**Behavioural Interview Questions And Answers**

Behavioural questions seek to find out how you act in different situations and how you will handle future challenges.  
  
"Most interviews have at least one Behavioural question," says Manisha Mali Gaspe, a recruiter for Oceania business advisory services at EY. "It doesn't matter if you're applying for a junior position if you're a graduate, or someone with a lot of experience, Behavioural questions are common."

**1) Imagine you have a low performer in your team. How you will handle the situation? what would you do to help them?**

First, we need to identify the issue. Whether it is an issue with a person's knowledge (or) the person has been assigned an entirely new subject/project on which the person doesn't have much knowledge. often time the low performance is not their performance because sometimes a junior engineer will be assigned to a senior engineer's task (or) sometimes there is an external dependency with other teams, where the other team didn't communicate well with this engineer.

In case the engineer is repeatedly doing mistakes instead of presumably learning from past mistakes. The person needs to fix something. The simplest way to handle the situation is to help them with constructive feedback and offer help by just making yourself available for them, answering their questions, and unlocking them. If they repeatedly ask for help again that's an issue.

If there is a consistent issue related to performance, no improvement is perceived, and nothing really changes then eventually you might have to escalate to the person's manager.

**2) Describe a time when there is a conflict with your team. How did you help to resolve the conflict? Did you do anything to prevent it in the future?**

        Suggested going with your personal experience.

Example: There is a person who created a pull request a day before the freeze. Although it is not having a critical code in the worst case it may create a bug in the Prod. So to avoid that the person who reviewed it rejected it directly with comments "don't want to proceed before the freeze, we can do it later". So both parties went into a heated argument and a few comments were exchanged and clearly, there is a conflict bottling up. In this situation, I took them to a conference room and explained the good intent in both parties. Because the person who rejected the PR is having good intent for the team and should not get any questions if something goes wrong because of this in the freeze.Other party also agreed the PR is not well written. So they both agreed to execute the PR post the freeze.

Example2: An activity on a list of servers is shared between two members. They segregated the list equally between them. Unfortunately, one person forget to do it on one server from his list and assumed it is on another person's list and taken care of by other person. But on the last day during the validation, they went into a tense argument no one accepted that the server is on their list. Then I am also involved in that call, we did deep and went through all the chat history at the time of segregating the servers and came to know the first person missed the server he admitted that it is his mistake and said sorry. And the argument ended up with a friendly discussion. To avoid these kinds of issues we usually have a 30-minute friendly talk within our team, once in a while to build better bonding in our team.

These kinds of issues can be avoided with better communication. Before doing any conflict action, both parties should have chat/call communication for a better understanding of the good/bad intent of the action instead of proceeding with an action. I personally feel a 30 minutes friendly talk within the team per week/couple of weeks will help the team for better bonding within the team.

**3) Why do you want to work at [company\_name]?**

Assume you are working at Company A and you applied for company B.

I want to work at B for 2 or 3 main reasons:

1. Technical point of view - A lot of technologies(mention a few technologies used in company B that you came to know during technical interviews) used in company B really excites me.

 2. Excited about company B's culture of moving fast.

 3. You can add more points according to the company

**4) Describe a time when you strongly disagreed with a co-worker about a work-related decision? How did you go about making the final decision? what did you do after the decision was made?**

The first step is to dig deep to really try to really try to understand every person who is part of the disagreement, all of their ideas, their arguments, and their stances. You have to understand why someone is disagreeing with you, and why they came to the conclusion they came to. And they have to understand, why and how you came to your conclusion.

The second step is to take a step back. When seeing things from a higher level your organization/your team might agree/disagree with the decision due to organization policies.

Finally the third step - You want to strongly disagree and you have done it but once it comes to making a decision you have to commit to the decision that gets made. You can't hold grudges, especially if the decision doesn't go your way.If it goes your way then obviously you will be happy. But in the end, you have to commit to the decision and move forward. If you still feel that's not a beneficial decision to the organization, you can escalate to the decision maker probably the manager/director with all the arguments to prove that it is a bad decision.

