# Project Workshop - 10

### Topic:

Performance & Tuning

1. Use indexing: Indexing can significantly improve the performance of a database. Make sure that all the tables are indexed appropriately based on the type of queries you will be running.
2. Optimize queries: Poorly written queries can negatively impact the performance of a database. Make sure that all the queries are optimized and use the appropriate indexes.
3. Use caching: Caching can help reduce the load on the database by storing frequently accessed data in memory. Consider using a caching layer such as Redis or Memcached.
4. Normalize the database: Normalization is the process of organizing the data in a database to minimize redundancy. This can help improve the performance of the database and reduce the risk of data inconsistency.
5. Use appropriate data types: Make sure that you use the appropriate data types for the columns in your tables. Using the wrong data type can negatively impact the performance of the database.
6. Set appropriate buffer sizes: Make sure that you set appropriate buffer sizes for the database server. This can help improve the performance of the database.
7. Use connection pooling: Connection pooling can help reduce the overhead of establishing new database connections. Consider using a connection pooling library such as HikariCP or Apache Commons DBCP.
8. Monitor database performance: Regularly monitor the performance of the database to identify any potential bottlenecks or issues. Use monitoring tools such as Nagios or Zabbix to help with this.