Chapter 12 – Data and Database Administration

Instructor: Mark Llewellyn
markl@cs.ucf.edu
CSB 242, 823-2790

School of Electrical Engineering and Computer Science
University of Central Florida
Chapter 12 Objectives

• Definition of terms.
• List functions and roles of data/database administration.
• Describe role of data dictionaries and information repositories.
• Compare optimistic and pessimistic concurrency control.
• Describe problems and techniques for data security.
• Describe problems and techniques for data recovery.
• Describe database tuning issues and list areas where changes can be done to tune the database.
• Describe importance and measures of data quality.
• Describe importance and measures of data availability.
Traditional Administration Definitions

• **Data Administration**: A high-level function that is responsible for the overall management of data resources in an organization, including maintaining corporate-wide definitions and standards.

• **Database Administration**: A technical function that is responsible for physical database design and for dealing with technical issues such as security enforcement, database performance, and backup and recovery.
Traditional Data Administration Functions

- Data policies, procedures, and standards.
- Planning.
- Data conflict (ownership) resolution.
- Managing the information repository.
  - Repositories contain the metadata that describes an organization’s data and data processing resources.
  - Information repositories are replacing data dictionaries in many organizations.
- Internal marketing of DA concepts.
Traditional Database Administration Functions

• Selection of DBMS and software tools.
• Installing/upgrading DBMS.
• Tuning database performance.
• Improving query processing performance.
• Managing data security, privacy, and integrity.
• Data backup and recovery.
Figure 12-1

Functions of DA and DBA

**Database planning**
- Develop corporate database strategy/policies
- Develop enterprise model (information architecture)
- Develop cost/benefit models
- Design database environment/select technologies
- Develop and market data administration plan

**Database analysis**
- Define and model data requirements (conceptual)
- Define and model business rules
- Define operational requirements
- Resolve requirements conflicts
- Maintain corporate data dictionary/repository

**Database design**
- Perform logical database design
- Design external model (subschemas)
- Design internal (physical) models
- Design integrity controls

**Database implementation**
- Specify database access policies
- Establish security controls
- Install DBMS
- Supervise database loading
- Specify test procedures
- Develop application programming standards
- Establish procedures for backup and recovery
- Conduct user training

**Operations and maintenance**
- Backup and recover databases
- Upgrade DBMS
- Monitor database performance
- Tune and reorganize databases
- Resolve access conflict
- Tune and rewrite queries
- Enforce standards and procedures
- Support users

**Growth and change**
- Implement change-control procedures
- Plan growth and change
- Evaluate new technology

**Function**
- DA = typically performed by data administration
- DBA = typically performed by database administration

Life-Cycle Phase

DA

DBA

DA/DBA
Evolving Approaches to Data Administration

• Blend data and database administration into one role.
• Fast-track development – monitoring development process (analysis, design, implementation, maintenance).
• Procedural DBAs–managing quality of triggers and stored procedures.
• eDBA–managing Internet-enabled database applications.
• PDA DBA–data synchronization and personal database management.
• Data warehouse administration.
Data Warehouse Administration

• New roles, coming with the growth in data warehouses.
• Similar to DA/DBA roles.
• Emphasis on integration and coordination of metadata/data across many data sources.
• Specific roles:
  – Support DSS applications.
  – Manage data warehouse growth.
  – Establish service level agreements regarding data warehouses and data marts.
Open Source DBMSs

• An alternative to proprietary packages such as Oracle, Microsoft SQL Server, or Microsoft Access

• mySQL is an example of open-source DBMS

• Less expensive than proprietary packages

• Source code available, for modification
Figure 12-2 Data modeling responsibilities
Database Security

- **Database Security**: Protection of the data against accidental or intentional loss, destruction, or misuse.
- Increased difficulty due to Internet access and client/server technologies.
Figure 12-3 Possible locations of data security threats
Threats to Data Security

• Accidental losses attributable to:
  – Human error
  – Software failure
  – Hardware failure

• Theft and fraud

• Improper data access:
  – Loss of privacy (personal data)
  – Loss of confidentiality (corporate data)

• Loss of data integrity

• Loss of availability (through, e.g. sabotage)
Figure 12-4 Establishing Internet Security
Web Security

