1. What is a data model?
   Mathematical representation of data, Operations on data, Constraints

2. What is the relational data model?
   A relation is a table, *Relation schema* = relation name + attributes, in order (+ types of attributes), *Database* = collection of relations, *Database schema* = set of all relation schemas in the database

3. Describe the levels of the relational model!
   Logical level: The relations are considered as tables, The tables has unique names,The columns address the attributes, The rows represent the records, Rows can be interchanged, the order of rows is irrelevant
   Physical level: The relations are stored in a file structure

4. What are the operations of core relational algebra?
   Union, intersection, and difference; Selection: picking certain rows; Projection: picking certain columns; Products and joins: compositions of relations; Renaming of relations and attributes.

5. What is an expression tree?
   Leaves are operands - either variables standing for relations or particular, constant relations; Interior nodes are operators, applied to their child or children

6. Describe the monotonity of relational algebra
   Monotone non-decreasing expression: applied on more tuples, the result contains more tuples,Formally if Ri・ Si for every i=1,...,n, then E(R1,...,Rn)・ E(S1,...,Sn)
   Difference is the only core expression which is not monotone

7. Describe the core relational algebra operations on bags in a few words
   A *bag* is like a set, but an element may appear more than once; Bags also resemble lists, but order in a bag is unimportant

8. What are the operations introduced by extended relational algebra?
   Selection applies to each tuple, so its effect on bags is like its effect on

sets; Projection also applies to each tuple, but as a bag operator, we do not eliminate duplicates; Products and joins are done on each pair of tuples, so duplicates in bags have no effect on how we operate

9. Describe the 2 anomalies that can be avoided with relational schema design?

*Update anomaly* : one occurrence of a fact is changed, but not all occurrences;*Deletion anomaly* : valid fact is lost when a tuple is deleted

10. What is a functional dependency?

*X ->Y* is an assertion about a relation *R* that whenever two tuples of *R* agree on all the attributes of *X*, then they must also agree on all attributes in set *Y*

11. What are entity, entity set, attribute in Entity-Relationship models?
    Entity = "thing" or object, Entity set = collection of similar entities, Attribute = property of (the entities of) an entity set
12. What is a relationship in the Entity-Relationship model?
    A relationship connects two or more entity sets. In a many-many relationship, an entity of either set can be connected to many entities of the other set. many-one: each entity of the first set is connected to at most one entity of the second set. one-one relationship, each entity of either entity set is related to at most one entity of the other set
13. When do we call an entity set weak?
    weak if in order to identify entities of E uniquely, we need to follow one or more many- one relationships from E and include the key of the related entities from the connected entity sets

14. What are the 3 design techniques we discussed?
    Avoid redundancy, Limit the use of weak entity sets, Don't use an entity set when an attribute will do

15. What is the Boyce-Codd Normal Form (BCNF)?

We say a relation *R* is in *BCNF* if whenever *X ->Y* is a nontrivial FD that holds in *R*, *X* is a superkey. *nontrivial* means *Y* is not contained in *X.* *superkey* isanysupersetof a key (not necessarily a proper superset)

16. What is the 3rd Normal Form (3NF)? What is a prime?

(3NF) modifies the BCNF condition so we do not have to decompose in this problem situation. An attribute is *prime* if it is a member of any key

17. Describe how the optimizer work!
    Check syntax + semantics,Generate plan description, Transform plan into “executable”, Execute the plan

18. Describe Cost vs Rule based optimizers!

Rule: Hardcoded heuristic rules determine plan; Cost: Statistics of data play role in plan determination. Best throughput mode: retrieve **all rows** asap, Best response mode: retrieve **first row** asap .

19. Describe data storage in Oracle!
    Oracle stores all data inside datafiles(Location & size determined by DBA, Logically grouped in tablespaces, Each file is identified by a relative file number (fno)) ; Datafile consists of data-blocks; Data-blocks contain rows

20. Describe balanced trees!
    Indexed column(s) sorted and stored separately( NULL values are excluded (not added to the index), Pointer structure enables logarithmic search(Access index first, find pointer to table, then access table.

