**Information and Database Management Systems I**

**(CIS 4301 UF Online)**

Fall 2024

Instructor: Alexander Webber

TA: Kyuseo Park

**Homework 5**

| | |
|---|---|
| Printed Name: | Kyle Lund |
| UFID: | 29501039 |
| Email Address: | kyle.lund@ufl.edu |

**Instructions**: Please provide your answers to the questions of the following pages in Word or handwritten on separate sheets of paper. Mark clearly to which question each answer belongs. Then convert or scan your work into PDF (the latter by using either a scanner or a suitable scanner app on your smartphone). Note that *only the PDF format* is allowed and that your submission must be a *single PDF file*. Finally, upload your PDF file into *Canvas* and follow the instructions there.

**Note**: All homework assignments are designed for a period of two, three, or even four weeks (see course deadline sheet). This means they cannot be solved in two or three hours but require a considerable amount of time and effort. Therefore, the first recommendation is to start with them as soon as they are posted. The second recommendation is to distribute the work on a homework assignment over the entire available period. The third recommendation is to submit the homework solutions *on time before the deadline*.

**Pledge** (Must be signed[1] according to the UF Honor Code):

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

Kyle Lund

Student signature

[1]Each student is obliged to print out this page, fill in the requested information in a handwritten and readable manner, make the *handwritten* signature, scan this page into PDF, and put this page as the first page of the PDF submission.

Question 1

A) Table is in 1NF

| Employees | |
|---|---|
| EmployeeID | EmployeeName |
| E01 | “John Doe” |
| E02 | “Jane Smith” |
| E03 | “Mike Brown” |

| Projects | | |
|---|---|---|
| ProjectID | ProjectName | ProjectDepartment |
| P01 | “Project Alpha” | “IT” |
| P02 | “Project Beta” | “Marketing” |
| P03 | “Project Gamma” | “Development” |

| EmployeeProjects | |
|---|---|
| EmployeeID | ProjectID |
| E01 | P01 |
| E01 | P03 |
| E02 | P02 |
| E03 | P02 |

B) Table is in 2NF

| BookPurchases | |
|---|---|
| Purchase ID | Book ID |
| 001 | B01 |
| 002 | B02 |
| 003 | B01 |
| 004 | B03 |

| Books | | | |
|---|---|---|---|
| BookID | Book Title | Author Name | Release Date |
| B01 | “The Great Gatsby” | “F. Scott Fitzgerald” | 1925 |
| B02 | “To Kill a Mockingbird” | “Harper Lee” | 1960 |
| B03 | “1984” | “George Orwell” | 1949 |

C) Table is not normalized

| CollectionBooks | | | | |
|---|---|---|---|---|
| Book | Author | Price | Publisher | Year |
| “Book A” | “Author X” | 20 | “Publisher Z” | 2020 |
| “Book A” | “Author Y” | 20 | “Publisher Z” | 2020 |
| “Book B” | “Author X” | 15 | “Publisher Y” | 2022 |
| “Book B” | “Author Z” | 15 | “Publisher Y” | 2022 |
| “Book C” | “Author Y” | 25 | “Publisher Z” | 2019 |
| “Book C” | “Author Z” | 25 | “Publisher Z” | 2019 |

Question 2

A) R is in 1NF as it has partial dependencies. New tables:

R0(A, C), S0 = {A->C}

R1(B, D), S1 = {B->D}

R2(A, B)

B) R is in 2NF but not 3NF as transitive dependencies. New Tables:

R0(A, B, C, E), S0 = {A->B, AB->C, A->E}

R1(C, D), S1 = {C->D}

C) R is in 2NF and 3NF, but not in BCNF.

