1) (22 points) Consider the following relational scheme for storing information associated with courses and faculty members at some university, which is not necessarily Florida Tech (Note that some of the rules may have changed from what you have seen before).

University_Info= (FACULTY#, DEPARTMENT, COURSE#, SECTION#, BOOK)

FACULTY#  A faculty identification number.
DEPARTMENT A department name.
COURSE#  A course identification number.
SECTION#  The number of a course section, e.g., 01, 02, 03, etc.

Suppose that the following rules apply in the university.

1. Department names are math, computer science, biology, etc.
2. Course identification numbers are unique.
3. Every faculty member is assigned to one or more departments, i.e., joint appointments are allowed.
4. Each course that is taught in a given semester, has one or more sections.
5. Every faculty member teaches one or more course sections.
6. Different faculty members can teach different sections of the same course.
7. No more than one faculty member is assigned the same section of the same course.
8. Every faculty member selects one or more books for each course that they teach.
9. Different faculty members teaching the same course can use the same or different books for their sections.
10. A faculty member may use different books for each section of a specific course that they teach.
11. No other rules apply.

Based on the above rules, circle each of the following functional dependencies that hold.

FACULTY# => DEPARTMENT  SECTION# => COURSE#
BOOK => COURSE#,SECTION#  COURSE#,SECTION# => FACULTY#
FACULTY#,COURSE#,SECTION# => BOOK  FACULTY#,BOOK => COURSE#
COURSE#,SECTION# => BOOK  SECTION#,BOOK => COURSE#
FACULTY#,COURSE# => SECTION#  COURSE#,BOOK => BOOK
DEPARTMENT,COURSE#,SECTION#,BOOK => FACULTY#
2) (20 points) Several axioms/rules for functional dependencies are shown below. Some of these were discussed in class, but others were not. Circle those that are true. In all cases, you may assume that α, β, γ, δ and ε are sets of attributes. Note that no proof is required.

(a) If α → β and α → γ then α → βγ
(b) If αβ → γ then αγ → β and βγ → α
(c) If α → β then αγ → β
(d) If α → β and β → γ then α → γ
(e) If α ⊆ β then α → β
(f) If α → β and γ → δ then αγ → βδ
(g) Let ε = γ∩δ. If α → γ and β → δ then αβ → ε, α → ε, and β → ε
(h) Let ε = γ∩δ. If α → γ and β → δ then ε → αβ
(i) If α → β then γα → β
(j) If αβ → γ then α → γ and β → γ

3) (4 points) The pseudo-transitivity rule states that if α → β and γβ → δ, then αγ → δ. Prove the pseudo-transitivity rule using Armstrong’s axioms.
4) Consider the algorithm shown below for decomposing a given relation scheme.

(1) let $F_c$ be a canonical cover for $F$;
(2) $i := 0$;
(3) for each functional dependency $\alpha \rightarrow \beta$ in $F_c$ do
(4) if (none of the schemas $R_j$, $j=1,2,\ldots,i$ contains the attributes in $\alpha$ and $\beta$) then begin
(5) $i := i + 1$;
(6) $R_i := (\alpha, \beta)$;
(7) end;
(8) if (none of the schemas $R_j$, $j=1,2,\ldots,i$ contains a candidate key for $R$) then begin
(9) $i := i + 1$;
(10) $R_i :=$ any candidate key for $R$;
(11) end;

(a) (4 points) Are the relational schemes resulting from the above algorithm guaranteed to be in BCNF (yes or no)?

(b) (4 points) Are the relational schemes resulting from the above algorithm guaranteed to be in 3NF (yes or no)?

(c) (4 points) Is the resulting decomposition guaranteed to have a lossless join (yes or no)?

(d) (4 points) Is the resulting decomposition guaranteed to preserve dependencies (yes or no)?

(e) (4 points) If your answer to part (d) is “yes” then explain why (i.e., which parts/lines of the algorithm are responsible for guaranteeing this, and how do they do it), and if your answer to part (d) is “no” then explain why not.

(f) (4 points) What is the purpose of the section of the algorithm given on lines 8-11? More specifically, why does the algorithm add another relation containing a candidate key if none is contained in any of the other relational schemes?
5) (30 points) Consider the following relational schemes. Note that for this set of schemes, an employee can work for multiple companies, for different managers at those companies, and at differently hourly-rates.

employee (employee-name, street, city)
company (company-name, city)
works (employee-name, company-name, hourly-rate)
manages (employee-name, company-name, manager-name)

(a) Give a relational algebra expression for a list of the names of those companies that are located in Tampa, have an employee named “Smith,” but do not have any employees who live in Atlanta.

(b) Give a tuple calculus expression for a list of the names and cities of residence for those employees who are managed by Jones on a job at IBM for more than $10 per hour.

Give an SQL statement for the following.

c) An employee is said to be poorly-compensated if their average hourly-rate (among all the companies they work for) is less than $15. A company is said to be stingy if the highest hourly-rate paid to any of its employees is $50. Give an SQL statement for a sorted list of employee names along with their average hourly-rate, for those employees who are both poorly-compensated and work for at least one stingy company. Note that the name of the column containing the average should be labeled "avg-hourly-rate."

<table>
<thead>
<tr>
<th>employee-name</th>
<th>avg-hourly-rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carson</td>
<td>10.50</td>
</tr>
<tr>
<td>Jones</td>
<td>14.75</td>
</tr>
<tr>
<td>Morris</td>
<td>12.62</td>
</tr>
</tbody>
</table>