TCS Digital

3 rounds → Technical → 30/35 min
→ Manager → 15/20 min
→ HR → 5/10 min

Technical (30 min)

Projects | cookexp. | OOPS | Programming | DBMS/SQL | CS fund.

1) Problem statement
2) Tech Stack
3) Features
4) Extend → Real time
5) Use - API
6) Cross Ques → all
7) Advantage and features of projects

Programming

1) features of language
2) basic function
3) In built Data structure
4)

DBMS/SQL

1) SQL Basics - 20-30 Questions
2) SQL → Table, ER, Schemas, Key → Normalization → ACID

OOPS

1) Definition
2) Example
3) Real time Use Case

CS fundamental

1) OS } — Basic
2) CN }

Mangerial

TCS → Working → Company brief

Tech stack, relocation → (AI/ML)

Problem Tracking → Situation based → Team issue

project

(i) Agile model & Waterfall module
(ii) In-depth questions of project
(iii) Technologies used
(iv) why u choose to work with these technologies?
(v) why choose MySQL why not MongoDB?
(vi) what are the security measures you took
(vii) If some attackers perform SQL injection in your database, how u will prevent this
(viii) Linux & cloud computing, Normalization → forms
(ix) html vs react
(x) Compler Query regarding joins
(xi) Puzzle Questions
(xii) Diff blw rdbms and DBMS (depth rdbms)
(xiii) Pattern problem
(xiv) why u had not hosted the project
(xv) Candidates key could be null
(xvi) Diff. blw foreign key and candidate key
(xvii) what is multer? why u used this
(xviii) Method overloading, yielding
(xix) global, static variable
(xx) Git Syntax & Operators
(xxi) Thread, OOPs, SQL, Query, Types SQL, Deadlock

language based

Q. Advantages of C++
Q. Why you choose C++?
Q. What is Pointer?
Q. Diff New & Malloc()? Operater & function

Q. Project related
→ role
→ Difficulty
→ Puzzle Question

Q OOPS Ques & DBMS
Q logic of longest Distinct character in string
Q. Situation based Questions

Q. Hobbis, & related Question

Q Why html is markup language
Q. what diff blw Html and Php
Q. Apache's system works
Q. Diff blw Trucate and delete
Q. What is JSP servelate
Q. Types of CSS & details
Q. which types of css have less/ast priority
Q. Also give me Query on Delete and Truncate and types of error.

Q SQL (DDL, DML, TCL, DCL, DQL)
Q Union and Join diff.
Q. Delete and Truncate

- Read and write operations in Java
- Inheritance all types
- why not multiple inheritance in java
- Polymorphism & types
- Method overloading in details
- Exception handling
-
- As I comfortable in c++ they asked why are u not using latest language
- SQL joins
- inner joins query example
  String buffer and string difference
- SQL Views
- SQL insert query example
  OOPs concepts
  HTML 5 tags
  Project based
- Encapsulation
  Abstract class vs Concrete classes
- SQL Joins
- SQL delete vs drop vs truncate
  commands linux
  20/0 what is OOP?
  Do classes contain memory (oops)
  Can We declare constructor private?
  why b.tech in IT
  NOT exam Questions
  Conflict in team

SQL

Q. Find 2nd highest salary from table?

```
Select max(salary) from Employee
where Salary < (select max(salary) from
                Employee);
```

OR

```
Select max(salary) As SecondHighestSalary
From Employees
where Salary <(select max(salary)
               from Employee);
```

Q. What are DCL Commands (Data Control Language)

DCL is subset of SQL that deals with permissions and access control to database.

```
GRANT Select, insert on employees To Alice;
```

```
Revoke privilages_type on object from user;
```

Q. Write Query to print each employee's emp-id, name and their manger's name.

```
Select E1.emp-id, E1.name As Employname,
                  E2.name As MangerName
FROM Employee E1
 LEFT JOIN Employees E2 ON E1.manager_id =
                           E2.emp-id;
```

Q. Write a query to calculate average salary

```
Select dept-id, AVG(salary) As avg-salary
 from Employees
  GROUP By dept-id;
```

Q. Use Join to combine two tables ie employee information and department information

```
Select e.emp-id, e.emp-name, e.salary, d.dept-n
from employees e
Inner join departments d on e.dept-id = d.dept-id
```

Q Handle null values in SQL

```
Select emp-id, salary, bonus,
       ISNULL(salary, 0) + ISNULL(bonus, 0) As
                                   total-pay
FROM employees;
```

Q. Diff b/w Where and having in SQL

* where filters rows before grouping
  used with individual column condition

* Having filters groups after aggregation
  used with aggregation functions.