21. Describe B-trees!
    B-trees consist of: Node blocks (Contain pointers to other node, or leaf blocks), Leaf blocks (Contain actual indexed values,Contain rowids (pointer to rows)) Also stored in blocks in datafiles

22. What is OLTP (On-Line Transaction Processing)?

Short, simple, frequent queries and/or modifications, each involving a small number of tuples

23. What is OLAP(On-Line Analytic Processing)?

Few, but complex queries --- may run for hours.Queries do not depend on having an absolutely up-to-date database. Sometimes called Data Mining.

24. What is the star schema? What it is consits of?

A star schema is a common organization for data at a warehouse. It consists of:act table : a very large accumulation of facts such as sales;Dimension tables : smaller, generally static information about the entities involved in the facts

25. Describe the 2 ROLAP techniques we discussed!
    Bitmap indexes : For each key value of a dimension table (e.g., each beer for relation Beers) create a bit-vector telling which tuples of the fact table have that value.
    Materialized views : Store the answers to several useful queries (views) in the warehouse itself.

26. Describe how MongoDB replica set works!
    1 Primary, 2-48 Secondaries, Copies from original data,Operations affecting multiple rows replicate row by row. Client always connects on primary, Writing always goes on primary, Reading can go on secondary

27. What is sharding? Describe two possibilities for sharding in a few words! Collections distributed in cluster based on shard key, shard key = indexed data, Inside shard can have replicates, Lowering the load on servers (Querying on multiple machines Fewer data on a single machine). Range based sharding(shard key storing based on domains); Hash based sharding (distributing data based on given hash function)

28. Describe the 4 levels of MongoDB „write concern” in a few words! Unacknowledged(Client does not get feedback of writing), Acknowledged(Client detects errors), Journaled(Mongod does not acknowledges, until it does not write into the journal), Replica Set Acknowledged(Can be set, how many replicas have to confirm the writing before it is noted to the client)

29. Describe how MongoDB stores the data!

Every MongoDB instance includes: Namespace file, Journal file, Data file. Data file stores in extents: BSON documents, Indices, MongoDB metadata data. Extent: logical container

30. Graph databases: Describe the elements of a property graph!

Nodes – entities, Edges- relationships between the nodes, properties – attributes and metadata, labels – grouping.

31.Describe the 3 ways to store graphs in clusters!

"Black Hole" server, Chatting network, Minimal cut.

32. How do publish/subscribe processes work?

Publisher sends data to the topic, clients subscribe to topics, and if new data arrives in the topic the clients get it.

33. Describe the persistence solutions in Redis!

Snapshooting mode – binary dumps, in every x seconds or after y operations; Append Only File- every operation will be written out to a file, when restarting every operation runs again.

34. Wide column store: Describe the HBase architectural components!

HMaster- Master server,monitors all server regions. Region server- basic building element of HBase cluster, responsible for handling, managing,executing, reading, writing HBase operations. Zookeeper- coordinator, clients communicate with region servers via zookeper.

35. Describe the HBase read mechanism!

There is a special HBase Catalog table called the META table, which holds the location of the regions in the cluster. Here is what happens the first time a client reads or writes data to HBase Client Meta Cache Region Server Region Server DataNode DataNode .The client will query the .META server to get the region server corresponding to the row key it wants to access The client caches this information along with the META table location Meta table location Request for Region Server Get region server for row key from meta table .META location is stored in ZooKeeper.

## 36. Describe the HBase write mechanism!

There is a special HBase Catalog table called the META table, which holds the location of the regions in the cluster. Here is what happens the first time a client reads or writes data to HBase Client Meta Cache Region Server Region Server DataNode DataNode .The client will query the .META server to get the region server corresponding to the row key it wants to access The client caches this information along with the META table location Meta table location Request for Region Server Get region server for row key from meta table .META location is stored in ZooKeeper.

## 37. XML Data: What are the properties of a well formatted XML?

Has corresponding end tag for all of its start tags (closing pair), nesting of elements within each other must be proper, two attributes in each element must not have the same value and attributes must be given in quotes, markup characters must be properly specified, document can contain only one root element.

## 38. When do we say an XML document is valid for a given DTD?

If the document fits on the regular expressions. Types of the attributes are correct. Used identifier and references are correct

## 39. List 4 of the XML building elements!

To, from, heading, body

## 40. List 4 of the possible XML nodes!

Document, element, attribute, text, instruction, comment, namespace.

## 41. Describe the node order in the XML trees!

A parent node precedes its children and attributes, among the sibling nodes attributes come first, then other types; order of the attributes depends on implementation.

