**Homework#1**

**ACS575 Database Systems, Spring 2024**

**Q1. The most classic paper to introduce database systems is M. Stonebraker and J. M. Hellerstein, “What Goes Around Comes Around”. Read the paper and answer each of the following questions briefly.**

**(1) What is the notion of data independence? Why is it highly desirable?**

**Ans:** Data independence refers to the capability of modifying the physical or logical structure of a database without causing any impact on the application programs utilizing the data. It is advantageous for database systems as it provides flexibility and adaptability. Physical data independence allows changes in the storage of data without affecting applications, while logical data independence enables modifications to the database schema without impacting application programs. This flexibility is crucial for the smooth evolution and maintenance of complex and long-lived database systems.

**(2) What are the key ideas behind the relational model? Why/how it improves previous model?**

**Ans:** Ted Codd's 1970 relational model addressed challenges faced by IMS programmers by introducing three key ideas:

* Simple Data Structure: Tables replaced complex storage proposals, improving logical data independence.
* High-Level Language: A set-at-a-time Data Manipulation Language (DML) provided efficient access and high physical data independence, enabling flexible data storage.
* Flexibility and Representation: The model accommodated diverse scenarios, simplifying representation of complex relationships, a notable improvement over previous models.

These ideas contributed to the relational model's success, establishing it as the standard with SQL as its language.

**(3) What is the motivation of the OR (Object-Relational) model? What are the main additions in ORDB?**

**Ans:** The Object-Relational (OR) model was developed to address challenges in data management, particularly in the context of geographic information systems (GIS) and engineering applications. The primary motivation behind its creation was the inefficiency of conventional relational systems in storing and querying geographic positions. The Object-Relational Database (ORDB) model introduced user-defined data types, operators, functions, and access methods into a SQL engine to provide a more flexible and customizable approach to data management. This adaptation aimed to meet specific market needs, such as GIS and engineering applications, and included support for persistent programming languages like persistent C++. Despite these efforts, the ORDB model faced challenges in gaining widespread market acceptance due to the absence of standards and vendor competition, resulting in limited commercial success.

**(4) Data can be divided into four categories: rigidly structured data, rigidly structured data with some text fields, semi-structured data, text (unstructured data). Describe each one and give an example for that.**

**Ans:**

The document classifies data into four categories based on their structure:

* Rigidly Structured Data: Data in this category must adhere to a specific schema, crucial for core business processes. Examples include payroll data, where any deviation from the defined structure may lead to errors.
* Rigidly Structured Data with Some Text Fields: This category involves data with significant structured information (e.g., personnel records) along with some free text fields. An example is health plan information in personnel records with additional free text fields for manager comments during employee reviews.
* Semi-Structured Data: This category includes data with some structure, but instances can vary in the fields present and how they are represented. Examples include want ads and resumes, where there is a degree of structure, but instances differ in their representation.
* Text (Unstructured Data): Documents in this category have no specific structure, characterized by free-form text. Examples are documents without predefined structure, where content is not organized according to a specific schema.

These categories provide a framework for understanding the varying degrees of structure and organization within different types of data.

**Q2. After reading “Section 5” of M. Stonebraker et al. "One Size Fits All", give a short answer on each question following:**

**(1) Does one size fits all? If not, explain the reason.**

**Ans :**

The document contends that the "one size fits all" concept is obsolete in the commercial world. Criticizing traditional database management systems, it suggests a shift towards independent database engines, citing examples from the data warehouse and stream processing markets. The authors conclude that the uniform approach is no longer viable in the commercial database management system market.

**(2) What components should be considered in a sensor network system with databases.**

**Ans :**

In a sensor network system with databases, efficient operation relies on key components:

* Data Storage: Employ a robust database management system for efficient storage of large data volumes from the sensor network.
* Data Processing: Enable capabilities for real-time data analysis and aggregation from the sensor network.
* Communication: Ensure robust communication for effective data transfer between sensor nodes and the central database.
* Energy Efficiency: Design the system to conserve energy, particularly in communication and data movement, given the resource-constrained nature of sensor networks.
* State Management: Implement capabilities for managing state information crucial for sensor network applications, including reference data, translation tables, and historical data.
* Application Logic: Support domain-specific message processing and query activities, facilitating seamless integration of application logic with database processing.
* High Availability: Design the system with fast recovery and replication capabilities to handle potential failures and ensure high availability.
* Lightweight Database Abstractions: Incorporate lightweight database abstractions optimized for data movement, diverging from traditional data storage methods.

By considering these components, a sensor network system with databases can effectively meet the unique requirements and challenges of sensor network applications.

**(3) Describe the characteristics of (typical) text search systems which are different from conventional business-processing applications.**

**Ans :**

Text search systems differ significantly from conventional business-processing applications, as outlined in the document. The traditional database management system (DBMS) architecture, designed for business data processing, is ill-suited for text search systems due to their unique characteristics:

* Data Storage Technology: Text search engines require specialized storage systems, unlike conventional DBMS technology, to handle massive and ever-increasing datasets efficiently.
* Workload: Text search systems handle a combination of streaming data incorporation into the search index and ad hoc look-up operations, with write operations being mostly append-only and read operations sequential—distinct from the transactional nature of conventional business-processing applications.
* Storage Machines and High Availability: Involving numerous storage machines, text search systems prioritize high availability, necessitating fast recovery and replication capabilities, considering failure as the norm.
* Performance and Flexibility: Traditional DBMSs with built-in text search capabilities are inadequate for text search systems due to their heaviness and inflexibility, falling short of meeting specific performance and availability requirements.
* Database Abstractions: Designing flexible, lightweight database abstractions optimized for data movement, such as TinyDB, is crucial for meeting the specific needs of text search systems.

In summary, the unique requirements and workload patterns of text search systems underscore the necessity for specialized database engines tailored to their distinct characteristics.

**Q3. Name the six (/five) phases of System Development Life Cycle (SDLC), and explain the purpose and outputs of each phase.**

Ans :

The System Development Life Cycle (SDLC) has five phases:

**Getting and Understanding Requirements:** First, we gather and study what the system needs from the people involved. This helps us figure out what the business and users want. We then create a detailed document with these requirements and check if the project is doable.

**Creating a Design Plan**: Next, we make a plan for how the system will look and work based on the gathered requirements. This plan includes diagrams and technical details. It's like creating a blueprint for the system. The result is a detailed document describing how the system will be built.

**Building the System:** Now, we actually build the system using the plan we created earlier. We write the code and put everything together to make a working prototype or a beta version of the system.

**Testing the System**: After building, we test the system to make sure it does what it's supposed to do. We check if there are any mistakes or problems and fix them. The result is a system that has been tested and is ready to be used.

**Maintenance :** Lastly, we maintain and update the system to make sure it keeps working well. We make changes if needed to meet the evolving needs of the business. The result is a system that stays current and meets the business requirements.