ADM Solutions

Data Models and Access Paths

(4 points) In the motivation of his seminal paper *A relational model of data for large shared data banks*, Codd argues that Indexing can cause problems in data banks. What is the reason for using indexes in a database? Identify the dependency and explain shortly.

Index is thought of as a purely performance-oriented component of the data representation. It tends to improve response to queries and updates but it slows down response to insertions and deletions. Index is a redundant component of the data representation.

Application programs should remain invariant if indices are created or destroyed. In the case of TDMS and IMS, the user's application logic does not depend on the existence of the unconditionally provided indices. In case of IDS additional indexing chains can be used and application programs taking advantage of the performance benefit of these indexing chains do not operate correctly if these chains are removed.

(b) (5 points) Given the following data structures 2 and 5 from this paper.

Structure 2. Parts Subordinate to Projects

File	Segment	Fields
F	PROJECT	project #
		project name
		project description
	PART	part #
		part name
		part description
		quantity-on-hand
		quantity-on-order
		quantity committed

Structure 5. Parts, Projects, and Commitment Relationship as Peers

File	Segment	Fields
F	PART	part #
		part name
		part description
		quantity-on-hand
		quantity-on-order
G	PROJECT	project #
		project name
		project description
H	COMMIT	part #
		project #
		quantity committed

Which data model from the lecture would you choose to represent structure 2 and why?

Might be document databases (Need some time to solve)

(c) (1 point) Which database system would you use for structure 5?

PostgreSQL

Extensible Record Stores

(4 points) Explain the differences between the Memtable and Immutable Sorted Data Files.

Memtable:

- Data structure in main memory
- Collects most recent writes of fixed sized with timestamp
- In case of upsert, contains value
- In case of delete, value is empty (tombstone)
- When memtable is full, it flushes to immutable a sorted data file

Immutable sorted data files:

- One written, data files cannot be changed
- Write-optimized storage model. Very fast writing.
- When memtable is full, data is written to a sorted data file
- Sort records by key

(b) (6 points) In an extensible record store database, we use a Bloom Filter of length=16 with three hash functions h_1, h_2, h_3 to determine set membership for the data stored in a file. The following keys are inserted in the Bloom filter:

$Key_1 : h_1(key_1) = 1, h_2(key_1) = 4, h_3(key_1) = 9$

$Key_2 : h_1(key_2) = 5, h_2(key_2) = 8, h_3(key_2) = 13$

Which of the results (true positive, false positive, true negative, false negative) would be returned by the Bloom filter for the following queries?

Insert key1: $Key_1 : h_1(key_1) = 1, h_2(key_1) = 4, h_3(key_1) = 9$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0

Insert key2: $Key_2 : h_1(key_2) = 5, h_2(key_2) = 8, h_3(key_2) = 13$

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	1	1	0	0	1	1	0	0	0	1	0	0

$Query_1$: Search for Key_x where $h_1(key_x) = 4, h_2(key_x) = 8, h_3(key_x) = 9$

False positive: two 1's from key1 and other 1 from key2

$Query_2$: Search for Key_y where $h_1(key_y) = 5, h_2(key_y) = 8, h_3(key_y) = 13$

True positive: all the 1's from key2

$Query_3$: Search for Key_z where $h_1(key_z) = 8, h_2(key_z) = 13, h_3(key_z) = 15$

True negative

Consistent Hashing

Year	Nominee	Winner?	key	Filename	h(key)
2021	Jeffrey Andrew Weinstock	no	2021_weinstock.txt	weinstock.txt	10
2021	Maria Dahvana Headley	no	2021_headley.txt	headley.txt	17
2021	Jo Fletcher	no	2021_fletcher.txt	fletcher.txt	1
2021	Clive Bloom	no	2021_bloom.txt	bloom.txt	8
2021	Charles Coleman Finlay	yes	2021_finlay.txt	finlay.txt	5
2020	Leslie S. Klinger	no	2020_klinger.txt	klinger.txt	5
2020	Ellen Oh	no	2020_oh.txt	oh.txt	15
2020	Sheree Thomas	no	2020_thomas.txt	thomas.txt	4
2020	Charles Coleman Finlay	no	2020_finlay.txt	finlay.txt	5
2020	Ebony Elizabeth Thomas	yes	2020_thomas.txt	thomas.txt	4

RC: 2 WC: 3

Time	Event	A	B	C	D
0	Nodes A-D are registered	x	x	x	x
1	Put weinstock.txt	x		x	x
2	Put headley.txt	x	x		x
3	Put fletcher.txt	x	x	x	
4	Node A unavailable	x			
5	Get thomas.txt		x	x	
6	Find new leader	x	x	x	x
7	Copy replica	x	x	x	x

After time 3:

