ECE 569 Database System

Description:

The concept of the database has evolved significantly over the decades. It began with file-based systems in the 1950s, then progressed to hierarchical (tree-like) and network (graph-like, pointer-based) databases in the 1960s. The relational database model, introduced in the 1970s, marked a major milestone, followed by distributed databases in the same decade. The 1980s saw the rise of the data warehouse for analytical workloads. In the 1990s, object-oriented and object-relational databases emerged to support multimedia and complex data types, along with federated databases for integrating heterogeneous systems. The 2000s brought the era of NoSQL, Big Data, and cloud databases, enabling large-scale, flexible data management. The 2010s introduced NewSQL (combining SQL with scalability) and data lakes for raw, schema-flexible storage. Now, in the 2020s, we see the emergence of AI-integrated and LLM-powered databases, such as those used by ChatGPT.

Although the term “database” is commonly associated with relational databases due to their historical impact, ECE569 reinstates the original meaning: a structured collection of data that can be easily retrieved, managed, and updated. Through Database Design and Implementation, this course begins with relational databases and the foundational ACID properties, then explores distributed systems, including federated databases, data lakes, peer-to-peer (P2P) models, cloud databases, NoSQL, and NewSQL architectures. The study concludes with an examination of AI-integrated databases, including applications like ChatGPT, and the underlying protocols for concurrency control, consistency, and recovery. The USPTO Public Patent Search tools will also be introduced as a resource for exploring emerging database technologies relevant to this course.

Class Syllabus:

Week 1: Database introduction, SimpleDB installation and demo, introduce to JDBC

Week 2: Disk and File Management, persistent data storage, RAID data storage technologies.

Week 3: Memory management; buffer manager, log manager, optimize data management

Week 4: Buffer pool and buffer manager, page replacement strategies, buffer contention

Week 5: Transaction management, transaction object; ACID database properties; recovery management, log records, redo, undo, checkpointing

Week 6: Recovery manager, log record class; rollback & recover; Concurrency management, concurrency control, serializable schedule, lock table, lock protocols, deadlock and deadlock detection.

Week 7: Record management, homogeneous/nonhomogeneous record storage, structure of records; fixed/variable length of fields, record layout strategies; insert, delete, modify, retrieve data; table scans.

Week 8: Query processing, relational algebra and operators, query tree, pipelined query processing, SQL predicate structure and SQL predicate creation.

Week 9: Distributed database, data partition/distribution, DDBMS directory; DDBS characteristics (transparency, replication, autonomy, fault tolerance, scalability, balanced ACID – CAP theorem); distributed computing, P2P, client/server, federated database, data lake, cloud computing, data delivery alternatives

Week 10: DDBMS transparent access; types of transparency; distributed concurrency control protocols, reliability protocols, data control protocols; DDBMS query processing (localization & optimization), query plan

Week 11: Concurrency control synchronous protocols: 2PC, 3PC; asynchronous protocols: Primary-backup protocol, quorum-based protocol, gossip protocol, Paxos/Raft consensus protocols; DDBMS failure/recovery control/management

Week 12: Distributed data control; deadlock management: wait-die, wound-wait, timeout, deadlock detection algorithms; security control; NoSQL overview, ChatGPT overview.

Week 13: NoSQL key features, data structures: document store, key-value store, column-family, graph database; NewSQL key features, how NewSQL manage CAP; NewSQL data structure/schema

Week 14: AI-Integrated database, LLM, Pre-training, transformer architecture: self-attention, multi-head attention, positional encoding/encoder-decoder