**Ch2-NormEx1-2**

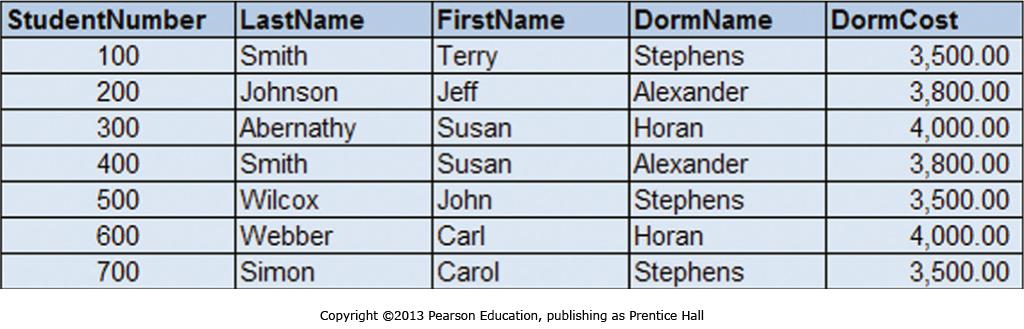
**In-Class:** Chapter 2 Normalization Examples 1 and 2 - In-Class \_\_\_ \_\_\_ \_\_

Perform the following steps for the STU\_DORM and EMPLOYEE relations below:

1. Identify all candidate keys.
2. Look for the functional dependencies in the relation.
3. Ask if any determinants exist that are not candidate keys.
   1. Place the columns of the functional dependency in a table of their own.
   2. Make the determinant the primary key of the new table.
   3. Leave the determinant as a foreign key in the original table.
   4. Create a referential integrity constraint.

**Normalization Example 1**

Normalize the following STU\_DORM relation:



Step 1: The candidate keys are: StudentNumber, (LastName, FirstName)

Step 2: The function dependencies are:

DormName 🡪 DormCost

StudentNumber 🡪 FirstName, LastName, DormName, DormCost

(FirstName, LastName) 🡪 StudentNumber, DormName, DormCost

Step 3: Are there any determinants that are not candidate keys? Yes.

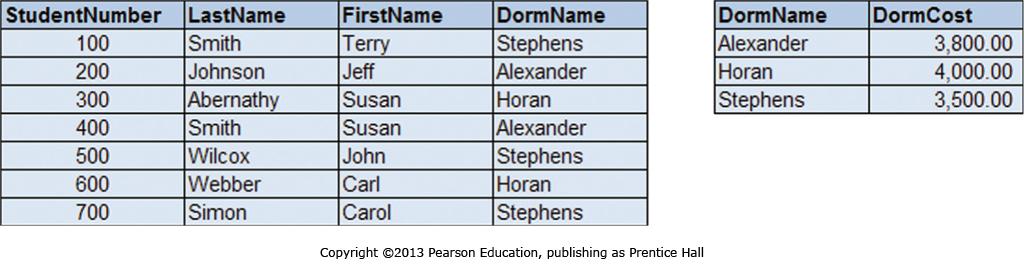
DormName is a determinant, but not a candidate key.

Step 3a, b: DORM ( DormName, DormCost )

Step 3c: STUDENT ( StudentNumber, LastName, FirstName, *DormName* )

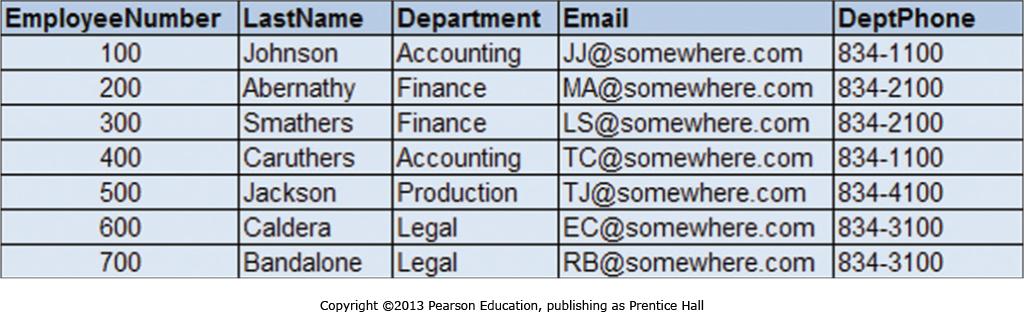
Step 3d: DormName in STUDENT must exist in DormName on DORM.

Hint: The resulting STUDENT and DORM relations will be:



**Normalization Example 2**

Normalize the following EMPLOYEE relation:



Step 1: The candidate keys are: EmployeeNumber, Email

Step 2: The functional dependencies are:

Department 🡪 DeptPhone

EmployeeNumber 🡪 LastName, Department, Email, DeptPhone

Step 3: Are any determinants not candidate keys? Yes.

Department is a determinant, but not a candidate key.

Step 3a, b: DEPARTMENT ( Department, DeptPhone )

Step 3c: EMPLOYEE ( EmployeeNumber, LastName, *Department*, Email )

Step 3d: Department in EMPLOYEE table must exist in Department column in DEPARTMENT table.

Hint: The resulting EMPLOYEE and DEPARTMENT relations will be:

