1. Data Models and Access Paths

(a) (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.

(4)

In large shared data banks we use Indexing for easy access, and to maintain refrencing among the data.

But the problem associated with Indexing is it is very costly and also difficult to maintain. cause Issue in data migration if get broken.

(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?

(2)

We can use Relational Data model because in structure 2 we can see a Relation between Project and Parts.

Because part is some how dependent on Project sub-ordinate.

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

(0)

Non-Relational Database - Direct Accessible.

2. **Extensible Record Stores**

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

Memtable : Data which is frequently used / called is stored in main memory table for fast access. (✓)

Immutable : Data files which are accessible by using Range Query by providing the contigeous Range. f.

05

(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?

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

True Negative. f

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

True Positive ✓

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

False Negative. f

3. Consistent Hashing

The World Fantasy Awards are given each year by the World Fantasy Convention for the best fantasy fiction and art published in English during the preceding calendar year. The awards have been described by sources such as The Guardian as a "prestigious fantasy prize" and one of the three most renowned speculative fiction awards, along with the Hugo and Nebula Awards (which cover both fantasy and science fiction.

The organizers of these Awards collect documents containing the expert opinion written for the nominees per year. The system also hashes the keys to a value between 0 and 19. This results in the following data structure:

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

We use four nodes to store data in our system, which uses dynamic hashing to distribute the data. The virtual nodes are named Node A (with token 1), Node B (token 7), Node C (token 10) and Node D (token 18). For fault tolerance, we configure the system to use 3 replicas, with a read quorum of 2 and a write quorum of 3.

(a) (5 points) During the operation of the system, documents are stored and queried, and nodes become unavailable. Below you find the timeline of operations. Please mark in the table which nodes are affected by the described events or operations with an X.

Timeline		Affected Nodes			
Time	Operation/Event	A	B	C	D
0	Nodes A-D are registered in the system	x	x	x	x
1	PUT(2021_weinstock.txt, "weinstock.txt")	x		x	x
2	PUT(2021_headley.txt, "headley.txt")	x	x		x
3	PUT(2021_fletcher.txt, "fletcher.txt")	x	x	x	
4	Node A become unavailable	x			
5	GET(2020_thomas.txt)		x	x	
6	Reorganization: Find new leader(s)		x	x	x
7	Reorganization: Copy replica(s)		x	x	x

(b) (5 points) The nodes store the documents based on their key. Below you find two tables that shall represent the status of the node storage at two different times: after time 3 and after time 7. Please fill out the correct content, and mark with "L" or "F" whether a Node stores the document as a Leader or as a Follower.

Storage on Virtual Nodes (until time 3)							
A - Token 1		B - Token 7		C - Token 10		D - Token 18	
Stored Keys	L/F	Stored Keys	L/F	Stored Keys	L/F	Stored Keys	L/F
2021-wainstock.txt				2021-wainstock.txt		2021-wainstock.txt	
2021-lendley.txt		2021-lendley.txt				2021-lendley.txt	
2021-fletcher.txt		2021-fletcher.txt		2021-fletcher.txt			

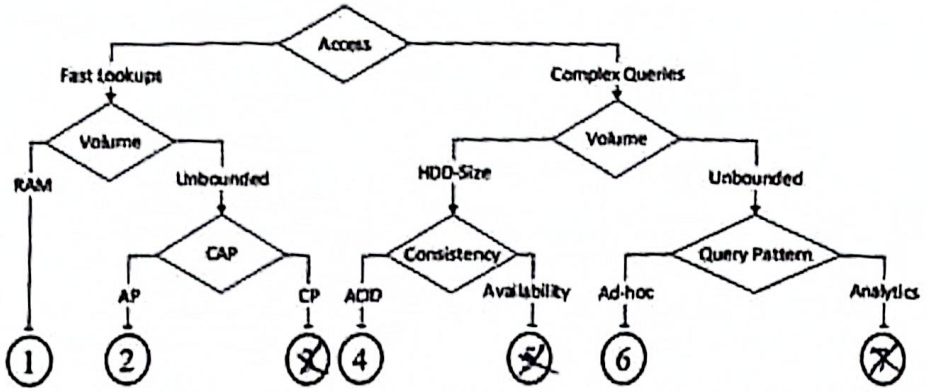
25

Storage on Virtual Nodes (after time 7)							
A - Token 1		B - Token 7		C - Token 10		D - Token 18	
Stored Keys	L/F	Stored Keys	L/F	Stored Keys	L/F	Stored Keys	L/F

Optional space for assumptions / remarks:

4. Graph Databases

(a) (2 points) In the NoSQL decision tree figure below, mark all cases where you can place graph databases with a cross.



