Database Technologies
Rapid Fire Question

Q1 What is normalization? What is its need? Explain 1NF, 2NF, 3NF & BCNF in detail.

→ Normalisation is the concept of table design : table, structure, data types, width, constraints, relation.

Needs :-

- ⊙ Efficient table structure
- ⊙ Avoid unneccessary data duplication
- ⊙ Reduce problems of insert, update & delete.

A table with primary key having proper datatypes & widths to column containing all data is in un-normalized form. (UNF)

1NF → [Remove repeating group into new table, make key elements as PK & add keys to main table as composite PK] Repeat steps.
One to many

2NF → [Columns that are independant on composite PK are to be removed into new table & ~~co~~ key columns on which non-key were dependant is to be added to new table & made PK there] Repeat to sepe. non-key
Many to many.

3NF → [Non-key elements are examined. Inter-dependant column not directly related to PK are removed. PK of new table are retained to main table]
Repeat to seperate non-key into new table if unrelat

BCNF → Seperate table to be created to ensure data consistency so user will have to input less data.

Q2 Explain entity relationship diagram & explain all types of relationships with examples.

→ To avoid redundancy of the data, data should be organised into multiple tables so that they are related to each other

Entity relations is outcome of normalization process.

□ depts	□ emeet	□ meet.	□ address.	□ emp
⊙ deptno	⊙ meetno	⊙ meetno.	⊙ empno	⊙ empno ⊙ mgr
⊙ dname	⊙ empno	⊙ topic	⊙ dist	⊙ ename ⊙ deptno.

* Types of relationships & examples :-

1] one to one → meet & emeet.

2] one to many → emp to all

3] many to one → all to emp

4] many to many → emp, meet, emeet.

Q3 Which are different types of joins in SQL? Give examples.

deptno	dname.
10	DEV
20	QA
30	OPS
40	ACC

empno	ename	deptno
1	Amit	10
2	Rahul	10
3	Nilesh	20
4	Nitin	50
5	Sarang	50

<u>Cross join</u>

⊙ Will return all possible combinations.

⊙ Each emp will be repeated 4 times. once for each dept.

<u>Inner join</u>

Amit	DEV
Rahul	DEV
Nilesh	QA

⊙ Only matching records.

<u>Left join</u> & <u>Right join</u> (depends on syntax)

Amit	DEV
Rahul	DEV
Nilesh	QA
Nitin	NULL
Sarang.	NULL.

⊙ Display matching rows & all remainig records from one table respectively (Left or right)

<u>Full outer joins</u>

All records from both tables.

Q4 What is stored procedured? What are advantages of stored procedure?

→ Stored procedure is a routine. that contains multiple SQL statements along with programing contructs. It can take any or no no. of arguments & returns no value.

Advantages :-

1) Better performance
2) Higher productivity
3) Easy to use
4) Scalability
5] Maintainability
6] Security.

Q5 How stored function is different than stored procedure?

→

Stored function	Stored procedure
① Must return a value	① Returning value is optional
② Only has input parameter	② can have both i/p & o/p parar
③ function can be called from procedure	③ Procedure cannot be called from function.
④ Only allows DQL	④ Allows DML & DQL.
⑤ Can be used in SELECT/ WHERE/HAVING	⑤ Cannot be used in SELECT WHERE/HAVING
⑥ Cannot use transaction	⑥ Can use transaction
⑦ Cannot use try-catch	⑦ Can use try-catch.

Q6 What is difference between CHAR, VARCHAR & TEXT? Does data type matter while creating index?

→

CHAR	VARCHAR	TEXT
① Fixed inline Storage.	Variable inline Storage	Variable external storag
② For smaller, rest of space is unused	Stores length & characters.	Stores entire text.
③ Fastest access	slower than CHAR.	Very slow access.

TEXT is not ideal for indexing.

Q7 Explain ACID properties of transaction. What is use of transaction?.

→ A- Atomicity – All queries are executed as a single unit. If any query is failed, other queries are discarded.

C- Consistency – When transaction is completed, all clients see the same data.

I - Isolation – Multiple transactions are processed concurrently.

D- Durable – When transaction is completed, all data is saved on disk.

Transaction is a set of DML statements. If any power failure occurs, system or network failure automatically rollback current state. They are isolated from each other & are consistent.

Q8 What is table level locking & row level locking? What is optimistic locking & pessimistic locking?

→ When an user update or delete a row within a transaction, if the table has an index, it will lock the row that is being modified. This is called as row locking.

When a user update or delete a row within a transaction, if the table doesn't have an index, it will lock the entire table. This is called as table locking.

When the locks are automatically released when commit or rollback is done, the process is called as optimistic locking. (locks are auto-enabled during transaction)

When rows are manually locked in advance before using update or delete, the process is called as pessimistic locking.

Q9 Write a query to find third highest salary of employee using subquery & without using subquery.

