Information and Database Management Systems I (CIS 4301)
(Spring 2017)
Instructor: Dr. Markus Schneider
TA: Yang Peng
Homework 5

Name: 
UFID: 
Email Address: 

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

_______________________________________________
Signature

For scoring use only:

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Maximum</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise 1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Exercise 2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Exercise 3</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Exercise 4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 1 (Functional Dependencies and Normal Forms) [20 points]

1. Consider the relation $R (A, B, C, D, E)$ with FDs: $AB \rightarrow C$, $C \rightarrow B$, $A \rightarrow D$.
   Decompose the $R$ into BCNF relations. Please specify the decomposed relation schemas as well as projected FDs after decomposition. Show your steps. [6 points]

2. Let $R (A, B, C, D, E, F)$ be a relation with FDs
   $F: AB \rightarrow C$, $BC \rightarrow AD$, $D \rightarrow E$, $CF \rightarrow B$.
   Do the two FDs $f_1 : AB \rightarrow D$ and $f_2 : AB \rightarrow F$ hold in $R$? [6 points]
   If yes, please prove you answer in two different ways:
   a. use Armstrong's axioms;
   b. use close test.

   If not, please give a counter example relation with two tuples that satisfy $F$ but does not satisfy the FD

3. Consider the relation Part ($part, manufacturer, seller, price$) with the following functional dependencies:

   part $\rightarrow$ manufacturer;
   part, seller $\rightarrow$ price.

   (a) Keeping in mind the FDs, make an instance of this relation that has redundant information. [4 points]

   (b) If we apply the decomposition step from BCNF decomposition, what attributes would each of the new relations have? And project the FDs onto each of the new relations. [4 points]
Exercise 2 (Functional Dependencies and Normal Forms) [30 points]

1. Consider the relation S (A, B, C, D, E). Let the following functional dependencies be defined over the relation S:

   A → BC, CD → E, B → D, E → A

(a) Identify whether this relationship is in 3NF and BCNF? Explain why. [4 points]
(b) Provide the projection of FDs for the subset of attributes {ABE}. [2 points]
(c) Give a BCNF decomposition of S that is lossless. Explain with steps. [4 points]
(d) Is your BCNF decomposition dependency preserving? Explain Why. [2 points]
(e) Give a 3NF decomposition of S that is lossless and dependency preserving. [2 points]

2. Consider the following functional dependencies over the attribute set ABCDEFGH:
A → E, BE → D, AD → BE, BDH → E, AC → E, F → A, E → B, D → H, BG → F, CD → A

(a) Find a minimal cover for this set of functional dependencies. [4 points]
(b) Decompose the relation ABCDEFGH into a lossless 3NF schema. Explain with steps. [6 points]
(c) Check whether your answer to (b) is in BCNF. If not, decompose it into a lossless BCNF schema. [3 points]
(d) Find a candidate key in relation S. [3 points]
Exercise 3 (Triggers) [25 points]

Considering the following tables in a database:

CREATE TABLE EMP(
    EMPNO NUMBER(4) NOT NULL,
    ENAME VARCHAR2(10),
    JOB VARCHAR2(9),
    MGR NUMBER(4),
    HIREDATE DATE,
    SAL NUMBER(7, 2),
    COMM NUMBER(7, 2),
    DEPTNO NUMBER(2));

CREATE TABLE DEPT(
    DEPTNO NUMBER(2),
    MGRNO NUMBER(2),
    DNAME VARCHAR2(14),
    LOC VARCHAR2(13));

CREATE TABLE BONUS(
    ENAME VARCHAR2(10),
    JOB VARCHAR2(9),
    BONUS NUMBER);

CREATE TABLE SALGRADE(
    GRADE NUMBER,
    LOSAL NUMBER,
    HISAL NUMBER);

CREATE TABLE GRADENUM(
    GRADE NUMBER,
    NUM NUMBER,
    HIGH NUMBER);

1. Add a constraint to table 'EMP' that checks whether 'DEPTNO' refers to column 'DEPTNO' in table 'DEPT'. The constraint should also guarantee that once a department in the 'DEPT' table is deleted, the employee records of that department in table 'EMP' are also deleted. [4 points]

2. Create a constraint that checks the attribute 'SAL' of table 'EMP' whether the salary of employees is higher than 3000 and lower than 12000. Also, create a primary key constraint on the empno and create a constraint on the ename to be unique. [6 points]
3. Create a trigger that displays the employees' average bonus, before a record in table 'BONUS' is updated or before a new record is inserted. [5 points]

4. Create a trigger that does the following: If a department’s deptno is updated then update the employees’ depno also who works for the department. [5 points]

5. Create a trigger that does the following: When the employees’ salary is modified to some extent (more than 25% or less than 20%), then display the error message. [5 points]
**Exercise 4 (Function, Block and Procedure) [25 points]**

We assume the same database schema as in Exercise 3.

1. Create a function that calculates the average of all employees' salaries of a given department.
   - **input:** PDNAME: department name
   - **return:** AVG_SAL: average of all employees' salaries

   Then write an anonymous block to call your function and output the result. [6 points]

2. Write a block which updates the 'EMP' table. If an employee's salary is less than 3500, then increase his salary by 6 percent, and output the employee’s name, original salary and new salary. Make use of the cursor concept. [6 points]

3. Create a procedure that takes department number and changes the manager for the department to the employee in the department with highest salary. [6 points]

4. Write a procedure which first groups the salaries of employees. If one's salary falls into a salary boundary bounded by 'LOSAL' and 'HISAL' in table 'SALGRADE', then we consider this employee in the group of this 'GRADE'. Then insert records to table 'GRADENUM', set 'NUM' as the number of employees whose salary fall into its boundary. If 'NUM' > 5, set 'HIGH' = 1, otherwise, set 'HIGH' = 0. [7 points]