• Static HTML files are easy to secure
  – Standard database access controls
  – Place Web files in protected directories on server

• Dynamic pages are harder
  – Control of CGI scripts
  – User authentication
  – Session security
  – SSL for encryption
  – Restrict number of users and open ports
  – Remove unnecessary programs
W3C Web Privacy Standard

• Platform for Privacy Protection (P3P)

• Addresses the following:
  – Who collects the data.
  – What data is collected and for what purpose.
  – Who shares the data.
  – Can users control access to their data.
  – How are disputes resolved.
  – Policies for retaining data.
  – Where are policies kept and how can they be accessed.
Database Software Security Features

- Views or subschemas
- Integrity controls
- Authorization rules
- User-defined procedures
- Encryption
- Authentication schemes
- Backup, journalizing, and checkpointing
Views and Integrity Controls

• Views
  – Subset of the database that is presented to one or more users
  – User can be given access privilege to view without allowing access privilege to underlying tables

• Integrity Controls
  – Protect data from unauthorized use
  – Domains—set allowable values
  – Assertions—enforce database conditions
Authorization Rules

- Controls incorporated in the data management system
- **Restrict:**
  - access to data
  - actions that people can take on data
- **Authorization matrix for:**
  - Subjects
  - Objects
  - Actions
  - Constraints
Figure 12-5 Authorization matrix

<table>
<thead>
<tr>
<th>Subject</th>
<th>Object</th>
<th>Action</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Dept.</td>
<td>Customer record</td>
<td>Insert</td>
<td>Credit limit LE $5000</td>
</tr>
<tr>
<td>Order trans.</td>
<td>Customer record</td>
<td>Read</td>
<td>None</td>
</tr>
<tr>
<td>Terminal 12</td>
<td>Customer record</td>
<td>Modify</td>
<td>Balance due only</td>
</tr>
<tr>
<td>Acctg. Dept.</td>
<td>Order record</td>
<td>Delete</td>
<td>None</td>
</tr>
<tr>
<td>Ann Walker</td>
<td>Order record</td>
<td>Insert</td>
<td>Order aml LT $2000</td>
</tr>
<tr>
<td>Program AR4</td>
<td>Order record</td>
<td>Modify</td>
<td>None</td>
</tr>
</tbody>
</table>
Some DBMSs also provide capabilities for **user-defined procedures** to customize the authorization process.
**Encryption** – the coding or scrambling of data so that humans cannot read them.

Secure Sockets Layer (SSL) is a popular encryption scheme for TCP/IP connections.
Authentication Schemes

• Goal – obtain a positive identification of the user

• Passwords: First line of defense
  – Should be at least 8 characters long
  – Should combine alphabetic and numeric data
  – Should not be complete words or personal information
  – Should be changed frequently
Authentication Schemes (cont.)

• Strong Authentication
  – Passwords are flawed:
    • Users share them with each other
    • They get written down, could be copied
    • Automatic logon scripts remove need to explicitly type them in
    • Unencrypted passwords travel the Internet

• Possible solutions:
  – Two factor—e.g. smart card plus PIN
  – Three factor—e.g. smart card, biometric, PIN
  – Biometric devices—use of fingerprints, retinal scans, etc. for positive ID
  – Third-party mediated authentication—using secret keys, digital certificates
Security Policies and Procedures

• Personnel controls
  – Hiring practices, employee monitoring, security training

• Physical access controls
  – Equipment locking, check-out procedures, screen placement

• Maintenance controls
  – Maintenance agreements, access to source code, quality and availability standards

• Data privacy controls
  – Adherence to privacy legislation, access rules
Database Recovery

- Mechanism for restoring a database quickly and accurately after loss or damage

- Recovery facilities:
  - Backup Facilities
  - Journalizing Facilities
  - Checkpoint Facility
  - Recovery Manager
Back-up Facilities

• Automatic dump facility that produces backup copy of the entire database
• Periodic backup (e.g. nightly, weekly)
• Cold backup–database is shut down during backup
• Hot backup–selected portion is shut down and backed up at a given time
• Backups stored in secure, off-site location
Journalizing Facilities

• Audit trail of transactions and database updates
• Transaction log—record of essential data for each transaction processed against the database
• Database change log—images of updated data
  – Before-image—copy before modification
  – After-image—copy after modification