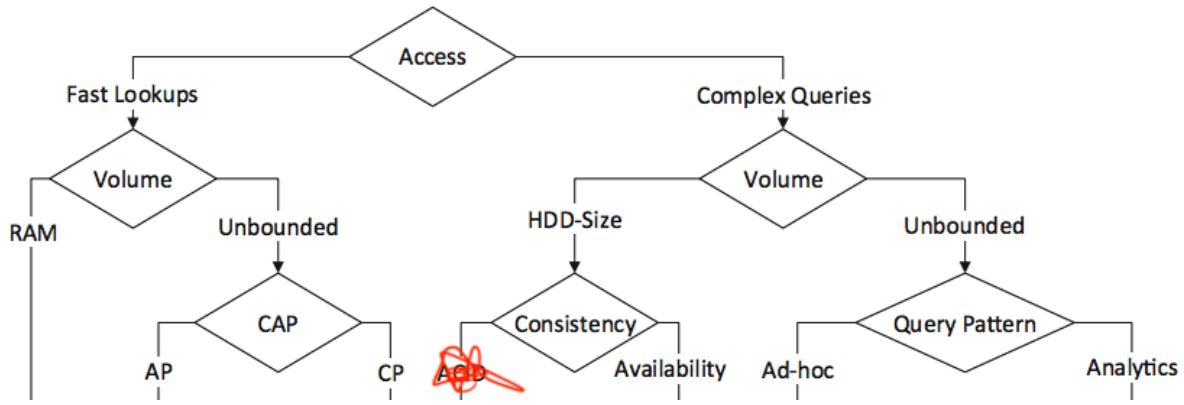
A-token 1	L/F	B-token 7	L/F	C-token 10	L/F	D-token 18	L/F
weinstock.txt	F			weinstock.txt	L	weinstock.txt	F
headley.txt	F	headley.txt	F			headley.txt	L
fletcher.txt	L	fletcher.txt	F	fletcher.txt	F		

After time 7:

A-token 1	L/F	B-token 7	L/F	C-token 10	L/F	D-token 18	L/F
		weinstock.txt	F	weinstock.txt	L	weinstock.txt	F
		headley.txt	F	headley.txt	F	headley.txt	L
		fletcher.txt	L	fletcher.txt	F	fletcher.txt	F

Graph Databases

a)



Only the red marked part: Neo4j

b) Short term solution:

- Vertical scaling. Adding more resources (CPU/RAM/DISK) to the same server.

Long term solution:

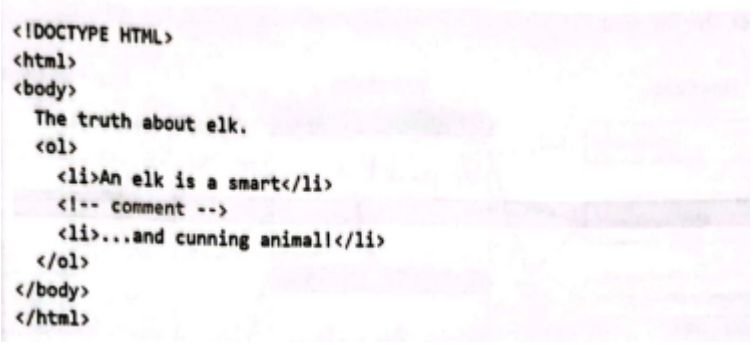
- When complex queries have to be optimised for latency, as for example in social networking applications, MongoDB is very attractive, because it facilitates expressive ad-hoc queries.
- HBase and Cassandra are also useful in such a scenario, but excel at throughput-optimised Big Data analytics, when combined with Hadoop.
- Scale MongoDB horizontally through sharding (preferred) and replica sets.

Challenges

- Serving my customers while migrating from one database from another

- Converting whole data from one format to another: from nodes and edges to documents and collections

Document Databases



```
<!DOCTYPE HTML>
<html>
<body>
  The truth about elk.
  <ol>
    <li>An elk is a smart</li>
    <!-- comment -->
    <li>...and cunning animal!</li>
  </ol>
</body>
</html>
```

a) Yes

An XML document with correct syntax is called "Well Formed".

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested
- XML attribute values must be quoted

b) No, Valid XML documents have a correct structure and content according to a schema

Some extra description which is not needed for answer but understanding

What is a DTD?

DTD stands for Document Type Definition.

A DTD defines the structure and the legal elements and attributes of an XML document.

What is XML Schema?

An XML Schema describes the structure of an XML document, just like a DTD.