Ex:-

```
Select dept_id, COUNT(*) AS emp_count
from Employees
where salary > 30000
Group By dept_id
having COUNT(*) > 5;
```

Q. Find count of employee in each department

```
Select dept_id, COUNT(*) AS employee_count
From Employees
GROUP BY dept_id;
```

Q. Update salary by 10%

```
UPDATE Employees
SET Salary = salary * 1.10;
```

Q. diff b/w where and having in SQL

* where filters rows before grouping
  used with individual column conditions

* Having filters groups after aggregation
  used with aggregation functions.

Ex:-

```
Select dept_id, COUNT(*) AS emp_count
from Employees
where salary > 30000
Group By dept_id
having COUNT(*) > 5;
```

Q. Find count of employee in each department

```
Select dept_id, COUNT(*) AS employee_count
From Employees
GROUP BY dept_id;
```

Q. Update salary by 10%

```
UPDATE Employees
SET Salary = salary * 1.10;
```

Q. To retrieve employees who joined in last 30 days.

```
Select *
from Employees
where join_date >= Current_date - interval 30 Day
```

Q. Indexes and their functions:-

→ They are special data structures used to speed up data retrieval.

→ They reduce the no of rows scanned during query by pointing directly to location of desired data.

```
# Create INDEX idx_emp_name ON Employees
                              (emp_name);
```

Q. Create a new table with columns from employee ID name, and Salary

```
CREATE TABLE Employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(100),
    salary DECIMAL(10,2)
);
```

Q. Top 5 higest-paid employees

```
Select emp_id, emp_name, salary
from Employees
ORDER BY salary DESC
LIMIT 5;
```

OR

```
Select TOP 5 emp_id, emp_name, salary
from Employees
ORDER BY salary DESC;
```

Q. Delete Duplicate rows

```
DELETE FROM Employees
where emp_id NOT IN (
    select Employees
    GROUP BY emp_email
    );
```

1. INNER JOIN

→ Returns only matching rows from both tables.

```
SELECT *
From Employees e
INNER JOIN Departments d ON e.dept_id =
                            d.dept_id;
```

2. LEFT JOIN

→ Returns all rows from table and matching rows from right table.

```
SELECT *
from Employees e
LEFT JOIN Departments d ON e.dept_id = d.dept_id
```

3. Right Join

→ Opposite of left Join
→ returns all rows from right table and matching rows from left.

```
SELECT *
FROM Employees e
RIGHT JOIN Departments d ON e.dept_id = d.dept_id
```

4. Full outer join

→ Returns all rows where there is a match in one of tables.

```
FULL OUTER JOIN Departments d ON e.dept_id = d.dept_id
```

Q. find employees who have not been assigned to any department

```
Select *
from employees
WHERE dept_id IS NULL;
```

Q. Primary Key:

→ It is a unique key identifies each record in a table.

```
# CREATE TABLE Employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(100),
    Salary DECIMAL(10,2)
);
```

Q Foreign Key:

→ It is a column in a table that refers to primary key in another table

```
Create table Departments (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(100)
);

Create table Employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(100),
```

dept_id INT,
FOREIGN KEY (dept_id) REFERENCES Department
(dept_id)
);

Q Add column to an existing table

ALTER TABLE Employees
ADD email VARCHAR(100);

Q Find names of employee starting with 'A'

Select first_name from employees
where first_name LIKE 'A%';

Q Print details of employees whos salary lies b/w 10000 and 50000

Select * from Employees
WHERE Salary Between 10000 and 50000;

Bubble Sort (check the adjacent element)

```
#include <bits/stdc++.h>
using namespace std;

void bubbleSort(vector<int>& arr){
    int n = arr.size();

    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-i-1; j++){

            if(arr[j] > arr[j+1])
                swap(arr[j], arr[j+1]);
        }
    }
}
int main(){
    vector<int> arr = {5, 1, 4, 2, 3};
    bubbleSort(arr);

    for(auto i : arr)
        cout << i << " ";
    return 0;
}
```

Insertion Sort (make a key & check for it position)

```
Void insertionSort(int arr[], int n){
    int i, key, j;

    for(i=1; i<n; i++){
        key = arr[i];
        j = i-1;

        while(j>=0 && arr[j]>key){
            arr[j+1] = arr[j];
            j = j-1;
        }
        arr[j+1] = key;
    }
}
```

Selection Sort (select first element check last element)

```
Void selectionSort(int arr[], int n){
    int i, j, min_index;
    for(int i=0; i<n-1; i++){
        min_index = i;
        for(j=i+1; j<n; j++)
            if(arr[j] < arr[min_index])
                min_index = j;

        Swap(arr[min_index], arr[i]);
    }
}
```

Star Pattern Question

①

```
int n=5;
for(int i=0; i<n; i++){
  for(int j=0; j<n; j++){
      cout<<"*";
  }
  cout<<"\n";
}
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

②