(b) (8 points) A German company stores its customer data in a graph database to be able to represent relationships that they mine from social networks (who is friend of whom, who works in the same company) and discussion forums (who answers to whom).

After some while, the server used for this purpose comes to its limits, since the number of customers grows and they added more and more data per customer (including images and short videos with testimonials). Within this year, they will exceed the HDD size with the German customers only; next year, they plan to roll-out the system in whole Europe.

How would you address this problem? Please sketch a short-term and a long-term solution, use the correct technical terms to explain the strategies and mention challenges that might arise with that.

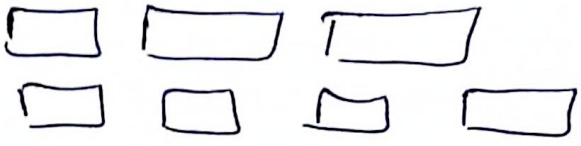
we can use the Document Database Mongo DB for this problem and we have to ~~about~~ plane about scale out so, we can scale our database accordingly ✓

This problem accur because of un-planed huge storage which is associated with relational Database.

Non-Relational Database is a good solution for such problems.

Because we can easily scale out (Horizontally).

Database servers



5. Document Databases

The following code example from `javascript.info` shows some HTML snippet:

```
1  <!DOCTYPE HTML>
2  <html>
3  <body>
4    The truth about elk.
5    <ol>
6      <li>An elk is a smart</li>
7      <!-- comment -->
8      <li>...and cunning animal!</li>
9    </ol>
10 </body>
11 </html>
```

(a) (1 point) Is this a well-formed XML document?

(a) No, It is a HTML Document. 0

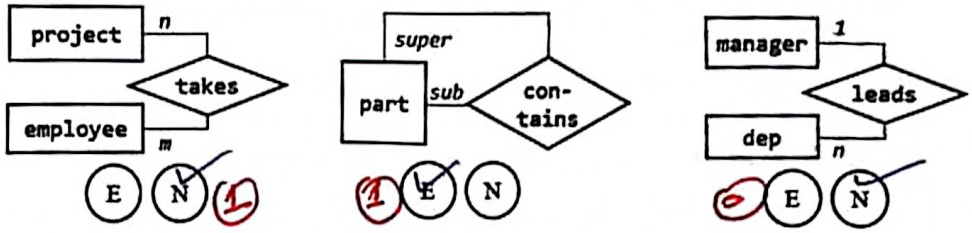
(b) (1 point) Is this a valid XML document?

(b) NO 0

(c) (1 point) How many nodes has the corresponding DOM tree (ignoring whitespace)?

(c) TWO NODES 0

(d) (3 points) In a document database system like MongoDB, one can choose between two modelling approaches: embedded data model (E) or normalized data model (N). Which of the models should be chosen for the following conceptual data model?



(e) (4 points) Name and shortly explain a sharding technique that is used by MongoDB.

