Interview notes

by Keshav Sharma

- Database
- DBMS
- OOPs
- OS
- CN
- S/W Eng.

With support from

Adobe

nasscom foundation

BRIDGING CREATIVITY TO CAREER

Skilling Pathway to Success

Onground Implementation Partner

etrainIndia
empowering education

26-01-2026
Monday

Database (DBMS) interview Questions

Basic (9)
Q2, Q3, Q4, Q5, Q7, Q12, Q6, Q11, Q14

Intermidiate (5)
Q1, Q6, Q2, Q5, Q12

{A} BASIC

(Q2) Primary Key & Foreign Key.
(PK) (FK)

→ (PK) unique record in table.
→ Data integrity. (unique)
→ (FK) next to PK of new table.
→ FK is the old PK of previous table.
→ Referential integrity. (consistent data/valid)

(Q3) CRUD operations

→ Create, read, update, delete.

Data = "Keshav", 20
Record => name = "keshav"
no = 20

→ Create :- Use "INSERT INTO" to add record in a table.

→ read :- Use "SELECT" to retrieve data.

→ Update :→ Use "UPDATE" statement to update.

→ Delete :→ Use "DELETE FROM" statement. to delete.

Q4) Types of joins & work

Inner Join :- Retrieve record, matching value in both tables.

Left outer join :- All left record match right (table 2) (table 1).

Right outer join :- All right record match left.

Full outer join :- Matching & non matching both are retrieved. Unmatched are NULL.

Q5) Ensure data integrity in Relational database.

→ 1) Primary Keys (PK)
2) Foreign Keys (FK)
3) Unique Constraints :- Column value are different.
4) Not Null constraints :- Prevent empty field.
5) Check constraints :- Follow defined rules (no invalid inputs)
6) Transactions :- Either full execute or not at all.

Q7) ACID properties & imp.

↳ Ensures :- 1) reliable transaction.
2) Data reliability. (trust)
3) integrity. (accurate)

Atomicity :-
Consistency :-
Isolation :-
Durability :-
even for

Q12) Data

→ Proce
inte
→ Be

Q6) Di

→ Onlin
→ Shor
(ins
→ Eg

Atomicity :- Full transaction or abort.

Consistency :- transaction keeps database valid.

Isolation :- Transaction is not interfiered.

Durability :- Ensure permanent storage of transaction result in database even for failure.

Q12) Data Normalization

→ Process of minimizing redundancy & maintain integrity.

→ Benefits
- ↝ Reduces duplicates
- ↝ maintain consistancy.
- ↝ Easier to view & maintain & update.
- ↝ Eliminate unnecessary data.

Q6) Differentiate OLTP & OLAP databases.

OLTP	OLAP
→ Online transaction Processing.	→ Online analytical processing.
→ Short online transaction, (insert, update, delete)	→ read heavy operations on large operations of data.
→ Eg :- Retail sales System.	Eg :- Data warehousing.

Q11) What is Relational Database. How differ from NoSQL database.

Relational	NoSQL
→ Structured table	→ Unstructured table.
→ ACID Property follow.	→ Does not provide ACID.
→ Best for complex Queries & transaction	→ Best for storing large unorganised data for web applications.
→ Predefined schemas & relationships.	

Q14) SQL function for data aggregation.

aggregation → one column result.

→ SUM(): Total sum of numeric column.

→ AVG(): Total average.

→ COUNT(): Total number of row

→ MIN(): Minimum value of column.

~~→ MAX()~~

→ MAX(): Maximum value in column.

{B} Inter

Q1) Explain impor

→ Transact
Oper
unit

→ It foll

→ impo
in Bankin
occur

*

Q2) Opti

1) Indexi

2) Avoidi

3) Quer
into

4) Anal
-an

5) Da

6) A

{B} Intermidiate 1, 2, 5, 6, 12

Q1) Explain database transactions & importance in application development.

→ Transaction means a sequence of operations performed as a single unit of work.
→ It follows ACID properties.
→ Importance :- Maintain data consistency, in banking transaction either transaction occur fully or revokes.

*

Q2) Optimize database Queries for performance.

1) Indexing :- Creating index on frequently used columns.
2) Avoiding Select :- only select the column u need.
3) Query refactoring :- Rewriting complex commands into simple & faster one.
4) Analysing execution plan :- Use tools to understand why the Query is slow.
5) Database Configuration :- Maintain settings & database properly.
6) Archive old data :- Remove old data (unused)

Q5) Handle Concurrent data access & prevent deadlocks?

→ Concurrent data access means when many users read, write on database at the same time.

→ deadlock mean when both processes are waiting for each other to execute.

Handling & preventing

→ Locking Mechanism → if one run, other must wait.

→ Transaction isolation level
→ database seperates users work.

→ deadlock detection → check if deadlock occur, kill one transaction.

→ Short Transaction → pay quickly not waiting 10min (unsafe)

→ Ordering access
→ lock A first then B.
prevent both fight in opposite direction.

PTO →

Q6) Databa
Query
→ databa
speci
colu
→ impo
1) Spe
2) im
3)
we
d

Q12)
→ W
related
Appl
→ R
→ R
→ M
→ LA

Q6) Database indexing & importance in query performance.

→ database indexing means creating a special shortcut from table column, for faster data retrieval.

→ importance

1) Speed up Query :- directly jump to matching row.

2) improve sorting :- index help database arrange quickly during sorting.

3) Enhance join performance :- when we join two tables, index helps database match record quickly.

Q12) Window functions & Applications.

→ Window functions perform calculation on related rows but does not remove row.

Applications :-

→ Ranking :- Assign ranks to rows.

→ Running total :- calculate total till current row.

→ Moving average :- average of last few rows.

→ LAG and LEAD :- Compare previous row to next.

27-01-2026
Tuesday

DBMS Theory interview Questions

Q1-Q15, Q19, Q20, Q23, Q26, Q27

(Q1) Database Management system?

(Ans-1) database management system (DBMS)

→ A software.
→ data integrity, maintain consistency.
→ update, store, retrieve data.
→ Provide interface to interact with database.

(Q2) Advantage of using DBMS

(Ans-2) → data integrity & consistency.
→ Fast data retrieval ⇒ indexing.
query optimization.
→ security → controlled access permision grant.
→ Reduce redundancy (duplicates).
→ Backup & retrieval.
→ Concurrent access. (Many user access same time)

(Q3) DBMS & ~~RDBMS~~ RDBMS

↝ database management system (DBMS)
↝ Relational " " (RDBMS).

DBM

→ Store ~~as~~ (non-relation
→ store as
→ No integri
→ Eg:- Micros

(Q4) Differen

4 types

1. Hierarch
2. Network
3. Relationa
4. Object

(Q5) Re

↝ r

Rou

Colu

DBMS	RDBMS
→ Store ~~as files~~, (non-relation form)	→ Stored in relations in row's & column's.
→ store as files.	→ Store in table.
→ No integrity constraint.	→ Follow integrity with Primary & foreign key.
→ Eg:- Microsoft Access.	→ MySQL, Oracle.

(Q4) Different Types of DBMS.

4 types

1. Hierarchical DBMS → Tree like Structure eg:- IBM's IMS
2. Network DBMS → Graph (many to many) Eg:- IDS
3. Relational DBMS → Table Structure:- Eg- MySQL
4. object-oriented DBMS → Store data as object as in OOPs. Eg:- object DB.

(Q5) Relation in DBMS.

→ row's & columns make a relation in DBMS.

row :→ record

column :→ Attribute

(Q6) table in DBMS

→ collection of data stored in row & columns.
row → record / column → attribute.

(Q-7) row & column in DBMS

→ row (tuple) :- Single record & contain value for each column.
→ column (attribute) :- characteristic / attribute.

(Q8) Primary Components of DBMS.

1) database engine :- store data, retrieve & manage.
2) database schema :- define structure of database.
3) Query processing :- interpret & execute SQL queries.
4) Transaction Manager :- Ensure ACID properties.
5) Storage Manager :- Physical storage of database.

PTO →

(Q-9) Primary key with Example.

→ (PK) is a unique record for each table.
(cannot be NULL.)
→ Eg:- Roll number column in a table.

(Q-10) Foreign key with example.

→ attribute in a table that links to primary key to form relation b/w two tables.
→ Eg!- branch_name in course_name.

(Q11) Normalization & importance

→ Organizing data by dividing large table into smaller tables maintain relation & reducing redundancy.
→ importance:- (1) Eliminate duplicates.
(2) improves integrity (3) Prevent anomalies (mistakes) during insertion, deletion.

(Q12) Denormalization, difference from normalization

(Ans-12) Adds/combines tables at cost of redundancy for performance boost.

(Q13) Candidate Key in DBMS.

A set of one or more attributes that can identify a tuple, multiple candidates are selected and only one is chosen as primary key.

(Q14) SQL SELECT statement Usage.

SELECT is a statement to query data from tables. It also allows to retrieve specific columns.

(Q15) View in DBMS, How differ from a table.

→ A view does not store data instead present data from other tables.

19, 20, 23, 26, 27

(Q19) DELETE VS TRUNCATE in SQL

DELETE :- Removes row & can be rolled back.

TRUNCATE :- Permanently deletes the row, cannot be rolled back.

(Q20) Index in DB

Index is like a used for fast re

(Q23) Join & ty

Join is used to more table

Types → (Al

1. inner join
2. left join
3. Right join
4. Full join
5. Cross jo
6. Self

(Q20) Index in DBMS & use

Index is like a table of content in book used for fast retrieval of data from table.

(Q23) Joins & type of joins.

Join is used to combine row of two or more table with related column.

Types → (All capital).

① inner join:- Only matching row both table.

② left join:- row from left table match right.

③ Right join:- " right " left ".

④ Full join:- All row (matched or unmatched with NULL)

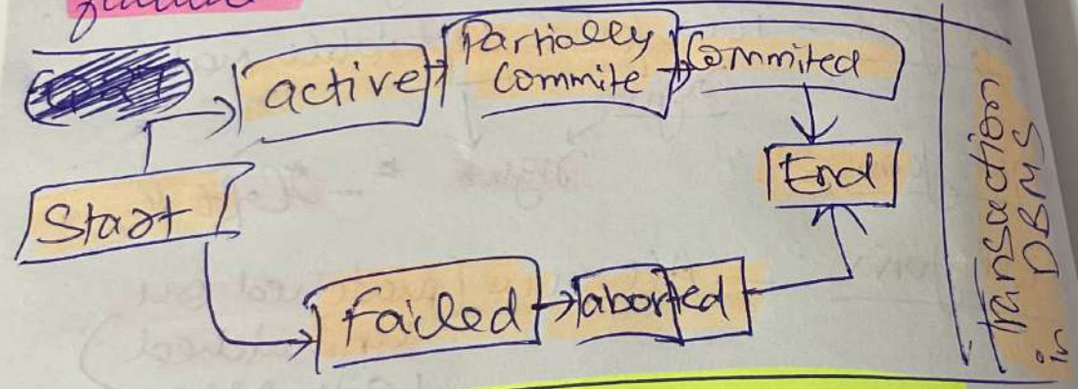
⑤ cross join:- Every row of one table to every row of other. (cartesian product).

⑥ Self join:- Compare every row & column within itself to find out relation / value.

PTO →

(Q26) Transaction in DBMS

Sequence of SQL operations executed as a ~~sequence~~ unit of single work.
→ maintain integrity
→ Maintain Consistency.
→ Provide Isolation.
Basically, assures database reach a valid state regardless of error & system failure.



(Q27) ACID properties in DBMS

Atomicity :- if one transaction fail, all fails.
Consistency :- Must follow all rules.
Isolation :- Independent Execution.
Durability :- changes are permanent even on failure.

OOPS THEORY
Interview Q

29-01-2026
Thursday

OOPS THEORY Interview Questions

Q1, Q2, Q7, Q8, Q9, Q10, Q11, Q12, Q13, Q14, Q5, Q23, Q24, Q25, Q30

Questions

(Q1) OOPS?

(Ans1) Object oriented programming
→ Software with bunch of objects talking to each other.
→ object (collection of data & methods)

(Q2) Why OOPs?

(Ans-2) ① Code looks like real world object, so developer and users understand better.
② Due to encapsulation we can change code without affecting how it's used.
③ Suitable for large companies as code is well organised.

(Q7) Class?

→ Blueprint used to create objects.
→ Contains :- ⓐ Variables.
ⓑ Methods.

(Q8) Object?

→ instance of a class.
→ Used to access data & method of class.
→ represents real world entity with data & methods.

(Q9) OOPs Features

→ Encapsulation :- Binding method & data together & protect data.

→ Abstraction :- Show required detail & hide internal detail.

→ Inheritance :- One class acquires property of others.

→ Polymorphism :- Same function behaves differently in different situations.

(Q10) Encapsulation

→ Wrapping data & methods together inside a class.

→ Protects data by allowing access only through methods. (data hide)

(Q11) Abstraction?

→ Shows only necessary details & hide any unnecessary details from the users.

→ Implemented using classes & variables.

(Q12) Inheritence & Purpose?

→ Child class derives property from parent class.

Imp → Code Reusability

→ Runtime Polymorphism (method overriding)

[Child class provide its own method already provided in parent class, and method call is decided at runtime].

Eg 1 - a = Dog()
a.sound()

(Q13) Polymorphism & Types

→ Same function or method works differently in different situations.

Types ① Runtime Polymorphism (method overriding)

② Compile time Polymorphism (method overloading)

(Q14) access specifier & significance in OOPs

→ Keywords that controls visibility & accessibility of class members (data & methods).

→ Types :-

ⓐ Public :- accessible from anywhere.
ⓑ Protected :- " " class & subclass.
ⓒ Private :- " " only inside class.

→ Helps achieve encapsulation & data hiding in OOPs.

(Q15) Overloading and Overriding

→ Overloading :- Same method name with different parameters in the same class.

→ Compile time (decided)

→ Overriding :- Same method name in parent & child class with different implementation.

→ Runtime (decided)

(Q23) Constr

→ A spec
object
created

→ Python

(Q24) Type

① Paramet

→ ~~###~~ W

→ def — in

② Non Pa

→ Const
(Ex

→ def

(Q25) dest

→ Calle

→ Used

Eg: class
- de

(Q23) Constructor ?

→ A special method used to initialise object data when object is created.

→ Python! - `__init__(self):`

(Q24) Types of constructors in Python

① Parameterized

→ We can input (argument)

→ `def __init__(self, name):`

② Non Parameterized

→ Constructor without parameter. (except (self).

→ `def __init__(self):`

(Q25) destructor in Python.

→ Called when object is 'deleted'.

→ Used to release resources (Files).

Eg:-
```
class Test:
    def __del__(self):
        del t    (t = Test())
```

(Q 30) Abstract class in Python

→ class that cannot be instantiated. (eg:- class Animal)
(a = Animal() X wrong, it's meant to used by other class like class DOG)

→ Used only for inheritance

→ forces child class to implement required methods.

→ Created as below

```
from abc import ABC, abstractmethod

class Animal(ABC):
    @abstractmethod
    def sound(self):
        pass

class DOG(Animal):
    def sound(self):
        print("Bark")
```

2-2-26
Monday

Operating System Interview Questions

Q1, 2, 3, 4, 10, 11, 12, 17, 18, 20, 25, 26, 43, 48, 75

(Q1) What is a process and process table?

(Ans 1) **Process**

→ Program currently running (in execution)
→ Eg:- Web browser
→ Operating System (OS) manage all running process.
→ OS gives CPU time to each process to run.
→ OS also provides memory & disk space.

Process Table

→ Keeps track of all running process.
→ Table contains entry for every process.
→ Store info. about resources used by process and current state of the process.

(Q2) Different states of process?

→ 3 States of Process

1) Running state

→ Process has all required requirements to execute.
→ OS has allowed it to use CPU.
→ One process run at a time.

2) Waiting state

→ Process is waiting for external event to happen.
→ Eg:- waiting for user input.

3) Ready State

→ Process has all resources except the CPU.
→ It's waiting for permission from OS to use the processor.

Note

In real ready and waiting state are managed using queues that store the process in these states.

(3) What is a

→ Single sequ
a process.
→ Threads a
process beca
properties
resources.
→ Thread im
by paral
same ti

Eg:- (1) In
tabs
(2) MS wo
→ One
→ Pro

(4) What
proce

(Q3) What is a thread?

