Please provide your answers via your experience or research related to the following questions:

* How does a B-tree differ from a B**+**-tree?  Why is a B**+**-tree usually preferred as an access structure to a data file?
* Discuss types of indexes supported by your RDBMS(s) that you have used. You may elaborate what conditions you will create what types of indexes?
* Share your experience on database performance problems and how to solve them. Also explain the tools you or your colleagues may use and required knowledge to conduct performance tuning.
* Discuss CRUD Matrix and Query by Example. Elaborate some of the usages.
* Share lessons learned from your database project.

B- Trees are trees that can have multilevel indexes which can be useful in searching for records in a data file. B- Trees are always balanced, which reduces wasted space by deletions. Insertion and deletion are generally simple, but can be complex in cases where you have to enforce the balance of the tree. B+ trees are a variation of B- trees, where the B+ trees only store data in the leaf nodes, and store keys only in the internal nodes. This makes B+ trees the generally preferred choice since records are linked like in a linked list, making search faster. B+ trees also have easier deletion than B- trees.

The RDBMS that I used for the database project was postgresql, which supports several types of indexes. Postgres supports B-tree, hash, gist and gin indexes. Postgresql also supports multicolumn indexes partial indexes. Some scenarios that I would use indexes is a B-tree for ordered records, or hashing for fast insert/search. I have not personally used any of these indexes for the database project.

I have no experience with improving database performance, however some strategies that I would use include things like indexing or reorganizing tables. Indexes can be used to increase search speed and performance for databases, especially for queries that use a lot of joins or have computationally expensive steps.

CRUD matrix is a good tool for understanding which functions perform which operations on which database entities. This can organize the functions and entities in the database, and show what operations access what data. CRUD does this in a graphical way that is easy to parse. Query by example is a useful tool which allows a user to retrieve data from a database using a GUI. This helps individuals with less experience with databases to specify what they want to query, and allows them to begin to learn. Most DBMS tools have some form of a query by example tool to help the users.

Throughout the database project I learned a lot about database implementation in SQL, which is exactly what I set out to do in this course, so for that I am thankful. In terms of strengths, I think I have a very good grasp of how to design a database in a way that makes sense with foreign keys and primary keys and relationships. One possible weakness still is with more complicated SQL queries, for example the query from HW4 was difficult for me. However, as I practice more I am getting better, and some of the queries included in this report were relatively complex.