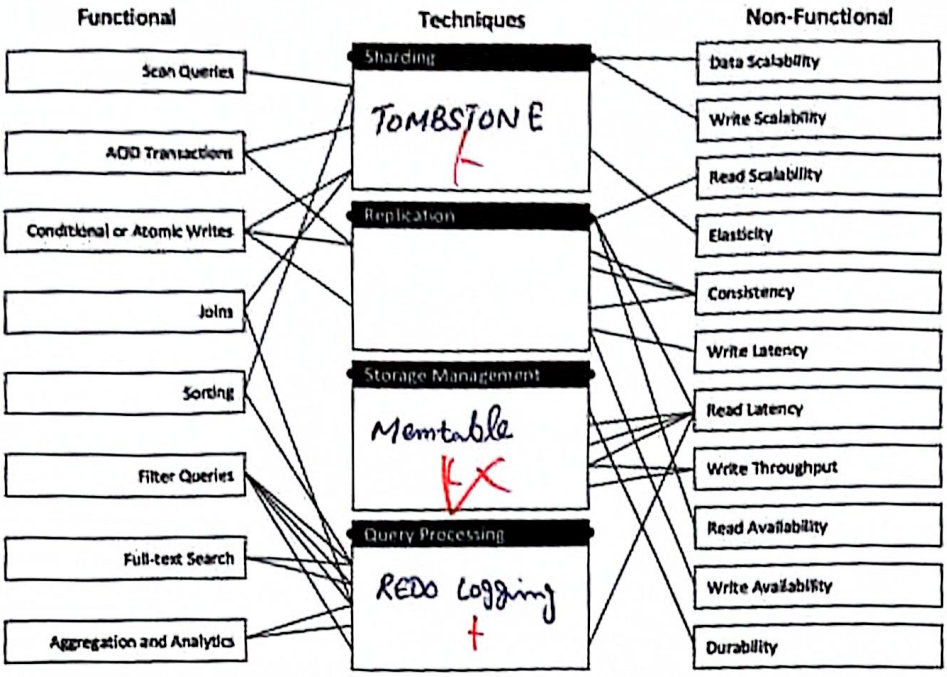
0 Tombstone is a Sharding Technique used by MONGO DB

In which we can ignore the old-version of data.

Remove the record if the version is old.

6. NoSQL Toolbox

The following figure (adapted from the paper *NoSQL database systems: a survey and decision guidance* by Felix Gessert et al. shows the so-called NoSQL toolbox. However, the techniques that connect the functional and non-functional system properties are missing.



0,5

(a) (4 points) Provide one example for each technique and write them in empty boxes of the figure above.

(b) (2 points) Pick one replication technique and explain it.

Backups: We can use backups as a replication technique because it will also helps up in Availability of Data and fast access.

0

(c) (4 points) In the following quote from the paper about a storage management technique, a central term is missing (denoted by —).

For in-memory databases, an — access pattern is ideal: It simplifies the implementation and random writes to RAM are essentially equally fast as sequential ones, with small differences being hidden by pipelining and the CPU-cache hierarchy. However, RDBMSs and many NoSQL systems (e.g. MongoDB) employ an — update pattern for persistent storage, too. To mitigate the slow random access to persistent storage, main memory is usually used as a cache and complemented by logging to guarantee durability. In RDBMSs, this is achieved through a complex buffer pool which not only employs cache-replace algorithms appropriate for typical SQL-based access patterns, but also ensures ACID semantics. NoSQL databases have simpler buffer pools that profit from simpler queries and the lack of ACID transactions.

Which term is missing?

(c) database

0

7. Query Languages

Consider the following example of a query to a database:

```
db.person.aggregate([
   {    $lookup: {
          from: "department",
          localField: "works_in",
          foreignField: "id",
          as: "dep_doc"
        } },
  {       $unwind: "$dep_doc"
   },
    {  $project: {"firstname":1, "lastname":1, "dep_doc.name":1
                  }} ])
```

