Module 3 – Moving Beyond the Relational Model

- Benefits of the relational model
    - Relational databases use SQL which is a widely adopted standard
    - ACID compliance ensures transactions are processed reliably
    - Relational models are well-suited for structured data with clear schemas
    - Can handle large data volumes (scalability)
    - There is extensive tooling and expertise available
- Ways a relational database can increase efficiency:
    - Indexing: Speeds up searches by organizing data.
    - Storage Control: Optimizes data layout for efficiency.
    - Column vs. Row Storage: Column-oriented storage benefits analytics; row-oriented storage benefits transactional systems.
    - Query Optimization: Uses execution plans to improve query speed.
    - Caching & Prefetching: Reduces repeated database calls.
    - Materialized Views: Stores query results for fast access.
    - Precompiled Stored Procedures: Speeds up frequently used operations.
    - Data Replication & Partitioning: Improves availability and load balancing.
- Transaction processing – a transaction is a sequence of one or more of the CRUD operations performed as a single, logical unit of work
    - Commit – the entire sequence succeeds (save all changes if successful)
    - Rollback / Abort – the entire sequence fails (revert changes if failure occurs)
    - Transactions help ensure:
        - Data integrity – prevents incomplete updates
        - Error recovery – enables system restoration
        - Concurrency control – manages simultaneous transactions
        - Reliable data storage – ensures durability
        - Simplified error handling – manifest failures systematically
- ACID properties
    - Atomicity – a transaction is all or nothing, it either completes fully or doesn't happen at all
    - Consistency – ensures database transitions from one valid state to another
        - Ex: if a transaction transfers money, it must debit one account and credit another without violating integrity constraints
    - Isolation – transactions execute independently, preventing interference
        - Three common issues arise without proper isolation:
            - Dirty Read – a transaction reads uncommitted changes from another transaction