**Question Bank**

**L2 panel**

**OOPS & Lang fundas**

**1** How to write immutable class,why do we need them and rules to create mutable class, cross questions.

**2** write a immutable class employee wth id, name and Address. then cross questions like, how will you write getter, setter and constructor so that immutability of this class is preserved.

**3** Type casting(implicit & explicit) cross questions

**4** create a singleton class and cross questions on it---- like i have 4 containers so how many singleton objects will be created.

**5** How to make Singleton as Lazy ,egar loading & thread-safe . Demostrate with code

**6** what is inheritance and encapsulation? interfaces and abstract classes

**7** Overloading and overriding, some cross questions, Co-varient return type

**8** Explain final keyword, with class, method and variable

**9** use of protected access specifier, some cross question

**10** What are different types of Class -Loaders

**11** What is Marker Interface ? How to use cloning ? What is diff b/w shallow cloning/deep cloning

**12** What is diff b/w String vs StringBuffer vs String Builder ?

**13** What is Method Hiding ?What is different use of static in Java?

**14** What is Memory Management & call by value/reference in Java?

**Exception handling**

**1** Exception Handling in java, Complete hierarchy of exceptions,Types of exception (Cross Questions)

**2** Checked vs unchecked Exception

**3** What is Exception handling with method overriding rules in Java, explain with code ,cross questions on code

**4** if we write return statement in catch block then will finally execute?

If we write System. Exit then will finally block execut

**5** What is try-with-resource in Java? Explain with code

**6** Diff b/w ClassNotFoundException vs NoClassDefFoundError? Exception vs Error & cross question

**Multithreading**

**1** What is multithreading (cross-questions) ? What are Thread States ,explain in details, ways to create thread in java?

**2** What is concurrency api in java (cross-questions).

**3** How many types of locks,object level lock, class level lock, synchronized ,Internal architecture of synchornized block,

**4** what are reentrant lock advantages

**5** Difference Between CountDownLatch And CyclicBarrier in Java

**6** What is Semaphore ? Explain with real-life example .

**7** Explain Excecutor services

**Collections**

**1** What all collection data-structure used in your project

**2** Internals of HashMap,what happen if 3 objects have same hashcode in hashmap.

**3** When to use Linkedlist and ArrayList

**4** Concurrenthashmap internal working?difference between concurrenthashmap and synchronnizedmap in collection method & counter questions

**5** where you used hashmap & concurrenthashmap in ur previous project ? how concurrent is handled in iterator & counter questions?

**6** Difference between LinkedHashMap and Treemap and their implementation.

**7** How to merge two sorted list into a single list

**8** Map<Employee, String> -- given, hash-code is overridden, entered three elements in map, what will be the size of map.

**9** class employee of (name and dob), we have a list of 1000 employee, which data structure we use if we need to find the number of employee

having dob at a particular date.

**10** what is priority queues?

**11** What is Generics in Java?

**Subtotal**

**Java 8**

**1** Java 8 basic questions, stream api, map , filter, reduce ...etc

**2** default & Static method in Interface ,explain the advantage

**3** what is functional interface ? How to use functional interface in Lamda expression.What is Optional class ?

**4** What is Method references ,Types of Method references

**Subtotal**

**Spring/Boot**

**1** how you are managing application.properties file for different type of data in SpringBoot?

**2** Type of scope in spring

**3** What are profiles in spring boot, and cross questions on it.

**4** Basic questions on Rest api related to project

**5** Spring Aop (how did you use in your project) ,Annotation @Aspect, point cut expression, @Before, @After, @AfterReturning, @After throwing, @Around

**6** Spring boot advantages ,Annotation @Qualifier, @Primary, @postconstruct and @predestroy their uses and scenario-based questions

**7** How did you handled exception in rest api ?

**8** Given a jar, having some class how to use that its sendmail() method in our springboot project.

**Problem solving/DS**

**1** write a program to find most repeating

ababbbcdadaca

output: b

**2** write a program for finding max and min number

i/P: 7, 15, 5, 8, 88, 21, 89

**3** given two sorted arrays a= {20,35,40,60,70} b ={10,15,45,90,100} . write a program which will give me the third array in sorted order and will formed with these two arrays.

**4** [({})] -- return true

[{(]}) -- return false

there will be input of the string either of the two, create a method which will return true or false,

**5** scenario based questions

we have the list of stocks in which we are maintaining last 10 prices, also we can get the latest price for every 1 secs by third party api(class), how to handle this scenario

**Miscellaneous**

**1** have you worked on jms? questions on queues and event

**2** have you worked on kafka? Cross-Questions

**3** have you worked on microservices ? Cross-Questions

**DB**

**1** Where vs Having clause

**2** Indexing internals, including data structures, types of indexes - clustered vs non-clustered

**3** DB performance tuning, including use of explain plan and how db engine processes queries

**4** Types of SQL joins - Inner, Outer, Self, Equi, Non-Equi, Cross etc with examples

**5** Normalization and its forms - why normalization, denormalization. Trade offs

**6** Table relationships: 1:1, 1:N. How to implement M:N

**7** Basic concepts - Trigger, Synonym,Cursor etc

**8** View vs Materialized View

**9** Stored proc vs Function

**10** Types of SQL statements: DDL, DML, DCL

**11** Write query to eliminate duplicate employee records

**12** Top N query

**13** Correlated subquery with example

**14** Truncate vs Delete; difference across DB implementations

**15** Handling null using NVL, NVL2, COALESCE

**16** Tuning a slow correlated subquery

**17** Transaction - ACID properties, handling txn in Distributed DBs - patterns

**18** Isolation levels

**19** sql query for emplyee table(columns - emp\_id,emp\_name, manager\_id, salary), list out all the manager which has more salary than the employee.

