1. I read both of the articles
2. **ON ATTACHED PDF**

3. Data Definition Language and Data Manipulation Language are two highly essential components of database management systems, each serving unique purposes in handling data. DDL defines the structure of a database, including the creation and modification of database objects like tables, indexes, views, and constraints. It is used by database administrators to establish the schema and set constraints for data organization. In contrast to this, DML is geared towards manipulating the data within those established structures. It encompasses operations like adding, retrieving, updating, or deleting data. DML employs the WHERE clause to filter and conditionally operate on data rows, making it more dynamic in data manipulation. While DDL statements are executed less frequently and are mainly handled by administrators to define the database's blueprint, DML statements are executed regularly by application developers or end-users to interact with and modify the database's contents directly. The distinction lies in their focus, where DDL shapes the database's architecture, and DML drives the day-to-day manipulation and retrieval of data within that structure.
4. Two-tier and three-tier client-server architectures represent distinct models for organizing and distributing computing resources in a network. In a two-tier architecture, the system consists of two primary layers: the client layer and the server layer. The client layer handles both the presentation and application logic, while the server layer is responsible for data storage and processing. This simplicity makes two-tier architectures straightforward but may lead to challenges in scalability and maintainability, as changes often require modifications on both the client and server sides. The three-tier architecture introduces an additional layer, the middle tier or application layer, between the client and the server. This results in a three-layered structure: presentation layer, application layer, and data layer. The client remains responsible for presentation logic, while the middle tier manages business logic and processing, leaving the server to handle data storage. This additional layer enhances modularity, scalability, and flexibility. Updates to the business logic can be implemented in the middle tier without affecting the client or server components, making maintenance and development more straightforward.