**5) Imagine you and your team are in the middle of a major project at work, with many moving parts and complicated contacts, a lot of work, etc. A new software engineer joining your team. Your part in onboarding them. what do you do?**

It is important to properly onboard a new engineer. Because if we don't onboard them properly it will not only you gonna give them a very bad first impression about the team and your company it will make them unhappy. But it will also be having a negative impact on your team. Because certainly, you have to suddenly take more work than they were supposed to do. You have to hand-hold them more and it is obviously bad for your team and company. So, onboarding a new engineer is very important even if you are working in the middle of a very large project with a lot of work.

3 things you can do, to give them a great onboarding experience:

1. Have proper documentation

2. Make yourself available for the new engineer

3. Give them simple tasks initially to get them to understand the environment instead of overloading them without a proper understanding of the environment.

**6) How would you go about distributing work for the project across a team of software engineers.If you led a project in the past, describe what you did?**

The first thing I will do is scope out the work, split the work into sensible chunks, and segregate them accordingly. The work chunk that is highly critical and that is very difficult has to be prioritized and often times it has to be assigned to the most capable person.

Logistical issues also should be taken into consideration while assigning work, like someone unable to attend meetings with people who are working in different time zones.

In my experience, scoped out the work not too granular, but not too high level and assigned it in a way that makes more sense. parts of the project that didn't require too many contacts were given to some of the new engineers that were just joining the team. Parts of the projects that were very critical and that needed a lot of contacts were given to existing engineers who had already been on the team.

**7) Describe a time when you made a mistake? How did you deal with the repercussions of the mistake?what lesson did you learn from the mistake?**

Answer this question from your own experience and at the end add a statement you learned a valuable lesson from this mistake, try to elaborate couple more lines about what are those lessons.

Example: One mistake that stands out to me is a mistake that I made a year ago. I want to test the activity in a sandbox environment. Unfortunately, there were no sandboxes available in our organization servers. So I wanted to test it on my personal machine. I would like to send a couple technical steps documents to my personal email. While doing that instead of searching for the needed documents in a folder, in hurry I zipped the folder with multiple documents and sent them to my personal email address. Unfortunately, that was flagged by the organization's security system. During the scan, they found that a few documents were having organization server IPs along with passwords. Although I don't have any bad intentions it appeared to be a data breach from a security standpoint. So Security team initiated an investigation during the process they connected with me, first thing they asked me to remove the zip file shared to my personal email. Then the investigation continued in the backend, they contacted my manager about the same to get more details about me as a person and my behavior. When I realized a mistake happened, first I voluntarily informed my manager it happened in hurry but still, I am able to accept it as a mistake. Since he knows me & my attitude for quite some time, he didn't blame me. He said let's see what security people come up with. Throughout the investigation, I am very cooperative with the security team and I accepted my mistake I told them I am okay accept any kind of punishment including termination. After a couple of discussions among the HR Team, Security Team, My Manager, and Directors, almost after 2-3 weeks they ended the investigation with a warning letter to me. This is a life lesson I have learned, never share anything from organization email to personal email, even if you do occasionally for some non-critical stuff double-check that there is no sensitive information included in that.

**8) Describe a challenging project that you worked on. why it was challenging? what was your role in the project? how did you deal with the various difficulties of the project?**

Suggested picking the answer from your own experience.

Example: One of the applications teams decided to upgrade and migrate a large Postgres database approximately 3TB to a new operating system and new version at the same time. They have a very tight schedule and they were after me every day for a month. There was no automation for this task and every action I did was manual because this specific application is using different versions and different operating systems where we can't use existing automation setups. Eventually, the same activity has to be done on Stage and Prod environments as well. The challenging part here is they need a different OS, different DB version, huge DB size, different backup utility, and a very tight schedule. I am the only one in the team who is supporting this specific domain and other folks in the team don't have an idea how to implement things in this specific domain. And there is no time to give them any knowledge transfer. Parallelly I wrote shell scripts for the installation and replication setup and eventually completed their request on time. Since this is the first time I did this kind of activity it looks for me little challenging, especially with a very tight schedule.