→ Single sequence of execution inside a process.

→ Threads are called lightweight process because they share some properties of processes and few resources.

→ Thread improves application performance by parallelism (multiple task at same time).

Eg:- (1) In a web browser, different tabs can run as different threads.

(2) MS word uses two threads:
→ One for text formatting.
→ Processing user input.

(Q4) What are the different between process and thread?

PTO →

Aspect	Process	Thread
Execution unit	Independent program under execution.	Smallest unit of CPU execution inside process.
Memory usage	has separate memory space.	shares memory of the process.
Communication	use IPC (inter-process communication).	uses shared memory.
Context Switching	Slow/heavy	fast/lightweight

(Q10) What is Demand Paging and how it works??

→ Means a page is loaded into RAM only when it is needed, not before.

Working :

PTO →

(1) Process Start

→ Full page is not loaded into RAM.
→ Only required pages are loaded.

(2) Page table check

→ OS check page table if page in RAM

(3) Page fault Occur

→ If page not in RAM, Page fault occur.

(4) Page is loaded

→ OS bring page from Secondary memory to RAM.

(5) Page table Loaded

→ The page table updated with new location.

(6) Execution ~~Acti~~ Continues

→ The page continues from where it is stopped.

PTO →

(Q11) Kernel & use

→ Kernel core of OS.

use

→ Manage CPU & Memory
- Decide which process get CPU time.
- Control memory usage.

→ Connect Software to Hardware
- Bridge b/w application & hardware.
- Use system calls and inter-process communication (IPC)

(Q12) What are different Scheduling algorithm?

→ Scheduling algorithm means which process get CPU and when.

Types

(1) FCFS : First Come, First Served

→ Run in order they arrive.

(2) SJN: Shortest Job Next

→ Process with shortest execution time runs first.

(3) Priority Scheduling

→ Process with highest priority runs first.

(4) SRT: Shortest Remaining time

→ Process with least remaining time is executed next.

(5) Round Robin (RR)

→ Each process gets fixed time (time slice) in circular order.

(6) Multi-level Queue Scheduling

→ Process are divided into different Queues based on type.

PTO →

(Q17) Explain FCFS

→ First come first serve

→ Process which comes first gets CPU first.

→ Working

- Process are executed in same order they arrive in ready Queue.
- Once process gets CPU it keeps it until finishes or goes for I/O.

→ Disadvantage

- If long process comes all short process must wait.

(Q18) Round Robin (RR) Scheduling

→ Each process gets a fixed time in circular order.

Working

(1) Fixed time given

→ Each process is given small fixed time.

(2) Process not finished

→ If process not finished / interrupted move to end of queue.

(3) CPU moves to next process, repeating the cycle.

---

(20) Banker's Algorithm

→ Deadlock Avoidance Algorithm.

→ Checks giving resources to a process is safe or not before allocation.

→ Working

→ Each process tells maximum resource need.

→ OS simulate resource allocation before giving them.

→ If safe allocated else denied.

(Q25) Basic Function of Paging

→ Paging : Store a program in memory in non continuous block.

Working

→ Divide memory
- Hard disk → Pages
- RAM → Frame

→ Loads Program parts
- Process split into pages
- Pages are placed into free frame in ram.

Benefit

→ Better use of memory.
→ NO continuous memory space need.
→ Reduce Memory waste.

(Q26) How swapping improves memory management?

Swapping

Process from ram to disk and brig back later when needed.

Works (Use)

→ Blocked or idle process moved to disk.
→ Creates free space in RAM.
→ New process are loaded in RAM.
→ CPU is kept busy.

Benefit

→ Better Memory utilization.
→ Better CPU utilization.
→ Allow more process than RAM size.

(Q43) Preemptive VS Non-Preemptive Scheduling

Preemptive Scheduling	Non-Preemptive Scheduling
→ CPU given for short time.	→ CPU given until finished.
→ Process can be stopped.	→ Process cannot be stopped.

→ High priority can run immediately.
→ Low priority may starve.
→ More flexible.
→ Complex & costly

→ High priority must wait.
→ Short priority wait behind long ones.
→ Less flexible.
→ Simple & low cost.