## 42. XQuery: Describe FLOWR expressions in a few words!

Built up by iteration, defining variables, ordering result, using predicates, constructing result (for, let, order, where, return)

## 43. List 3 XQUERY functions with examples (pick any 3)

count count((0,4,2)) → 3

max max((0,4,2)) → 4

subsequence subsequence((1,3,5,7),2,3) → (3,5,7)

empty empty((0,4,2)) → false()

exists exists((0,4,2)) → true()

distinct-values distinct-values((4,4,2,4)) → (4,2)

to (1 to 10)[. mod 2 eq 1] → (1,3,5,7,9)

## 44. Semantic web: What is RDF used for? What are the parts of RDF triples?

Connection between documents

• Contains triples: – –<subject, predicate, object>

• RDFS extends RDF with basic "ontology vocabulary": – Class, Property – Type,subClassOf – domain, range

## 45. What are the 4 RDF formats we discussed?

XML, N-Triples, Turtle, N3(Notation3)

## 46. Describe the 4 types of ontologies!

Top level, domain, task, application ontologies.

## 47. What is RDFS? What is OWL compared to RDFS?

RDF Schema describes classes and properties with the hierarcic connection between them. • OWL is a richer language which describes features between classes, properties.

## 48. SPARQL: Describe 4 SPARQL modifiers!

Filter – Optional – Limit – Order by

- Filtering the results - Optional information - Number of rows in result - Ordering result data – Distinct - Eliminating duplicates – Offset - Skip a specified number of solutions

49. Describe 4 SPARQL query forms!

SELECT, ASK, CONSTRUCT, DESCRIBE

50. Distributed databases: Describe Distributed DBMS and Multi DBMS architectures!

Distributed DBMS :Data layout can be different on the different machines, Local Inner Schema (LIS) is needed. • A single, global conceptual schema is needed (GCS), this contains the logical structure of all data type. • To handle the fragmentation and replication of data, Local Conceptual Schemas (LCS) are also needed, containing the the more abstract, logical structure of the locally stored data types. The union of these is the GCS. • Users can reach the database system through the External Schemas (ES). • Global queries are compiled into multiple local queries by the DBMS, which are executed at the right place.

Multi DBMS : • Fully autonomous • They do not know how to work together • Perhaps they do not know about the existence of each other • Perhaps they do not know how to communicate with each other • GCS only describes those databases that the local DBMSs want to share. The shared data are described in the local export schema (LES) in each DBMSs

51. Describe Single Server and Multiple Server architectures!

Single Sever:More efficient division of labor • Horizontal and vertical scaling of resources • Better price/performance on client machines • Ability to use familiar tools on client machines • Client access to remote data (via standards) • Full DBMS functionality provided to client workstations • Overall better system price/performance.

Multiple Server: Every single client handles itself the connection to the corresponding server or servers. Code on server side is simpler but it leads to fat clients. • A client only has to communicate with its ‚own’ server, then further communications happens between the servers. This case leads to thin clients and servers execute most of the data handling. directory • caching • query decomposition • commit protocols

52. Describe the CAP theorem and each part of it in a few words!

Data Models Relational (Comparison) Key-value Column-oriented Tabular Document oriented CA CP AP Pick Availability Consistency Partition Tolerance Each client can always read and write RDBMSs MySQL Postgres Aster Data Greenplum Vertica Dynamo Voldemort Tokyo Cabinet KAI Cassandra SampleDB CouchDB Riak All client always have the same view of the data Big Table Hypertable HBase MongoDB Terrastore Scalaris Berkeley DB MemcacheDB Redis The system works well despite physical network partitions

## 53. How do we decide if a fragmantation is correct?

Completeness • Decomposition of relation R into fragments R1 , R2 , ..., Rn is complete if and only if each data item in R can also be found in some Ri •

Reconstruction • If relation R is decomposed into fragments R1 , R2 , ..., Rn , then there should exist some relational operator ∇such that $R = \nabla 1 \leq i \leq nRi$ •

Disjointness • If relation R is decomposed into fragments R1 , R2 , ..., Rn , and data item di is in Rj , then di should not be in any other fragment Rk ($k \neq j$ ).

## 54. Describe the levels of replication!

1 Primary

2-48 Secondaries

Copies from original data

Operations affecting multiple rows replicate row by row

Client always connects on primary

Writing always goes on primary

Reading can go on secondary