**9) Describe a time when you have to deal with an outage at work. How did you handle the situation? what steps did you take after the issue is resolved?**

You can pick the answer from your experience.

Example: During a peak sales of a client, one of the NoSQL databases went down and it caused a revenue impact the application is not able to connect to the DB. All the leadership including directors, VPs, and even higher level people are on the call and there was huge pressure at that time as we have to acknowledge the questions in the call, update the same over the Chat, and in parallel have to work on fixing the issue. The reason for the DB down was the unexpected huge traffic that came in due to the huge promotions offered on the Retail site and the system was unable to handle the load and went down. So to fix the issue, stopped the incoming traffic at the load balancer layer and brought the DB cluster. Post that added new nodes with high capacity and removed old nodes. And introduced the traffic in stages instead of releasing 100% of traffic at a time. The new nodes with high capacity were able to serve the huge traffic. Post the issue is resolved we recommend all apps teams perform the proper load test before peak sales and also we are making sure we have at least 25% buffer resources available on the servers to add them instantly during unpredictable times. Also introduced GoAlert calls(on-call calls) for all kinds of production alerts.

**10) How do you think about receiving and giving feedback. Describe a time when you received tuff feedback and or a time you gave tuff feedback. How did you react to it? How did you give it?**

I am a firm believer in giving constructive feedback. You want to be very prompt and direct in giving feedback and in receiving feedback. It doesn't really help a person who is receiving feedback if you just tell them they are doing something wrong without telling them how they can improve. I still remember a day my colleague gave me feedback when I am writing a script. He suggested it would be better if I can reduce no of calls inside the script and he suggest a way how to do it. I totally agree with him and asked him more questions about the same. It helped me not only in that time but also helped in writing the scripts later. Most of the time, when you receive negative feedback you will not like that feeling but you quickly realize it is meant to improve you. I have given feedback to one of my colleagues when we were discussing one-one. He is a very experienced guy but his skillset was not improved much in the last 3-4 years. I gave him feedback to improve his skill set. Along with the feedback I have assured him I can help in that regard. And I have also shared the required documents to learn those skills as well. He felt happy about it.

**11) What aspects of software engineering do you think you are very good at? What about the areas you would like to improve? How do you plan on improving?**

Pick the answer from your own experience.

Example: This example assumes the person is a DBA. Along with DB technologies knowledge, I knew a few basic network-related and operating system-related concepts that helped me quickly find out the reason for the issue and able to reduce the impact duration. But still, I feel I need to improve some deep concepts of operating systems so that I can fix things on my own instead of waiting for other teams to fix them. The plan includes taking online courses, practicing, and keeping in touch with the platform team more often about the ongoing issues and how they resolve it.

**12) Describe a time when you went out of your comfort zone? why did you do it? What lessons did you learn from the experience?**

Again, pick it from your own experience.

Example: one time that I went out of my comfort zone was back in 2018. I am a firm believer that one of the best ways to grow is just to push yourself out of your comfort zone and challenge yourself with new things that you haven't done before, it is a great way to test to see if you like certain things, it is a great way to learn new things. So being a DBA who was working on only one DB tech stack. I took the responsibility of supporting two additional DB tech stacks, although I don't have any hands of experience with them. I was learning in parallel and supporting. What I learned from it, was probably a couple of lessons. On the positive side, I realized and got the confidence after a few months that I have done almost all kinds of activities to support those new databases. That will definitely help my career growth. On the negative side, it was a lot of pressure while handling calls as there was no help as everybody in the team is new to those technologies, even the issues were into my dreams, no quality sleep. So then realized it will good to take up one task at a time so that we can do it with more perfection. Anyway, that's a great experience.