# Sample Final

1. A relational scheme for invoice management consists of three tables:

```
Client (ClientID: int, Name: varchar(50), Billing_Address1:
varchar(100), Billing_Address_City: varchar(50), Billing_Postal_Code:
varchar(50), Payment_information: varchar(100))

Item (ItemID: int, Description: varchar(100), Price: float)

Invoice(ClientID, ItemID, quantity: int, price: float, Invoice_Date:
date, Shipping_Date: date, Shipping_Address: address).
```

(a) Explain whether this scheme has update anomalies and what they would be. Propose a normalization.

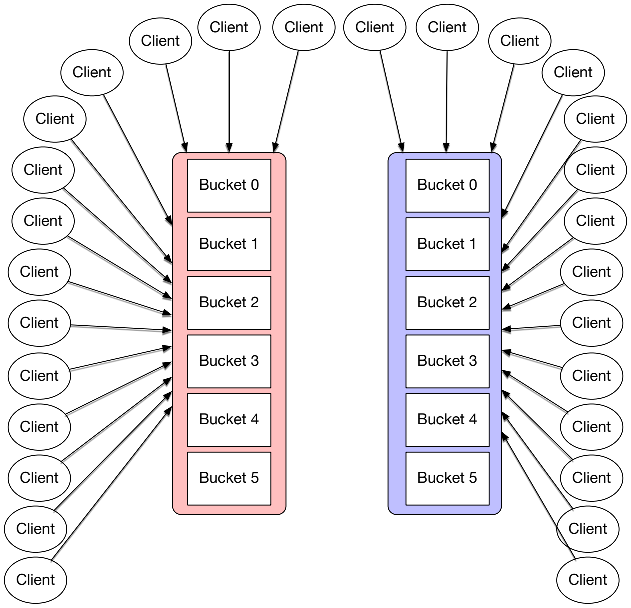
(b) Give an example for a JSON implementation of a record in each of the above relations. This would give us three different files.

(c) We can also “denormalize” the schema and use a single JSON file to store all the information. What would the JSON structure look like?

(d) What are the advantages and disadvantages of a “normalized” and a “denormalized”

(e) Develop a Neon-like graph representation for the system.

2. One possibility to make LH* survive node failure is to mirror the LH* file. In this scenario, each bucket exists in two incarnations. The set of clients is divided into an A-set and a B-set depending with which set of buckets the client is allowed to interact. Discuss some problems that can arise if the actions in the A-set and the B-set of buckets are not closely synchronized and an A-client wants to access a B-bucket because of an unavailability.



Another possibility does not coordinate between the two bucket sets but have them be administered independently. This could be because one set of servers has less RAM or worse performance. In this scenario, an A-client only interacts with the A-set for reads, but any writes have to be done by the client in two different buckets, one in the A-set and the other one in the B-set of buckets. Show that even though the number of buckets are different, it is possible to find all records in the B-set that belong to a certain B-set bucket. Thus, it is possible to recover a lost bucket on either side.

3. Explain why a simple scheme where a client writes to all replica can lead to inconsistencies.

4. How does Zookeeper prevent inconsistencies in replicated data?

5. Explain how updates in a system with read and write quora could actually function.

6. Why does Chubby use coarse-grained locks?