SELECT \*

FROM employees w,

employees m

WHERE w.manager\_id = m.emp\_id

AND w.salary> m.salary;

**20** Delete all employee records whose name starts with A and salary between 1000 and 2000

DELETE FROM employees

WHERE salary BETWEEN 1000 AND 2000 AND name like ‘A%’;

**21** student table -- student id, student -name

course table -- courseid, course name

mapping table -- studentid, course id

# tables re given

write sql , which will return courseid and number of student enroll for that course.

SELECT

c.class\_name "Class Name",

count(distinct j.student\_id) "Number of Students"

FROM

class c

LEFT JOIN junction j ON j.class\_id = c.class\_id

GROUP BY

1

**22** table1 employee = id, emp\_name, dept\_id

table2 department = id, dept\_name;

print emp\_name, dept\_name null have to be allowed

**23** SQL query to find employee(s) with highest or Nth highest salary

**24** employee table(emp\_id, slary, depart\_id, emp\_name) department table(depart\_id, depart\_name), fetch the emplist for the maximum salary of each department

SELECT DEPT\_ID, MAX(SALARY) FROM department GROUP BY DEPT\_ID;

**25** Retrieve count of all unique names (after removing duplicates) in the employee table

DELETE FROM [SampleDB].[dbo].[Employee]

    WHERE ID NOT IN

    (

        SELECT MAX(ID) AS MaxRecordID

        FROM [SampleDB].[dbo].[Employee]

        GROUP BY [FirstName],

                 [LastName],

                 [Country]

    );

**26** Find all employees that do not have managers

SELECT e.emp\_name,

e.job\_name

FROM employees e

WHERE manager\_id IS NULL;

**27** Employee(id,name,salary, mgrId)

EmployeeContact(id,phone)

Retrieve phone numbers of all employees without managers

**28** Customer(id, name, city, grade, salesman\_id)

Write a SQL query to find those customers who belong to neither the ‘New York’ city nor their grade value exceeds 100. Return customer\_id, cust\_name, city, grade, and salesman\_id.

**29** Salesman(id,name,city), Customer(id,name,city)

find the salespersons and customers who live in same city. Return customer name, salesperson name and salesperson city.

**30** Student(id,name,marks), Grade(minMarks,maxMarks,grade)

List all student names along with their grades instead of marks, where marks not available display TBD

**31** Employee(id,name,salary, mgrId)

Display hierarchy of employees

SELECT

sub.employee\_id AS subordinate\_id,

sub.first\_name AS subordinate\_first\_name,

    sub.last\_name AS subordinate\_last\_name,

    sup.employee\_id AS superior\_id,

    sup.first\_name AS superior\_first\_name,

    sup.last\_name AS superior\_last\_name

FROM employee sub

JOIN employee sup

ON sub.reports\_to = sup.employee\_id

ORDER BY superior\_id;

**32** Salesman(id,name,city), Orders(id, amount, date, customerId, salesmanId)

Find all the orders, which are generated by those salespeople, who live in the city of London.Return ord\_no, purch\_amt, ord\_date, customer\_id, salesman\_id.

**33** Employee(id,name,salary,dept\_id)

Find all employees whose salaries are greater than the average salaries in their department

SELECT name, department\_id, salary

FROM employees e

WHERE salary > (select avg(salary) from employees e2 where e2.department\_id = e.department\_id);

**34** Customer(id,name,country)

List the number of customers in each country. Only include countries with more than 5 customers:

SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5;

**35** Employee(id,name),Orders(id,name,employeeId)

List all employees along with any orders they might have placed.

SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

**36** Account(DebitOrCredit,Amount)

Return balance at the end of calculating all debits and credits in the table

SELECT client\_id

, SUM(COALESCE(CASE WHEN action\_type = 'debit' THEN action\_amount END,0)) total\_debits

, SUM(COALESCE(CASE WHEN action\_type = 'credit' THEN action\_amount END,0)) total\_credits

, SUM(COALESCE(CASE WHEN action\_type = 'debit' THEN action\_amount END,0))

- SUM(COALESCE(CASE WHEN action\_type = 'credit' THEN action\_amount END,0)) balance

FROM my\_table

GROUP

BY client\_id

HAVING balance <> 0;

**37** Employee(id,name),Orders(id,name,employeeId)

List all employee names who have placed at least one order

SELECT o.EmployeeID

FROM Database.Orders o JOIN

Database.Customers c

ON o.CustomerId = c.CustomerId

GROUP BY o.EmployeeID

HAVING COUNT(\*) >= 1;

**38** Employee(id,name),Orders(id,name,employeeId, amount)

List all employee names who have placed at least one order with the exact amount 1000.

SELECT o.EmployeeID

FROM Database.Orders o JOIN

Database.Customers c

ON o.CustomerId = c.CustomerId

GROUP BY o.EmployeeID

HAVING COUNT(\*) >= 1 AND o.amount = 1000;

Java stream

Structuring your coding project in java

Composition – what it is and why we use it?

How we can make Java code thread safe?

Brief description of a devops & CI/CD approach

What test do you write for your code?

Databases: how does the index work?