Functional Dependencies and Normalization

**Exercise 1:**

Consider the following relation:

Stock(#prod, #dep, pname, quantity)

Determine the functional dependencies on Stock.

Functional dependency exists when one attribute in a table uniquely determines another attribute. Functional dependency is represented as X🡪Y. X and Y can be composite

Following is the functional dependencies on Stock:

#prod🡪 pname, quantity

**Exercise 2:**

Consider the following relation:

Plane(#plane, type, constructor, capacity, owner)

Determine the functional dependencies on Plane.

Following is the functional dependencies on Plane:

#plane 🡪 type, constructor, capacity, owner

**Exercise 4:**

Is Stock (of Exercise 1) in 3NF? Why?

Stock is not in third normal form. According to the third normal form, the table must be in second normal form and any non-key attribute should not describe any non-key attribute.

In order for the table to be in second normal form , the table must be in first normal form and each non-key attribute must depend only on primary key. The table is already in first normal form. But pname and quantity depends only on #prod. So decompose the table into two tables as follows:

Stock1(#prod, pname, quantity)

Stock2(#prod,#dept)

Now the tables are in third normal form.

Is Plane (of Exercise 2) in 3NF? Why?

Plane(#plane, type, constructor, capacity, owner)

Plane is in third normal form as all the non-key attributes depend on the primary key #plane.

Is Stock (of Exercise 1) in BCNF? Why?

Stock is not in BCNF as it is not satisfying third normal form.

The table Stock is decomposed into two tables to satisfy third normal form.

Stock1(#prod, pname, quantity)

Stock2(#prod,#dept)

Stock1 and Stock2 satisfy BCNF.

**Exercise 5:**

What are the properties of the BCNF decomposition algorithm?

Apply the decomposition algorithm on Stock.

The properties of the BCNF decomposition algorithm are:

* Dependencies that exist in the table are not always preserved.
* The decomposition can be lossless. In other words, the data is not lost when the table is decomposed into two or more tables.

**Applying the decomposition algorithm on Stock:**

**step 1:** Find a FD that violates BCNF.

The functional dependency that violates BCNF is #prod🡪 pname, quantity.

**step 2:** Split the relation.

Stock1(#prod, pname, quantity)

Stock2(#prod,#dept)

Now, both Stock1 and Stock2 are in BCNF.

**Question 3**