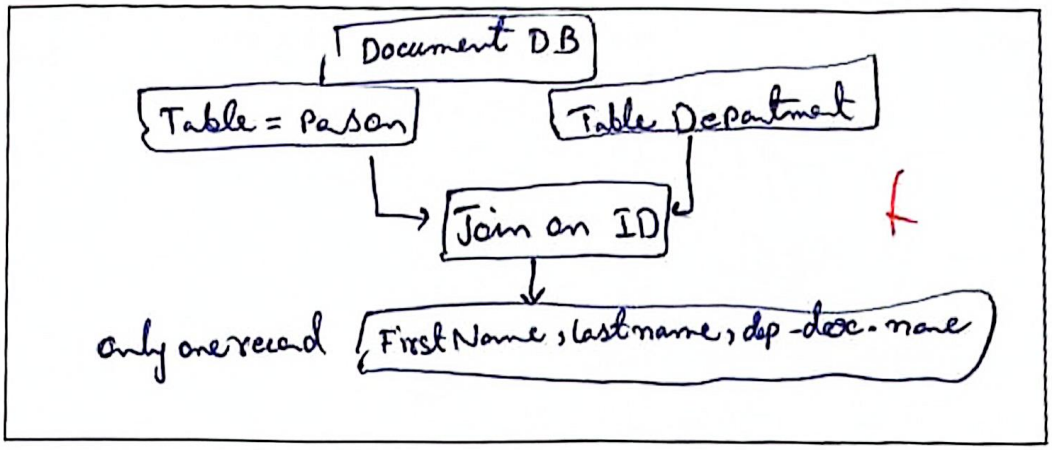
(a) (1 point) What query language is shown here?

(a) MONGO DB

(b) (1 point) Which database can answer such queries?

(b) ~~Extensive Record Store~~ Document Databases

(c) (4 points) What assumption about the underlying database schema can be drawn from that code snippet?

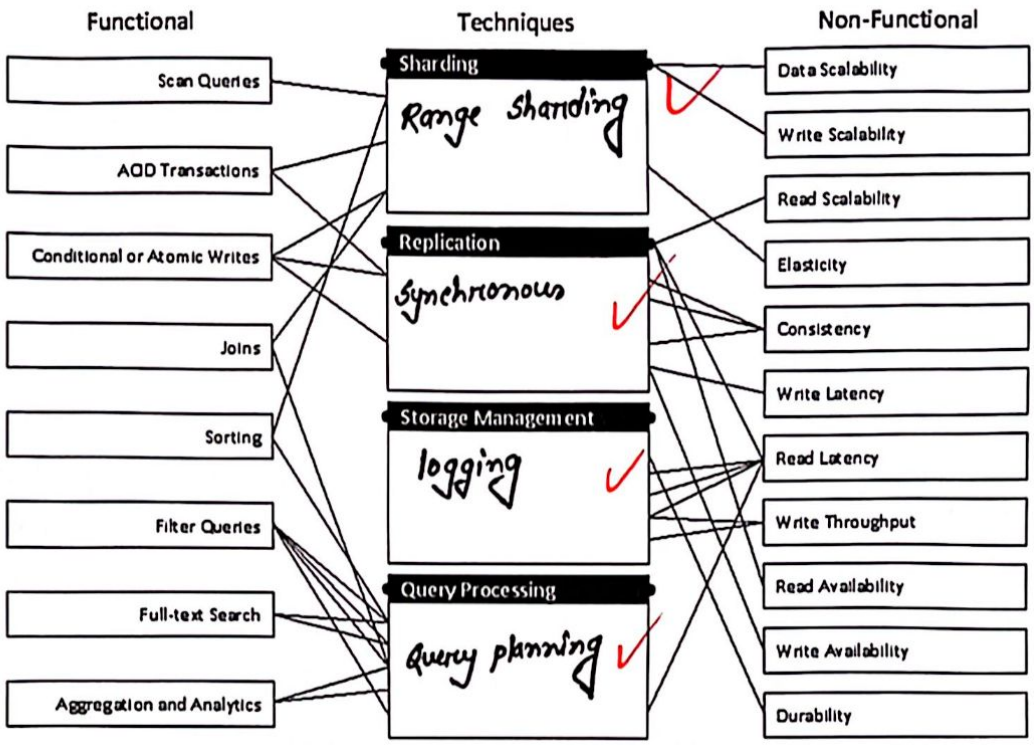


(d) (4 points) If you would need to write the same query in SQL (assuming a semantical equivalent relational schema), how would this query look like?

```
SELECT  p.firstname, p.lastname, d.dep_doc.name
FROM    Person, department
WHERE   p.person id = d.dept-doc id
```

1. NoSQL Toolbox

The following figure (adapted from the paper *NoSQL database systems: a survey and decision guidance* by Felix Gessert et al.) shows the so-called NoSQL toolbox. However, the techniques that connect the functional and non-functional system properties are missing.