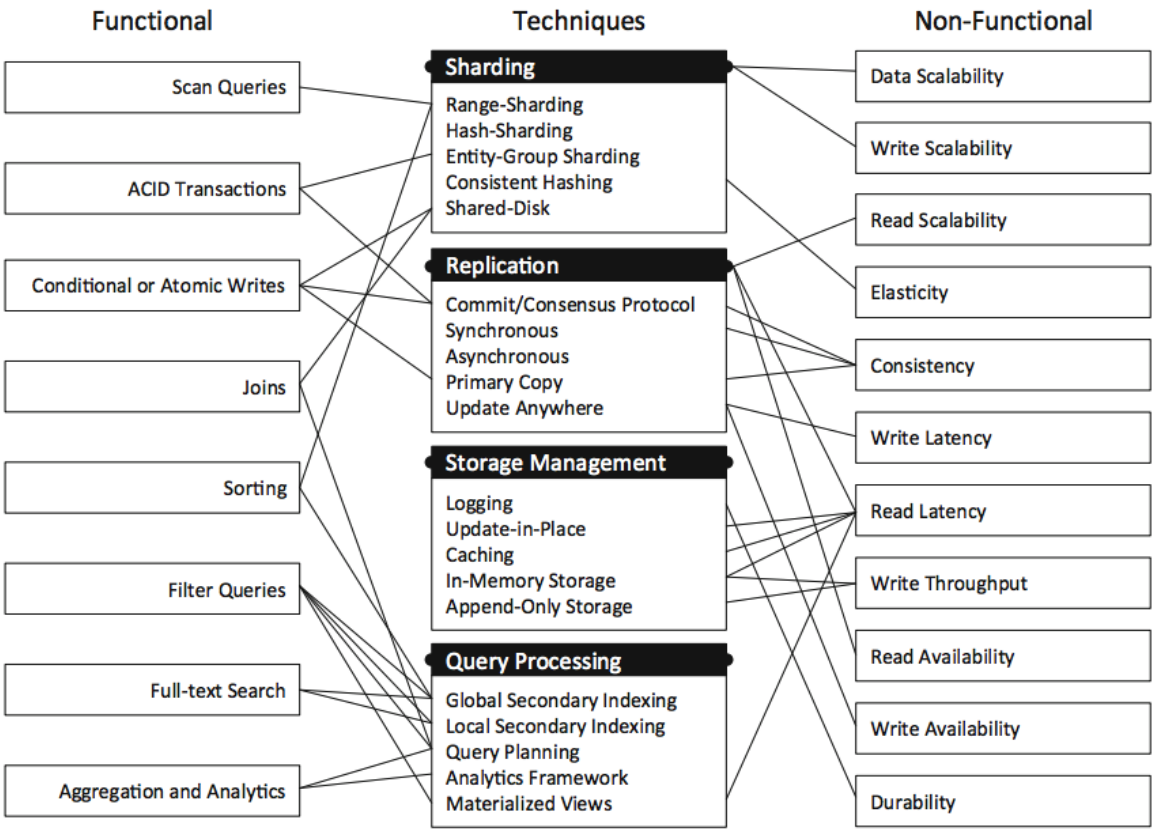
An XML document with correct syntax is called "Well Formed".

An XML document validated against an XML Schema is both "Well Formed" and "Valid".

c) Everything in an XML document is a node. So, 9 Nodes
   i) https://www.w3schools.com/xml/dom_nodes.asp
d) 1. N 2. E 3. E
e) Hashed Sharding: Documents are distributed according to an MD5 hash of the shard key value. Guarantees a uniform distribution of writes across shards, which is often optimal for ingesting streams of time-series and event data.

NoSQL Toolbox

a)



b) Pick one relocation technique and explain it
    i) Phase 1: Voting phase
        1) Coordinator sends message to all agents to „prepare" for operation
        2) Agents respond with „ready"
    ii) Phase 2: Decision phase
        1) Coordinator sends „commit" message to all agents
        2) Agents respond with „acknowledge_commit"
    iii) Phase 3: Optional abort phase (hopefully rare)
        1) If some clients did not respond: Coordinator sends „abort" to all clients who voted „ready" ; they have to undo / rollback their changes
        2) Coordinator might restart the protocol

c) update-in-place

Introduction:

- What kind of redundancy do we not want in relational DB? Logical Redundancy
- What is a page in DB?
- Page buffer
- Page miss?
- Pointer swizzling

Relational Databases:

- ACID
- Normalization (1NF, 2NF, 3NF)

Extensible Record Stores

- Column families
- Row key
- To identify a cell, combine rowkey:columnfamily:columnqualifier
- Time and "Upserts"
- Immutable sorted data files and Memtable
- Tombstones
- Compaction
- Redo logging

Distributed Db

- Distribution Transparency
- When you do fragmentation what do you consider
- Why do we need replication?
- Why do we need synchronization?
- Replication algorithm? Two-Phase Commit
- Paxos algorithm
- Strong consistency vs weak consistency
- CAP theorem

Graph Databases

- Property graph
- Convert table to property graph
- Convert ER to property graph how many nodes and edges