Final — Data at Scale

Problem 1:

The following database scheme is for a movie rental agency. The MovieFormat is VHS, CD, DVD, or BlueRay. The agency has an inventory number for each copy of a movie that they are renting.

MovieRentals(FirstName, LastName, MovieTitle, MovieFormat, MovieInventoryNumber, AddressStreet, AddressCity, AddressZipCode, TelephoneAreaCode, TelephoneNumber, Salutation).

MovieInventory(MovieTitle, MovieFormat, MovieType, MovieLeadingCast, MovieInventoryNumber)

(1) Explain why this scheme suffers from anomalies by example.
(2) Explain how to transform this scheme in BCNF.
(3) Assume that the original scheme would be implemented in an XML database. Give an example for a record each for each table.
(4) Assume that the database is changed to a single table. How would the XML record than look like?
(5) Implement the information in the database for a graph database like Neon.

Problem 2:

How does Zookeeper prevent inconsistencies in replicated data in the nodes.

Problem 3:

A very large file contains information on car ownership in a country. Besides data on the owner, the data also contains the current value of the car. Another, separate, very large file contains data on boat owners. Similarly to the first file, each record has data on the owner and the presumed value of the boat. Neither file can fit in the distributed cache of a cluster.

Use the map-reduce paradigm efficiently in order to find the name of all owners that have a boat that is more valuable than their car. Of course, owners can have more than one boat or more than one car, in which case the condition is that one of their boats is more valuable than one of their cars.

Luckily, for your, owners are identified by their tax-payer number in both files.