(a) (4 points) Provide one example for each technique and write them in the empty boxes of the figure above.

(b) (6 points) A clinic wants to build a data lake. In the process of requirements engineering, you start to map the different data formats, data types and requirements. You collect this information using a table for four different kinds of data. Please complete the table below with the missing classifications.

Data set	Code
Patient data (Names and Medication)	1
Medical Images	2
Payment / Transactions	3
Laboratory results	4

Data set code	1	2	3	4
Structured (S) or Unstructured (U)?	S	U	S	U
Availability: High (H) or Medium (M)?	H	M	H	M
Query Pattern: Ad-Hoc (A) or Complex Analytics? (C)	A	C	A	A/C
Consistency: Strong (S) or Eventual (E)?	S	E	S	E

2. Database Design Scenario

(a) (10 points) The owner of a car rental comes to a software company in which you started working not a long time ago. She asks you to create a database for her business. The task description is the following: The car rental rents out cars to its customers. Every car has a license plate, model, color and mileage. Every customer has a first name, last name and gender. It is also important for the car rental to know for how long the customer has his or her drivers license, from when until when the car is rented and how much she has to pay. One customer can rent only one car.

Your colleagues proposed the following database design:

Customer

f.n.	l.n.	gender	driving_exp
Matias	Johnson	m	13
Mark	Smith	m	16
Anna	Taylor	f	29

Car

licence_plate	model	mileage
db234e	Mercedes	2100
hn455j	BMW	123000
gh789e	BMW	34523

Rentals

licence_plate	f.n.	l.n.	model	price	from	till
db234e	Anna	Taylor	Mercedes	300	03.04.2020	16.04.2020
db234e	Mark	Smith	Mercedes	40	05.06.2020	10.06.2020
gh789j	Matias	Johnson	BMW	150	25.06.2020	02.07.2020

Prove your colleagues that the provided database design is not optimal. Use the database design criteria discussed during the lectures. Name at least 3 criteria and why are they violated.

3. **Hash-based distribution of data**

(a) (3 points) Your colleague Dave explains the benefits of hash-based data distribution as follows:

> Hash-based data distribution provides efficient data retrieval by creating a one-to-one mapping between keys and database locations, allowing direct access to data without searching. In addition, hash-based data distribution can handle complex queries involving joins, range-based access or sorting.

As often, he mixes up things and you need to correct this statement. Explain to him why.

Start your sentence with "Sorry Dave, but ..."

(b) (4 points) Given the following database table structure and data:

OrderID	CustomerName	Product	Quantity
1	John	Apples	20
2	Jane	Oranges	15
3	Mary	Bananas	10
4	John	Oranges	5
5	Jane	Apples	25
6	Mary	Bananas	30

Design a simple hash function that will distribute the data based on the 'CustomerName' field on three available buckets that uses the ASCII letter codes of the string. Discuss how your hash function ensures a uniform distribution of data.

(c) (3 points) Demonstrate how your hash function distributes the sample data tuples of the table above to the three buckets.

You can use the ASCII table below for your calculations:

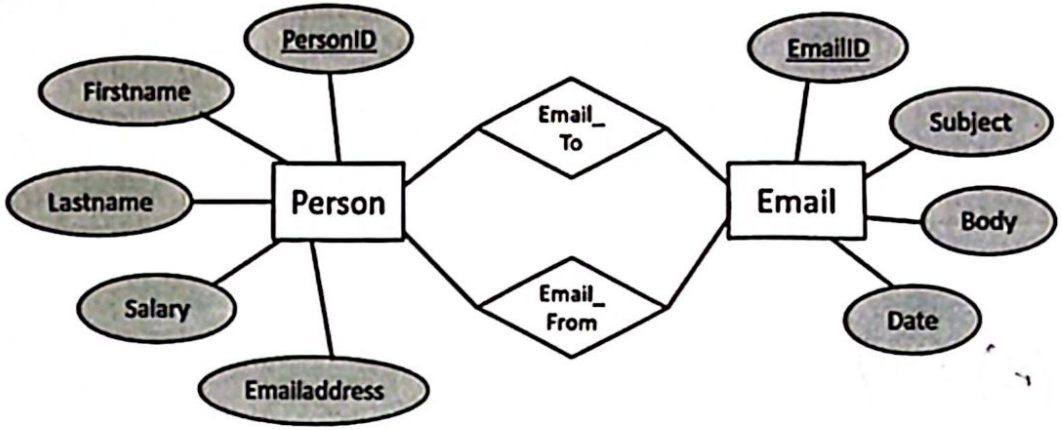
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	_	Y	Z
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	_	89	99

