Answer Keys for Assignment 2 (based on the version provided by Paea LePendu)  
Database Processing CIS 451/551  

1.  
   a) update works set salary = salary * 1.04  
      where name in (select distinct(manager_name) from manages)  
      and salary >= 10,000  
      
      update works set salary = salary * 1.05  
      where name in (select distinct(manager_name) from manages)  
      and salary < 10,000  
      
      update works set salary = salary * 1.02  
      where name NOT in  
           (select distinct(manager_name) from manages  
           where manager_name is not null)  
      *EXTRA CREDIT if add this line  
   
   b) create view startup_managers as  
      select employee_name  
      from works, manages  
      where works.employee_name = manages.manager_name  
      and works.company_name = ‘Startup Corp’  
      and salary >= 50,000  
      
      delete from employee  
      where employee_name in startup_managers  
      
      delete from works  
      where employee_name in startup_managers  
      
      delete from manages  
      where employee_name in startup_managers  
      
      EXTRA CREDIT: clean up foreign key dependencies in manages  
      (students can cascade delete, or otherwise deal with it in their own way)  

2. create view all_workers(name) as  
   select name from salaried_worker  
   union  
   select name from hourly_worker  
   
   Updates shouldn’t be allowed through this view because of the union. Two tables may  
   share same names. Updates on this view are problematic – which underlying table should you  
   update the name into? Also, there will necessarily be null values no matter which table you  
   chose, which we like to avoid.
3. a) 
\[
\text{select E.name} \\
\text{from A.employee as E} \\
\text{union} \\
\text{select W.name} \\
\text{from B.all_workers as W} \\
\] * this assumes the view in #2 otherwise elaborate \\
b) 
\[
\text{select E.name} \\
\text{from A.employee as E} \\
\text{where E.name not in} \\
\text{(select W.name from B.all_workers as W} \\
\text{where W.name is not null)} \\
\] * EXTRA CREDIT if add this line \\

4. Some comparisons of different approaches to do multi-valued foreign key dependencies. Any of 
these or some different but meaningful comparison is fine for grading.

<table>
<thead>
<tr>
<th>Solution 1</th>
<th>Solution 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>· can short-circuit evaluate (stop once true)</td>
<td>· implementation is transparent</td>
</tr>
<tr>
<td>· fast if the evaluation favors the “better” relation first</td>
<td>· view can be materialized or on-the-fly</td>
</tr>
<tr>
<td>· no extra space needed</td>
<td>· materialized takes space</td>
</tr>
<tr>
<td>· on-the-fly evaluation needed</td>
<td>· indexed view can be speedy</td>
</tr>
<tr>
<td>· can be time consuming if we have to sequentially search both relations for key</td>
<td>· on-the-fly view does not necessarily allow short-circuit evaluation</td>
</tr>
<tr>
<td></td>
<td>· modular, clean, clear, reusable</td>
</tr>
</tbody>
</table>

5. a) company(fname, minit, lname, ssn, bdate, address, sex, salary, superssn, dno) 

b) 0 (zero) records 

c) * Asterisk means they are a manager. 

  John Smith  
  * Franklin Wong  
  Alicia Zelaya  
  * Jennifer Wallace  
  Ramesh Narayan  
  Joyce English  
  Ahmad Jabbar  
  * James Borg  

d) John Smith  
  Alicia Zelaya  
  Ramesh Narayan  
  Joyce English  
  Ahmad Jabbar  

e) James Borg is the problem, his supperssn is null. add: where superssn is not null  
f) unknown (false is ½ points)
6. select r1.x, r2.y, r2.z, 3
   from R as r1, R as r2, R as r3
   where r1.x = r3.z AND
   r1.y = r2.x AND
   r1.z = r2.y AND
   r1.y = r3.y AND
   r2.z = r3.z*1.05 AND
   r3.x = 3