```
int n=5;
for(int i=0; i<n; i++){
  for(int j=0; j<i; j++){
      cout<<"*";
  }
  cout<<"\n";
}
```

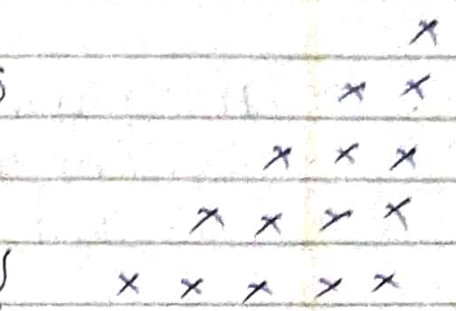
```
*
* *
* * *
* * * *
* * * * *
```

③

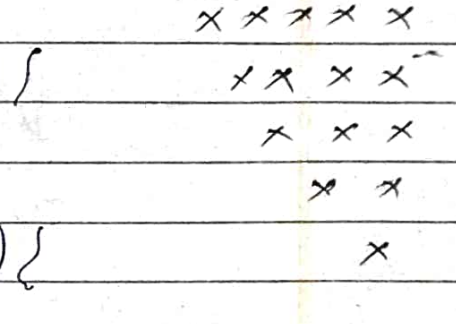
```
int n=5;
for(int i=0; i<n; i++){
for(int j=i; j<n; j++){
    cout<<"*";
  }
  cout<<"\n";
}
```

```
* * * * *
* * * *
* * *
* *
*
```

Q
```
int n = 5;
for(int i = 1; i <= n; i++){
    for(int j = i; j <= n; j++){
        cout << " ";
    }
    for(int j = 1; j <= i; j++){
        cout << "*";
    }
    cout << "\n";
}
```



Q
```
int n = 5;
for(int i = 1; j <= n; i++)
    for(int j = 1; j <= i; j++){
        cout << " ";
    }
    for(int j = i; j <= n; j++){
        cout << "*";
    }
    cout << "\n";
}
```



Q.
```
int n = 5;
for(int i = 1; i <= n; i++){
    for(int j = i; j <= n; j++){
        cout << " ";
    }
    for(int j = 1; j < i; j++){
        cout << "*";
    }
    for(int j = 1; j <= i; j++){
        cout << "*"; }
    cout << "\n";
```

Q Prime Number

```
int main(){
    int n=29;
    int cnt=0;

    if(n<=1)
        cout<<"Not Prime";
    else{
        for(int i=2; i*i<=n; i++){
            if(n%2==0)
                cnt++;
            }
        if(cnt>0)
            cout<<n<<"not prime";
        else
            count<<n<<" is prime";
    }
```

OR

```
for(int i=2; i<sqrt(n); i++)
        if(n%i==0)
            return false;
    return true;
}
```

Binary Search Algorithm

O(log n)

```
int binarySearch(int arr[], int key){
    int low=0, high = arr.length -1;
while(low <= high){
    int mid = low + (high-low)/2;
    if(arr[mid] == key){
        return mid;
    }
    else if(arr[mid] < key)
        low = mid+1;
    }
    else{
        high = mid -1;
    }
}
return ~~mid~~ -1;
```

Q to remove digit from string.

```
string removeNumbers(string str){
    int current=0;
  for(int i=0; i<str.length(); i++){
        if( ! isdigit(str[i])){
            str[current] = str[i];
            current++;
        }
    }
    return str.substr(0,current);
}
int main(){
   string str;
   getline(cin, str);
   cout<<remove.Numbers(str)<<endl;
}
```

OR

```
if(isdigit(str[i])){
        str.erase(i,1);
        i--;
    }
}
return str;
}
```

Q to remove digit from string.

```
string removeNumbers(string str){
    int current=0;
  for(int i=0; i<str.length(); i++){
        if( !isdigit(str[i])){
            Str[current] = str[i];
            current++;
        }
    }
    return str.substr(0,current);
}
int main(){
  string str;
  getline(cin, str);
  Cout <<remove.Numbers(str)<<endl;
}
```

OR

```
  if (isdigit(str[i])){
        Str.erase(i,1);
        i--;
    }
}
    return str;
}
```

© Reverse linked list

```
Node* reverse(Node* head){

    Node* prev = NULL;
    Node* curr = head;

    while(curr){
        Node* next = curr->next;
        curr->next = Prev;
            prev = curr;
            curr = next;
    }
    return prev;
```

About TCS

It is a global IT services and consulting company founded in 1968. It's part of Tata Group and one of the largest employee in India. It provides software development, consulting and digital solution across industries. It also known for innovation, client trust and strong training programs

CEO - K. Krithivasan