

To make an architecture of a database system, always think about the workloads (read and write)

JOINS are very expensive (use the network)

pg 62: user lookup = give me this particular user

pg 64 : TTL = time to live = 1 week => data will be reset in 1 week

pg 65 : table user-shorturl : give the shorturl for a special user

De-normalize data eliminate the use of JOINS

HBASE

pg 67 : sparse do not save all values (do not store null values Vs normalized database system)

pg70 : the number of column families should be reasonable because (hint: they will be distributed over machines)

pg 71: cell = atom of a database management system

pg75 : CRUD (create Read update delete)

pg76 : combine with spark to support SQL

pg78 : we can read and write at the same time

NTP :network time protocol (machines: roughly the same clock => help for synchronization)