# Qzui4

# MMDB

### 1. MMDB make use of modern hardware to store data. Mention three data storage layers that are commonly used in MMDBs e.g. SAP HANA. Also mention what is the usage of each layer, i.e. each layer stores what type of data?

CPU Caches

Main memory contain majority of data.

Disk is used to store large amount of data which do not fit into memory and log data.

Flash Drives

### 2. Regarding the concurrency control in MMDBs, does it depend on Locking or Multi-versioning? At which granularity it works (table cell, row, column, or entire table)? Draw a figure showing the concurrency control mechanism and the handling of suspended (waiting) transactions.

 Lock-based concurrency control.

 Both record-level and table-level

 Two bit locking, the first bit is for checking locking status, and the second bit is for waiting list status. When the first bit is 1, new coming transactions will be suspending in waiting list. On unlocking, unlocking transaction needs to check the second bit to awake suspended transactions in waiting list.

1

1

Record

t1

Waiting List

t2

### 3. The team has presented three ways to speed up queries and save space in MMDBs that are not applicable to Disk-based DBs. Mention Two of three ways and explain why they can not be used for traditional DBs.

 Hashing for random access.

 T-trees.

 Using pointer

## NoSQL DBs

### 1. Give names of three NOSQL DBs and state the data model of each one?

 MongoDB : Document store.

 Redis: Key-value pair.

 HBase: Column family.

### 2. Querying CouchDB involves three steps. Write down these steps?

 1. Define a view. Map functions take documents and emits key/values pairs.

 2. Construct B-tree index. Couch DB storage engine constructs a B-tree index.

 3. Query the view. Reduce functions operate on the sub tree to do aggregation.

### 3. Describe the figure below. State what does it show, and how the illustrated mechanism work?



The plot above shows the concurrency control using lock. Database system will lock the data on changing and commit to disk. Read queries must wait until system commit, and all queries will get same version of data.



The plot above shows multi-version concurrency control technique. Every read query will get latest version of data. The data state do not rely on the global state.