→

```
SELECT DISTINCT salary FROM emp ORDER BY salary DESC LIMIT 2,1
```

```
SELECT MAX(salary) FROM emp WHERE salary <
(SELECT MAX(salary) FROM emp WHERE salary NOT IN
(SELECT MAX(salary) FROM emp));
```

Q10 Write a query to print name of employee & his manager. Also print managers manager.

→

```
SELECT  e.EMPID , e.FIRSTNAME , m.FIRSTNAME,
        mm.FIRSTNA~~A~~ME  FROM  emp e
INNER JOIN emp m ON e.MGRID = m.EMPID
INNER JOIN emp mm ON m.MGRID = mm.EMPID ;
```

Q11 What is use of views? How to limit DML operations on views to the given criteria.

→ Views are used to provide a restricted view of the data from one or more tables. by joins or subqueries.

To limit DML operations on view, view can be created with CHECK OPTION.

Q12 What is normalization? Explain 1-NF, 2-NF, 3-NF & BCNF in details.

→ Answer is explained in the solution of question 1.

Q13 What is NoSQL database? What are its advantages & disadvantages?

→ NoSQL databases are non-tabular databases & store data differently than relational tables. They provide flexible schemas & scale easily with large amounts of data & high user loads.

Advantages :-

⊙ Handle large volumes of data at high speed with scale-out architecture.

⊙ Stores unstructured, semistructered & structured data.

⊙ Enables easy updates to schemas & fields.

⊙ Developer friendly.

Limitations :-

⊙ Frequent lack of data consistency.

⊙ Comparative lack of deep analytics support.

⊙ Lack of high availability features for private & hybrid clouds

Q14 Explain BASE transactions & CAP theorem.

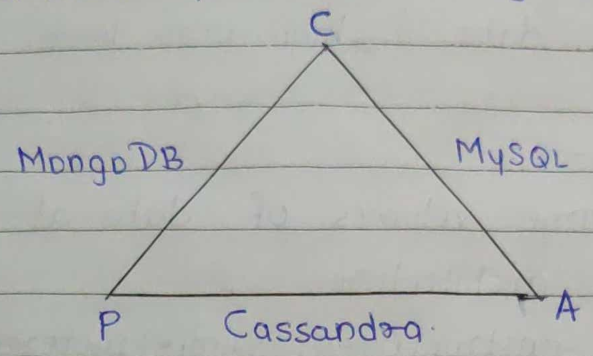
→ In a distributed environment network partitions among the member nodes is given. The CAP theorem states that in the event of a network partition, the system can either be available or consistent.

CAP :-

Consistency - It refers to the fact that all replicas having particular record must return exact same value. (Not neccessarily a physical guarantee, data systems can chooese to provide logical guarantees too. for e.g. quorum response

Availability : All the active nodes at any moment must be able to respond to diff. operations.

Partition-tolerance : The system must be able to tolerate network partitions among its participant nodes.



BASE :-

Basically-Available : A distributed system should be available to respond with some acknowledgement - even if it's a failure message, to any incoming request.

Soft-state : The system may keep changing states as & when it receives new information.

Eventually-consistent : The components in the system may not reflect the same value of record at a given point in time. They will settle it eventually though.

Q15 Explain types of NoSQL databases. Where they can be used?

→ ① Document databases -

A document database stores data in JSON, BSON, or XML documents. In document database, documents can be nested. Elements can be indexed.

Uses- ecommerce platforms, trading platforms & mobile app development accross industries.

② Key-value stores -

This the simplest type of NoSQL databases. Every data element in the database is stored as a key value pair consisting of an attribute name & a value. In a sense, it's like a RD with only two columns: key & value. (state & Maharastra)

Uses - shopping carts, user preferences & user profiles

③ Column-oriented databases -

While a relational database stores data in rows & reads data row by row, a column store is organised as a set of columns. When ronning analytics on few columns, they can be read without consuming memory with unwanted data. Columns are often of same type & benefit from more efficient compression, making reads even faster.

Uses - analytics.

④ Graph databases.

A graph database focuses on relationship between data elements. Each element is stored as a node. The connections between elements are called relationships or links. In graph database, connections are the first class elements of the database, stored directly. They are usually used with traditional RDBMS since very few realworld business system can survive solely on graph database.

A graph database is optimized to capture & search the connections between data elements, overcoming the overhead associated ~~win~~ with JOIN in SQL.

Uses - fraud detection, social network & knowledge graphs

Q16 Write queries to perform CRUD operations on MongoDB.

→

① CREATE → () contents

db.collection.insertOne () } field (value)
db.collection.insert Many ()

② Read

db. collection. find () → query criteria, projection

③ Update

db.collection.updateOne() } update filter
db.collection. updateMany() update action
db.collection. replaceOne ()

④ Delete

db.collection.deleteOne() } delete filter.
db.collection. deleteMany()