When the performance of the DB queries starts degrading, what can be done so that the DB query response is received in a reasonable time ?

=> Fetching the more data than needed, basically query is not optimized, we can always check that how much data a particular query is going to scan using explain statement

=> Always Create an in index for a particular column to for faster the process of execution time.

Database indexes

=> We can create a indexing of any column for faster the fetching of data. Internally it creates a tree like structure to reducing the time complexity of searching the data.

Type of indexes

=> Clustered Index: it sorts the data and store a it in tree format. Data row will be at leaf and primary key is a clustered index.

=> Non-Clustered Index:it stores the and sorts the data in key values.

When will indexes fail, and why?

=> When the particular indexes are NULL values, columns which has frequent update and insert operations.

What about transactions, when to use them, when not to use them? How can transaction create performance issues?

=> Transaction has ACID properties A(Atomicity) C(consistency) I(isolation) and D(Durability),i.e transaction is a single unit of work, where data should be consistent before and after the insert and update

=> if your priority is not about consistency and performance not use transaction because operations will become slower comparatively

What is Deadlock, what is the cause, and how can we avoid it?

=> Deadlock is when a Two process wants to acquire the a resource and but both the process are preventing each other to acquire the resource.

=> waiting for the particular resource or process to finish it if not kill the process.

=> by defining the priority to the process to enter the critical section.