Query Performance Analysis Report

1. Which query was most affected by more data?
Q1 (Simple Filter) was most affected. Without an index, it took **1,000,000.7 ms**, but with an index, only **424.7 ms**. This happened because without an index, the database had to scan all 1 million rows.

2. Which query improved the most after adding an index?
Q1 also showed the biggest improvement, with a **99.96% faster result**. The index let the system directly find the needed rows instead of checking the whole table.

3. Which query improved the least with indexing?
Q2 improved the least (about **11% faster**). This is because joins are already optimized in some ways, and indexing doesn't remove the heavy cost of the join operation itself.

4. Downsides of adding too many indexes: - **Insert becomes slower** (because new data must be added to all indexes). - **Update becomes slower** (since indexes need to be updated too). - **Delete becomes slower** (removing entries from indexes takes time). - **More storage use** (indexes take extra disk space). - **Extra maintenance work** (database keeps all indexes up to date).

Summary:
Indexes give a huge boost to query performance, especially for filters and lookups. But too many indexes can slow down data changes and use more space. The best approach is to index only the most commonly used columns in queries.