D) New tables:

R0(ABCE), S0 = {AB->CE, C->ABE}

R1(DE), S1 = {E->D}

Question3

A) Lossy

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| a | b | c1 | d1 | e | f1 |
| a2 | b | c2 | d | e | f2 |
| a3 | b3 | c | d3 | e | f |
| a4 | b4 | c | d | e4 | f |
| C->E | | | | | |
| A | B | C | D | E | F |
| a | b | c1 | d1 | e | f1 |
| a2 | b | c2 | d | e | f2 |
| a3 | b3 | c | d3 | e | f |
| a4 | b4 | c | d | e | f |
| CF->D | | | | | |
| A | B | C | D | E | F |
| a | b | c1 | d1 | e | f1 |
| a2 | b | c2 | d | e | f2 |
| a3 | b3 | c | d | e | f |
| a4 | b4 | c | d | e | f |
| EC->A | | | | | |
| A | B | C | D | E | F |
| a | b | c1 | d1 | e | f1 |
| a2 | b | c2 | d | e | f2 |
| a3 | b3 | c | d | e | f |
| a3 | b4 | c | d | e | f |
| A->B | | | | | |

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| a | b | c1 | d1 | e | f1 |
| a2 | b | c2 | d | e | f2 |
| a3 | b3 | c | d | e | f |
| a3 | b3 | c | d | e | f |

B) Lossless
R1 ∩ R2 = CF
CF forms a super key.

C) Not dependency preserving, breaks down on EC->A

Question 4
A)
Step 1, minimal cover:
AD->E
B->CD
DE->AB
Step 2, Convert to relational schemas:
R0(ADE)
R1(BCD)
R2(DEAB)
Step 3, Check if candidate key is contained in a relation:
Candidate key DE contained in R2
Step 4, Test for containment:
R0 is contained in R1+R2
Step 5, return
R0(BCD)
R1(DEAB)

B)
R vs Ri: R is the original relationship schema, and each Ri contains a subset of the attributes in R, while ensuring that no transitive dependencies exist within any of the Ri's. Collectively, the Ri's represent the decomposed form of R.
F vs Fi: F is the complete set of functional dependencies present in the relational schema R, and each Fi represents the subset of functional dependencies that only involve the attributes present in the corresponding Ri.

C) The new schema in (a) is in BCNF. Both of the relations and their functional dependencies in (a) are in BCNF as there are no transient dependencies. Because both relations are in BCNF, the relational schema as a whole is in BCNF.

D)
First FD violation: B->CD
Decompose: R1(BCD); R2(ABE)

F1 = B->CD
F2 = Empty set

Next iteration:
R1 is BCNF
R2 has no FD's and is therefore BCNF

Final schema:
R1(BCD)
R2(ABE)

E)
R vs Ri: R is the original relationship schema, and each Ri contains a subset of the attributes in R, while ensuring that no transitive dependencies exist within any of the Ri's. Collectively, the Ri's represent the leaf nodes of the recursion tree which represents the decomposed form of R.
F vs Fi: F is the complete set of functional dependencies present in the relational schema R, and each Fi represents the subset of functional dependencies that only involve the attributes present in the corresponding Ri.

Question 5

A) In Department table: foreign key (managerID) references Employee(eID)

B)
```
CREATE TABLE EmployeeProjectsCount (
        pID int not null,
        numOfEmployees int not null,
        primary key (pID)
        CHECK(numOfEmployees >= 0);
```

C)
```
CREATE OR REPLACE TRIGGER trg_after_employee_project_delete
AFTER DELETE ON EmployeeProject
FOR EACH ROW
BEGIN

  -- Decrement numOfEmployees
  UPDATE EmployeeProjectsCount
  SET numOfEmployees = numOfEmployees – 1
  WHERE pID = :OLD.pID;

  -- Check if numOfEmployees is now 0
  DECLARE
    remaining_employees NUMBER;
  BEGIN
    SELECT numOfEmployees
    INTO remaining_employees
```

```
    FROM EmployeeProjectsCount
    WHERE pID = :OLD.pID;

    -- If numOfEmployees is 0, update endDate
    IF remaining_employees = 0 THEN
      UPDATE Project
      SET endDate = SYSDATE
      WHERE pID = :OLD.pID;
    END IF;
  END;
END;
```