**Explain the concept of parameter sniffing and how it affects query performance in SQL Server**

Parameter sniffing refers to the process by which SQL Server generates an execution plan based on the parameter values passed to a stored procedure or query. When a stored procedure or query is executed for the first time, SQL Server generates an execution plan based on the parameter values passed to it. This execution plan is then cached for subsequent executions of the stored procedure or query that use the same parameter values.

Parameter sniffing can significantly improve query performance, as SQL Server can optimize the execution plan to take advantage of the specific parameter values being passed. However, it can also lead to performance issues in some cases.

When the parameter values passed to a stored procedure or query vary widely, SQL Server may generate a suboptimal execution plan based on the initial parameter values used.

For example, if a stored procedure is designed to return all employees who have worked for the company for more than five years, but is called with a parameter value of ‘1’ (which represents employees who have worked for the company for one year), SQL Server may generate an execution plan that is optimized for returning a small number of rows, which may be inefficient when applied to a larger dataset.

Similarly, if the dataset being queried is highly skewed, with a few outliers having significantly different characteristics from the majority of the data, the execution plan generated by parameter sniffing may not be optimal for the outliers.

To address these issues, SQL Server provides a number of options for managing parameter sniffing, including:

1. The OPTION (OPTIMIZE FOR) query hint, which allows you to specify a specific parameter value to optimize the execution plan for.

2. The OPTION (OPTIMIZE FOR UNKNOWN) query hint, which tells SQL Server to generate an execution plan that is optimized for a typical parameter value.

3. Parameter masking, which involves creating a copy of the parameter value within the stored procedure or query and using the copy instead of the original parameter value to generate the execution plan.

4. Dynamic SQL generation, which involves generating the SQL statement dynamically based on the parameter values passed at runtime, rather than using a precompiled stored procedure or query.

It’s important to note that parameter sniffing is just one factor that can affect query performance in SQL Server, and that there are many other factors to consider when optimizing a database. Nonetheless, understanding the concept of parameter sniffing and how it can impact performance is an important part of designing and maintaining efficient database systems.

Wang, X.Y.. SQL Server: Interview Questions and Answers (Advanced Topics in Database Book 4) (pp. 94-95). Kindle Edition.