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	_	y	z
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113		121	122

Calculations:

4. **Query Language.**

Consider that we have designed a conceptual schema for a company called "Eron". It has two entities: Email and Person with the properties shown in the ER schema below.



For a case study, the same schema has been implemented in different database management systems. Therefore, we use different query languages to execute queries on the data. Consider the query below for the following questions:

```
db. Person . aggregate ([ { $lookup : {
        from : "Email",
        localField : "Emailaddress",
        foreignField : "Email_From",
        as: "emails"
        } }, { $unwind : "$emails"},
    {  $project: {"Firstname":1, "emails.Body":1 }} ])
```

(a) (1 point) What is the name of the query language?

6

(b) (1 point) In which database management system can it be executed?

(c) (1 point) Does this query return the "Emailaddress" of the sender of the emails?

(d) (2 points) Describe shortly the result of the query.

In another database system, the following query was used:

```
MATCH (n:Email) - [:EMAIL_TO] - (p:Person{firstname: "Joe"})
WHERE n.email_date > "2001-09-01 00:00:00+0000"
and p.salary>1000 RETURN n,p
```

(e) (1 point) What is the name of this query language?

(1) cypher Query

(f) (1 point) In which database management system can it be executed?

(1) Graph Database

(g) (3 points) Would this query return all the emails sent by Joe after the date "2001-09-01 00:00:00+0000"? If not, what should be changed to return the desired result?

5. Extensible Record Stores

(a) (1 point) What is the main role of a memtable in Apache Cassandra?

- ○ Storing data on disk
- ● Reading data from sorted data files
- ○ Serving as a cache for recently accessed data
- ○ Temporarily storing write operations in memory

(b) (1 point) What triggers a flush operation, moving data from a memtable to the sorted data file?

- ○ When a data record is updated
- ○ When a data record is deleted
- ○ When a read operation is performed
- ○ When the memtable reaches a certain size limit

(c) (1 point) What happens to the data in the memtable when it is flushed to the sorted data file (SDF)?

- ○ The data is deleted from the memtable and moved to the SDF
- ○ The data is duplicated in the SDF and stays in the memtable
- ○ The data is moved to the SDF and the memtable size is reduced
- ○ The data is encrypted in the SDF and the original data stays in the memtable

(d) (3 points) Imagine you are a database engineer working on a web application that uses Apache Cassandra as its database. The application is a simple blogging platform where each record is a blog post. Each blog post has a unique identifier, the author's user ID, the post title, the post content, and a timestamp indicating when it was created.

Describe how Apache Cassandra handles write operations when a new blog post is created. In your answer, specify the roles of the memtable and SSTables.

(e) (3 points) Your team has noticed that under heavy write load, the application's write latency increases. Suspecting that this could be related to memtable flushes, your team decides to modify the memtable cleanup threshold. Explain how changing this threshold could affect the application's performance, and discuss any potential risks associated with this change.

(f) (1 point) Tick all new requirements that are supported by Extensible record stores.

- ○ Data structure complexity
- ○ Schema independence
- ○ Sparseness
- ○ Self-descriptiveness
- ☑ Scalability
- ○ Variability

6. **General Knowledge**

(a) (4 points) During the requirements analysis, customer interviews give you different statements about what they expect from the system. Please identify and name which property of the database system it refers to and fill the table below. Please use the term we introduced in the lecture.