(Q48) Context Switching

→ CPU stop one process and start other.

Working

→ Saved current state by CPU.
→ Stored in (PCB) Process control block.
→ CPU loads the saved state of next process.
→ Execution continues from where it was last stopped.

→ Why needed

→ Multiple process share CPU.

→ improved CPU utilization.

(Q75) Deadlock

→ Two or more process wait for each other to execute.

→ Process hold some resources.

→ They wait for other resources.

→ System get stuck

PTO →

4-2-26
Wednesday

Networking Interview Notes

by Keshav Sharma

Q3, 4, 5, 13, 14, 15, 20, 23, 24, 33, 36, 49, 50, 51, 54

(Q3) OSI Software / User support layers

→ (1) Application layer
→ Provides network to end-user applications.
→ closest layer to user.
→ enable web, email, file transfer.
→ interface b/w app & network.
→ Eg! - HTTP → web
FTP → File Transfer
SMTP → Email
DNS → Domain → IP

PTO →

(2) Presentation
→ Responsible
encryption
→ Converts
format
→ Handles
→ Handle

(3) Session
→ Respon
Managing
→ Sta
duri / aft
→ Pro

(Q4) Hos

(2) Presentation Layer

→ Responsible for data format, encryption and compression.

→ Converts data into common format.

→ Handles Encryption/decryption.

→ Handle Compression.

(3) Session Layer

→ Responsible for establishing, managing and terminating sessions.

→ Starts, maintains & ends session during/after communication.

→ Provides checkpoint/Recovery.

(Q4) Hardware/Network support layers in OSI Model.

PTO →

(1) Network layer

→ Responsible for Routing & addressing.
→ Finds best path b/w networks
→ Uses IP address.
→ works with routers.

(2) Data Link Layer

→ Responsible for node to node delivery and error detection.
→ Uses MAC address.
→ Detects error.
→ Data is sent as frames.

(3) Physical Layer

→ Responsible for transmitting raw bits over the medium.
→ Sends 0 and 1 as signals
→ Defines Cables, Connector, voltage.

(Q5) HTTPS Prot

→ HTTPS (Hyp

→ Secure versi
safe web co
→ Default po
→ Uses SSL
→ protects da
Authenti
→ URL Sta

(Q13) What is C
Availabil

→ (CIA)

(1) Confiden

→ Only
data

→ Prev
→ Eg:

(Q5) HTTPS Protocols

→ HTTPS (Hypertext transfer protocol secure)
→ Secure version of HTTP used for safe web communication.
→ Default port number :- 443
→ Uses SSL / TLS encryption.
→ protects data privacy, integrity, authentications.
→ URL starts with https://

(Q13) What is confidentiality, Integrity & availability? (CIA Triad)

→ (CIA) basic goal of cybersecurity.

(1) Confidentiality

→ Only authorised people access data.
→ Prevents unauthorised access.
→ Eg:- Passwords, encryption.

(2) Integrity

→ data should be accurate & not modified illegally.
→ Prevents unauthorised changes.
→ Uses hashing / digital signature.

(3) Availability

→ data should be available when needed.
→ System should be accessible.
→ DoS attack: threat

(Q14) VPN (Virtual Private Network)

→ Secure encrypted connection over the internet.
→ Extends private network over public internet
→ Uses encryption + tunneling
→ Allow remote access safely.

→ Hides Usen

(Q15) What is
Assymet

→ Symmetr
→ Same ke
decry
→ Fast &
→ Key s
→ Assymm
→ Publ
Pri
→ Mo

(Q20) IP

→ Firew
→ Filt
tra
→ First
→ All

→ Hides User IP Address.

(Q15) What is Symmetric and Assymetric encryption?

→ Symmetric Encryption

→ Same key for encryption & decryption.

→ Fast & simple

→ Key sharing is risky.

→ Assymmetric encryption

→ Public key → encrypt
Private key → decrypt

→ More secure but slower.

(Q20) IPS Vs Firewall

→ Firewall

→ Filter incoming & outgoing network traffic based on rules.

→ First line of defence.

→ Allow/Block using IP/Port/Protocol.

