Corporation' and avg (salary) > avg (select salary from works where company_name = 'First Bank Corporation')

3.3
a)
update employee
set city = 'Newtown'
where employee_name = 'Jones'
b)
update works
set salary = case
when salary * 1.10 > 100000 then salary * 1.03
else salary * 1.10
end
where company_name = 'First Bank Corporation' and (select employee_name, manager_name from works, manages where manager_name = works.employee_name)

3.6
The query selects values of p.a1 where either the value of p.a1 is equal to r1.a1 or in the case where p.a1 is equal to r2.a1 or the case where it is possibly equal to both. If either is empty and the other matches it will still return p.a1.

3.9
a) select employee_name from works where company_name = 'First Bank Corporation'
b) select employee_name from employee, works, company where company.company_name = works.company_name and employee.employee_name = works.employee_name and company.city = employee.city
c)

d) select employee_name from works as Person where Person.salary > avg (select salary from works where company_name = Person.company_name)
e)

3.11
a) select A from r
b) select * from r where B = 17
c) select * from r,s
d) select A, F from r,s where C = D

3.21
a) select name from (books natural inner join loan, employee natural inner join loan) where publisher = ‘McGraw-Hill’
b)
c)