# FALL 09 MIDTERM SOLUTION

## Problem 1

1) FALSE
2) TRUE
3) TRUE
4) FALSE
5) TRUE
6) FALSE
7) FALSE
8) FALSE
9) TRUE
10) FALSE

## Problem 2

a) See Figure 1.

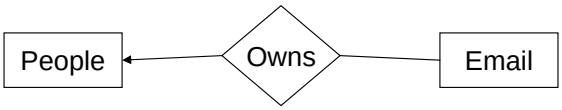


Figure 1: Problem 2a

b) See Figure 2.

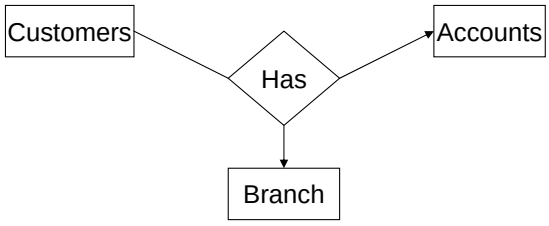


Figure 2: Problem 2b

c) See Figure 3.

d) 1. ER-style approach:
Account(<u>AccountID</u>, Balance)
CheckingAccount(<u>AccountID</u>, Overdraft)

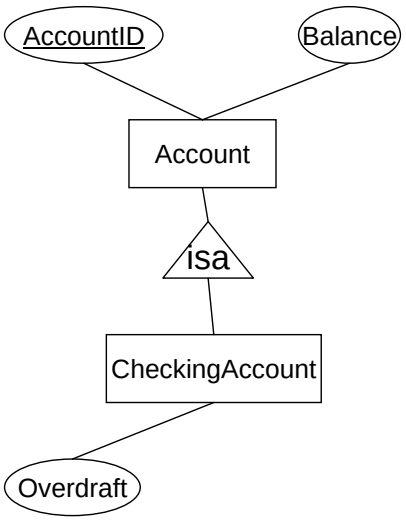


Figure 3: Problem 2c

2. Object-Oriented approach:
   Account(AccountID, Balance)
   CheckingAccount(AccountID, Balance, Overdraft)

e) The Object-Oriented approach uses the least amount of storage since each object can only belong to a single table. While in the ER approach, CheckingAccount may store fewer attributes, but all its tuples are repeated in Account.

f) 1. Manufacturer is nothing but a name and it's at the "one" end of any relationship, hence it should not be an entity set.

   2. See Figure 4.

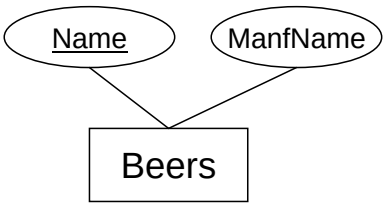


Figure 4: Problem 2f

# Problem 3

a) (3 points) 1 point for identifying both update and delete anomaly, 1 each for examples.
1. Update anomaly: A doctor may have more than one patient, so an update anomaly may result if a doctor's name is changed for a given doctorID for only one patient.
2. Delete anomaly: Deleting patients' diagnosis could delete the name of their doctor.

b) (4 points) answer should provide similar information as below:
Case 1: There are no nontrivial FDs - only nontrivial FDs can violate BCNF.
Case 2: A → B holds, but B → A does not. A is the key, and the only nontrivial FD is A → B; no BCNF violation.
Case 3: B → A holds, but A → B does not. Symmetric to Case 2.
Case 4: Both A → B and B → A hold. A and B are both keys; no BCNF violation.

c) (4 points) answer should provide similar information as below:
Definition of 3NF is that the left hand side of any applicable FD is a superkey or the right hand side is a part of a key; that is, for any FD A?B that holds in R, A is a superkey or B is a part of a key. If every key in R is a single attribute, B on the right of the FD cannot be a part of of a key. Therefore, only way for relation R to satisfy the condition for 3NF is that, for any FD A → B, A is a superkey of R, and R thus satisfies the condition for BCNF.

d) (4 points) 1 point on selecting violating FD, 1 point for expanding selected FD, 1 point each for 2 decomposition steps.
1. Decomposing R on C → A, D (expanded to C → A, B, D)
R1(A, B, C, D)
R2(C, E)

2. Decomposing R1 on D → B
R1.1(A, C, D)
R1.2(B, D)

Therefore: final decomposition is
R1.1(A, C, D)
R1.2(B, D)
R2(C, E)

# Problem 4

a) a,b,c

b) 1) $\Pi_{Shows.cinema}(\sigma_{ticketPrice<8}(Shows) \bowtie \sigma_{movieFanName=James}(Likes))$
2) $\Pi_{Shows.cinema}(\sigma_{isStudent=true}(Frequents \bowtie MovieFan)) - \Pi_{Shows.cinema}(\sigma_{isStudent=false}(Frequents \bowtie MovieFan))$
3) $\Pi_{shows.movieFanName}(Shows \bowtie Frequents \bowtie Likes \bowtie \sigma_{isStudent=true}(MovieFan))$

# Problem 5

a) **SELECT Office, Email**
**FROM Employee**
**WHERE FirstName='John' AND LastName='Smith'**

b) **SELECT Days**
**FROM Employee, Vacations**
**WHERE Employee.EmployeeID=Vacations.EmployeeID AND Employee.FirstName='John' AND Employee.LastName='Smith'**

c) **SELECT FirstName, LastName**
**FROM Employee**
**WHERE EmployeeID NOT IN(SELECT DISTINCT EmployeeID FROM Vacations)**

d) **SELECT DepartmentName**
**FROM Department, Employee, Vacations**
**WHERE Department.DepartmentID=Employee.DepartmentID**
**AND Employee.EmployeeID=Vacations.EmployeeID**
**GROUP BY DepartmentName**
**HAVING SUM(Days) = (SELECT max(t1.DAYS) FROM**
**(SELECT Employee.DepartmentID, SUM(Days) as DAYS**
**FROM Department, Employee, Vacations**
**WHERE Department.DepartmentID=Employee.DepartmentID**
**AND Employee.EmployeeID=Vacations.EmployeeID**
**GROUP BY Employee.DepartmentID) as t1)**

# Problem 6

a) CHECK is not checked if the database modification does not change the attribute with which the constraint is associated, e.g. foreign key constraint. Suppose we have two tables Department and Employees, and DepartmentName is a foreign key in Employees. Modifications on the DepartmentName of the Department table are not reflected in the Employees table. Another example is shown in Example 7.7 (page 321 in the book).

b) Changes to the referenced attributes are mimicked at the foreign key. When we make changes (i.e. delete/update) a tuple from a relation, to maintain referential integrity, the system will also make changes to the referencing tuples from the referenced relation.

c) Triggers allow users to specify when the check occurs, where as for assertions are checked every INSERT/DELETE/UPDATE.