IPS (Intrusive Prevention System)

→ Detects & stop attack real-time.
→ Monitor network for malicious activity.
→ Block attack automatically.
→ More advanced than firewall.

(Q23) Main Purpose of DNS server

→ Domain Name System (DNS)
→ domain name → IP address.
→ Internets phonebook.
→ Humans use names (google.com) but Computer use IP
→ Finds correct server when we open a website.

(Q24) Protocol and port no of DNS.

→ Protocol : TCP & UDP

Port Number : 53

UDP 53 : Normal DNS queries (fast)

TCP 53 : Large response / zone transfer.

(Q35) Bluetooth

Bluetooth

→ Short range
communication for
data transfer.
→ Range is
→ Low Power
consumption
→ Low Da
→ Connects
device (

(Q36) Rev

→ This
server
to C
→ Cl
→ Inv
→

(Q35) Bluetooth Vs Wi-fi

Bluetooth	Wi-fi
→ Short range wireless communication for small data transfer.	→ Wireless network technology for internet access.
→ Range is 10meter	→ Range : 100meter
→ Low Power consumption	→ High Power consumption
→ low bandwidth	→ High bandwidth
→ Connects few device (earbuds)	→ Many users use by router.

(Q36) Reverse Proxy

→ This sits between clients & web servers and forwards clients request to correct servers.

→ Client dont talk directly to Server.

→ Improved Security, load balancing, performance.

→ Hides real servers (Protects backend).

(Q49) Define 'Jitter'

→ Jitter is the variation in packet delay during transmission.

→ Packet arrive different time unevenly.

→ Affects voice/video calls & streaming.

→ Measured in milliseconds (ms)

(Q50) Why bandwidth is important to network performance parameters?

→ Bandwidth is amount of data that can be transmitted in a given time.

→ Measured in bps (bits per second)

→ Higher bandwidth → more data sent at once.

→ Affects download speed, calls.

→ Latency coordinate for network speed.

(Q51) Identify

→ if IP in

Eg:- 10.

its Privat

else: P

(Q54) Flow

→ Flow c

to pre

data

hand

→ Use

→ Rec

→ Se

(Q53) Identify Private or Public IP

→ if IP in reserved ranges
eg:- 10.0.0.0 - 10.255.255.255
its Private IP
else: Public IP

(Q54) Flow control achieved by TCP

→ Flow control is used by TCP to prevent sender from sending data faster than receiver can handle.

→ Use Sliding window protocol.

→ Receiver send window size (buffer)(capacity)

→ Sender sends data within that limit.

END

5-2-26
Thursday

Software Engineering Interview notes by Keshav Sharma

Q3, Q4, Q5, Q6, Q7, Q9, Q11, Q12, Q14, Q15, Q19, Q28

(Q3) SDLC & its phases

→ Software development life cycle.

→ Phases

→ Planning & Analysis
  → understand project goals, cost & feasibility.

→ Requirment definition
  → write SRS = Software requirment specification.
  → Functional + Non-functional requirments.

→ Design
  → Create System architecture & blueprint

→ Development
  → Coding of so
→ Testing
  → Find bugs
  properly
→ Deployme
  → Release
  time.

(Q4) Differen
→ Waterf
  → linc
  → Bes
→ V-mo
  → Tes
  → Eac
  pl
→ Incre
  → S
  → E

→ Development
→ Coding of the software

→ Testing
→ Find bugs, ensure system works properly.

→ Deployment & Maintenance
→ Release software + fix/update over time.

(Q4) Different SDLC Models.

→ Waterfall Model
→ linear, step by step process
→ Best when requirements are fixed.

→ V-Model (Verification & Validation)
→ Testing done along with development
→ Each phase has matching testing phase.

→ Incremental model
→ Software built in small parts.
→ Early working version delivered.

→ RAD Model (Rapid application development)
  → focuson fast development + prototype
  → Continuous user feedback
→ Iterative Model
  → development in repeated cycles.
  → improve software after each version
→ Spiral Model
  → Combines iteration + risk analysis
  → best for large & high risk projects
→ Prototype model
  → Early prototype shown to users.
  → Feedback to refine system.
→ Agile Model
  → Development in small sprints
  → Highly flexible & customer feedback driven.

