Tanner Maasen

Database Systems

I. Importance of Db's in the Internet/Smartphone World

- PC's not popular until early 80's

Internet: global computer network of networks
- development of WWW, Web Browsers, & Web Sites
- "brick-and-mortar" – online shopping

Web 2.0: Web applications; dynamic content; accounts

Data: recorded facts & numbers

Database: structure used to hold/store data
- process data to provide information

Client-Server architecture: client apps are used by users to obtain services (browsing, shopping, etc). Services are provided by servers that hold the dbs needed

II. Characteristics of Db's

1) Relational Db
- data stored in tables (r x c)
- a row has an instance of the thing of interest
- a column has a characteristic common to all rows
- row = record
- column = field
- use naming conventions when creating tables

* a db contains both data & relationships among the data

Primary Key: a unique identifier for each row of data (id's)

Surrogate Key: auto-generated primary keys that're assigned in the db itself

Composite Key: when 2 or more columns combine to ~~form~~ form a primary key in a table

Foreign Key: provides link b/w 2 tables; the same column in 2 tables → creates a relationship

Information:

- Knowledge derived from data
- Data presented in a meaningful context
- Data processed by summing, ordering, avg, grouping, comparing, or other similar ops (use SQL)

Database Examples

1.) Single-User Db Apps
   - not many records; can buy software license
2.) Multi-User Db Apps
   - around 100K rows in 5-10 tables
   - concurrency - control mechanism for dupl. data
   - CRM - customer relationship mgmt
3.) E-Commerce Db Apps
   - very important
   - largest dbs track browser behavior (Web Activity Databases)

- these dbs use existing data for reporting to provide insight to mgmt
- Data Mining apps predict future performance

III. The Components of a Db System (4)

1.) Users
2.) the Db app - SQL is universally known
3.) the DBMS
4.) the Db itself

- DBMS is ~~used~~ a computer program used to create, process, and administer the db (i.e. Microsoft SQL Server) (DB2, Access)
  - companies never write their own b/c it's huge
- Db App is 1 or more programs that serve as intermediary b/w users & the DBMS
  - Apps use SQL ~~queries~~ statements & pass them to the DBMS

* Basic Functions of Db Apps
- Create & Process Forms
- Process User Queries
- Create & Process Reports
- Execute Application Logic
- Control the App Itself

* Functions of a DBMS
- Create db & tables
- Create supporting structures (indexes) - easier to locate lot of data
- CRUD
- Maintain db structures
- Enforce rules
- Control concurrency
- Perform backup & recovery

Referential Integrity constraints: rules to ensure no bad data is entered that'll cause errors

Concurrency: ensuring 1 user's work doesn't interfere w/ anothers

- DBMS also contains security features & backup/recovery

The Database: a self-describing collection of integrated tables

Integrated Tables: tables that store both data and the relationships among the data

Metadata: data that describes other data

- can be used to see if a certain db structures exist

Indexes: db structures that speed up the searching/sorting of data

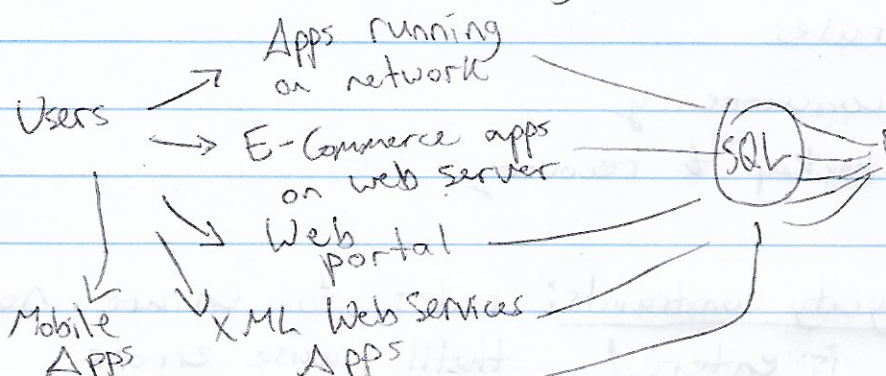
IV. Personal vs Enterprise-Class Db Systems

- 2 types

Personal →

- Microsoft Access is a DBMS and an application generator, which are the form, report, & query components
  - Access is a personal system that's very good at hiding the underlying SQL; easy to use
  - Usually first thing students work with } to learn on

Enterprise →



- the more powerful the DBMS, the more difficult it is to use

IV. Database Design

- proper structure of tables, relationships b/w them, appropriate data constraints, & more

3 types of design:

1.) design from existing data
2.) design for new systems development
3.) redesign of an existing db

1a) Analyze spreadsheets & other data tables
1b) Extract data from other db's
1c) Design using normalization principles

2a) Create data model from application requirements
2b) Transform data model into db design

3a) Migrate db's to newer db's
3b) Integrate 2 or more db's
3c) Reverse-engineer & design new db's using normalization & data model transformation

Data Warehouses / Data Marts: db's that store data specifically organized for research & reporting, then usually exported to analytical tools

- Data model is transformed into database design
  → blueprint ← construction

Data Migration: database is adapted to new or changing requirements (tables, relationships, constraints, etc.)

Redesign = Migration OR Integration

V. What You Need to Learn

Knowledge Worker: user that prepares reports, mines data, etc
programmer: writes apps that process the db.
db administrator: designs, constructs, & manages the db

VI. Brief History of Db Processing

- emerged around 1970
- original data stored on magnetic disks & drums
* - Figure 1-27 Db History
- need for data integration / consolidation

DL/I: used hierarchies/trees to represent relationships b/w data
- E.F. Codd developed relational db model, using relational algebra
- DB2 (IBM) & Oracle Db are still used today
- in 1991, Microsoft Access emerged & stole all business for PC DBMS
- 1980: OOP → OO DBMS → object-relational DBMS (hybrid
  - it emerged too late & never caught on
- 1990's: internet is here → problem: HTTP is stateless protoc
  - soon fixed
- open source & free DBMS downloads emerged

- late 1990's: XML developed to fix HTML problems w/ exchanging db data
- post 2009: NoSQL & Big Data movement

NoSQL: "Not Only SQL"; db's that don't follow the relational model

Big Data: something ~~was~~ was needed to handle larger sets of data

  - Big Data (+) non-relational db (NoSQL) = Facebook / Twitter
  - virtualization / cloud computing in the future (Ch. 12)