Customer Statement	Database Property / Support for
"Is there a way to make sure my database won't crash or become sluggish as more users use my application?"	[redacted]
"My data has a lot of blank spaces, and I need a database that won't store them all as it's taking up too much storage."	[redacted]
"How can I ensure my database won't become a bottleneck for my application's performance as it gets more popular?"	[redacted]
"What happens if my data exceeds the database's capacity? Is there a way to prevent that?"	[redacted]
"I'm concerned about future changes to my data schema. Can the database adapt easily?"	[redacted] adding data
"I'm looking for a database that includes metadata to describe the meaning of the data stored."	[redacted]
"I'm looking for a database that can handle data with changing formats or attributes."	[redacted]
"Is there a way to make sure my data is protected from any failures or errors in the database?"	[redacted]

(b) (2 points) Explain the differences between the consistency definition in CAP and in ACID.

- ACID (Availability, Consistency, Isolation, Durability)

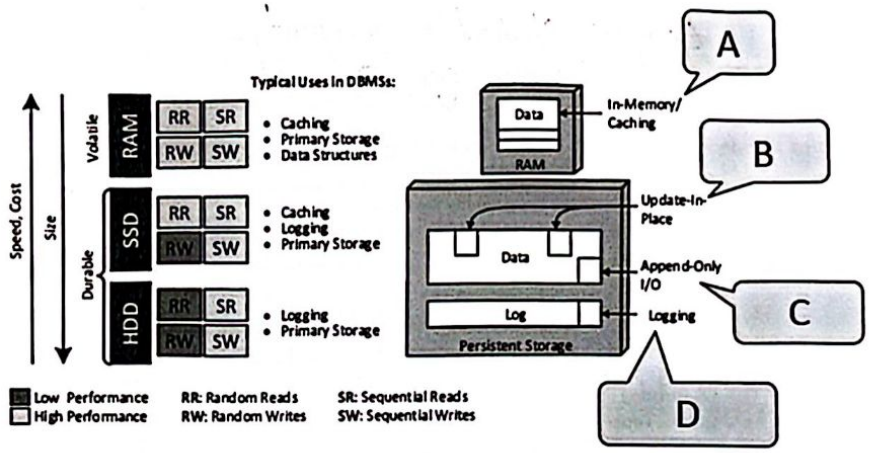
(b) (2 points) Explain the differences between the consistency definition in CAP and in ACID.

- ACID (Availability, Consistency, Isolation, Durability)
- CAP (Consistency, availability, Partition tolerance)

The basic differences are durability, Isolation & in partition tolerance.

2

(c) (4 points) The following figure (from a talk of Wolfram Wingerath) gives an architectural overview on typical uses in database management systems. Techniques like in-memory data management or caching, Update-in-place, Append-Only-I/O, and Logging are typically combined to improve the overall performance. To highlight the specific benefits of these four techniques, we'd like to add short statements.



Which statements could be added to this figure to summarize the benefits of these specific techniques?

Statement A (for In-Memory/Caching):

High performance ✓

Statement B (for Update-in-Place):

Sequential writes

1

7. **Graph Databases**

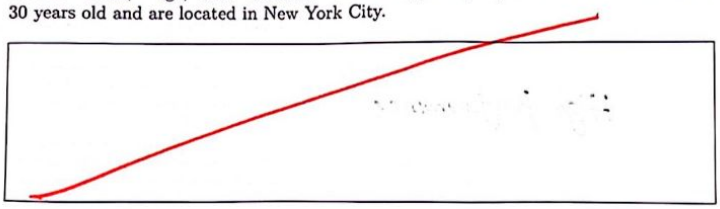
(a) (3 points) Define a property graph and describe its main components.

6

property Graph: A graph who holds the property. of a

Example: student's property name, age, ID

(b) (3 points) Suppose you have a property graph representing a social network, where nodes represent users and edges represent friendships. Each user node has properties such as "name," "age," and "location." Write a Cypher query to find all users who are 30 years old and are located in New York City.



(c) (4 points) Give two cases where graph databases outperform traditional relational databases and explain why.

3.5

Solution:

Graph databases are schemaless, so that is much more flexible, where the traditional database has schema, it's not flexible like Graph DB. ✓

Graph DB can process more complex queries than relational data bases. (✓)

INSTRUCTIONS

You are herewith instructed