(Q5) Wa
→ A
mo
De
Phase
1) Re
2) De
3) im
4) Int
5) De
6) M
Feat
→ N
→
→
→
Us
→

(05) Waterfall method & use cases

→ A linear and sequential SDLC model where each phase must finish before the next starts.

Phases

1) Requirement analysis :- Collect & finalize requirements.
2) Design :- System architecture & design.
3) Implementation + Unit testing → Coding modules.
4) Integration & testing (System) :- Combines modules & test
5) Deployment :- Release Software
6) Maintainance :- Bug fix + update

Features

→ No going back in previous phase.
→ Documentation heavy.
→ Simple & easy to manage.
→ Changes are difficult after start.

Use cases

→ government projects → Banking System
→ Healthcare System → Large enterprise

(Q6) Black box testing

→ internal code hidden
(Tester check input → output)
→ No coding knowledge
→ Functionality & user requirement focus
→ done by tester / end user.
→ Tested
  → user interface
  → input & output
  → system behaviour
  → functional requirments
→ Eg:- login page → enter username & password
  ↓
  check if login works

(Q7) White box testing

→ Internal code & logic are known for testing.
→ tester knows source code
→ Focus: code logic, paths, conditions

→ done by d
→ Tested
→ Code struct
→ loops & con
→ internal l
→ Eg:- All i

(Q9) Debugg
→ Proces
in soft
→ After
→ Perf
→ inpu
→ Bug fo

(Q11) Use-
→ cli
intera
→ Comp
(1) Sy

→ done by developers.

Tested

→ code structure
→ loops & condition
→ internal logic & paths.
→ Eg:- All if-else work correctly.

---

(10) Debugging

→ Process of finding & fixing bugs/error in software.
→ After testing find bugs.
→ Performed mainly by developers.
→ improves performance & correctness.
→ Bug found → locate cause → fix code → retest

---

(11) Use-case diagram (UML)

→ diagram showing how user interact with a system.

→ Components

(1) System:- Application being built.

(2) Actor: User or external system

(3) Use-case: Action by user.

→ Purpose
  → understand system functionality.
  → Show user requirment clearly.

→ Eg:- Actor: User
  Use case: Login, Register, Make payment

(Q12) Verification Vs Validation

Verification	Validation
→ check s/w built correctly as per SRS.	→ Checks wether software meets user needs.
→ Static: No code execution	→ Dynamic: involve running code
→ Done using reviews, inspection	→ Done using testing
→ "Are we building the product right?"	→ "Are we building the right product?"

(Q14) Cohesi

→ Cohesion
→ Meas
the task
→ High
→ Bette
→ Eg:-
→ Coupli
→ Mea
→ Low
→ Less
→ Eg:-

(Q15) A

→ A
Softwar
iteratio

(Q14) Cohesion and Coupling

→ Cohesion

→ Measure of how strongly related the tasks are inside module are

→ High cohesion = module does one specific task

→ Better design → preffered.

→ Eg:- Login module only handles login.

→ Coupling

→ Measure of dependency b/w modules.

→ Low coupling = modules are independent.

→ Less dependency → better system.

→ Eg:- Login module works without affecting payment module.

(Q15) Agile Software development model?

→ A flexible SDLC model where software is developed in small iteration with continuous feedback.

Features
→ Works in short cycle (sprints)
→ Continuous customer feedback.
→ Allows changes anytime
→ Frequent working software delivery.

Team work
→ Developers + Testers + Customer work together.

→ Advantages
→ Fast delivery.
→ High customer satisfaction.
→ Easily adapts to changing requirements.

(Q19) Regression Testing
→ Testing done to ensure new code changes does not break existing features
→ Re-run old test cases.
→ Done after bug fix or update.
→ Ensures existing functionalities

still works.

→ Eg:- After adding payment feature
→ check if login still works.

(Q28) What is SRS?

→ Software requirment specification.
→ A document that describes all software requirments before development starts.
→ Contains functional + non-functional requirments.
→ Act as agreement b/w clients & developers
→ Used as base for design & development.

→ Purpose

→ Avoid misunderstanding.
→ Provide clear project scope.

END