CS 1555

Lecture 3

**Introduction (continued)**

NoSQL

- NoSQL is an overloaded term, it means that the database doesn’t use the table model and doesn’t have ACID properties

Reliability

- Enforcing integrity constraints (transactions are not committed if they violate integrity constraints)

- Ensuring data integrity despite failures

- Security: encryption, authentication, data domains & compartmentalization

- Access control: who, what, how

File system performance problems (opposed to database model)

- redundancy: waste of space

- waste of effort: same change in multiple places

- need for coordination: inconsistent data and independent updates

- no such performance problems with databases

DBMS serves as interface between application and database

Efficiency and performance

- Space efficiency: minimizes redundancy by storing data once

- Time efficiency: eliminates need for multiple updates, enhances query performance with optimizations and access methods, allows many users to access concurrently

When SQL-DBMS is inappropriate

- Disadvantages: price to buy, additional expertise needed

- Overkill for simple/small databases: if application is simple or not expected to change, if concurrent access isn’t required, if it can tolerate failures

When DBMS is needed

- need to integrate large volumes of related data efficiently and correctly

**Relational Database Model**

Mathematical concept of relation

- Let D1, D2, …, Dn be domains

- Cartesian product D1 x D2 x … x Dn is the set of all possible ordered n-tuples such that v1 belongs to D1, v2 belongs to D2, …, vn belongs to Dn

- A relation is any subset of the Cartesian product

- Empty set is also a valid subset

Two notations

- Relational schema R is denoted by R = {A1:D1, A2:D2, …, An:Dn} or R = {A1, A2, …, An}

- Set of attributes notation: tuple t of r(R) is denoted by t = {A1\*v1, A2\*v2, …, An\*vn}, vi belongs to Di, 1 <= I <= n or t = <(A1:v1), A2:v2), …, (An:vn)>, vi belongs to Di, 1 <= i <= n

- List of attributes notation: tuple t of r(R) is denoted by t = (v1, v2, …, vn), vi belongs to Di, 1 <= i <= n

SQL insert: implicit (list) vs explicit (set)

- Implicit: INSERT INTO student VALUES(…)

- Explicit: INSERT INTO student(field 1, field 2) VALUES(value for field 1, value for field 2)