Produces an *audit trail*
From the backup and logs, databases can be restored in case of damage or loss.
Checkpoint Facilities

- DBMS periodically refuses to accept new transactions
- System is in a quiet state
- Database and transaction logs are synchronized

This allows recovery manager to resume processing from short period, instead of repeating entire day
Recovery and Restart Procedures

• Disk Mirroring—switch between identical copies of databases.
• Restore/Rerun—reprocess transactions against the backup.
• Transaction Integrity—commit or abort all transaction changes.
• Backward Recovery (Rollback)—apply before images.
• Forward Recovery (Roll Forward)—apply after images (preferable to restore/rerun).
Transaction ACID Properties

• Atomic
  – Transaction cannot be subdivided

• Consistent
  – Constraints don’t change from before transaction to after transaction

• Isolated
  – Database changes not revealed to users until after transaction has completed

• Durable
  – Database changes are permanent
Figure 12-10 Basic recovery techniques
a) Rollback
Figure 12-10 Basic recovery techniques (cont.)

b) Rollforward
Database Failure Responses

- **Aborted transactions**
  - Preferred recovery: rollback
  - Alternative: Rollforward to state just prior to abort

- **Incorrect data**
  - Preferred recovery: rollback
  - Alternative 1: rerun transactions not including inaccurate data updates
  - Alternative 2: compensating transactions

- **System failure (database intact)**
  - Preferred recovery: switch to duplicate database
  - Alternative 1: rollback
  - Alternative 2: restart from checkpoint

- **Database destruction**
  - Preferred recovery: switch to duplicate database
  - Alternative 1: rollforward
  - Alternative 2: reprocess transactions
Concurrency Control

• **Problem**—in a multi-user environment, simultaneous access to data can result in interference and data loss.

• **Solution**—Concurrency Control
  
  – The process of managing simultaneous operations against a database so that data integrity is maintained and the operations do not interfere with each other in a multi-user environment.
Simultaneous access causes updates to cancel each other.

A similar problem is the **inconsistent read** problem.
Concurrency Control Techniques

• Serializability
  – Finish one transaction before starting another

• Locking Mechanisms
  – The most common way of achieving serialization
  – Data that is retrieved for the purpose of updating is locked for the updater
  – No other user can perform update until unlocked
Figure 12-12: Updates with locking (concurrency control)

This prevents the lost update problem
Locking Mechanisms

• Locking level:
  – Database—used during database updates
  – Table—used for bulk updates
  – Block or page—very commonly used
  – Record—only requested row; fairly commonly used
  – Field—requires significant overhead; impractical

• Types of locks:
  – Shared lock—Read but no update permitted. Used when just reading to prevent another user from placing an exclusive lock on the record
  – Exclusive lock—No access permitted. Used when preparing to update
Deadlock

- An impasse that results when two or more transactions have locked common resources, and each waits for the other to unlock their resources.

Figure 12-13
The problem of deadlock

John and Marsha will wait forever for each other to release their locked resources!
Managing Deadlock

• Deadlock prevention:
  – Lock all records required at the beginning of a transaction
  – Two-phase locking protocol
    • Growing phase
    • Shrinking phase
  – May be difficult to determine all needed resources in advance

• Deadlock Resolution:
  – Allow deadlocks to occur
  – Mechanisms for detecting and breaking them
    • Resource usage matrix
Versioning

- Optimistic approach to concurrency control
- Instead of locking
- Assumption is that simultaneous updates will be infrequent
- Each transaction can attempt an update as it wishes
- The system will reject an update when it senses a conflict
- Use of rollback and commit for this
Figure 12-15 The use of versioning

Better performance than locking
Managing Data Quality

• Causes of poor data quality
  – External data sources
  – Redundant data storage
  – Lack of organizational commitment

• Data quality improvement
  – Perform data quality audit
  – Establish data stewardship program (data steward is a liaison between IT and business units)
  – Apply total quality management (TQM) practices
  – Overcome organizational barriers
  – Apply modern DBMS technology
  – Estimate return on investment
Data Dictionaries and Repositories

• Data dictionary
  – Documents data elements of a database

• System catalog
  – System-created database that describes all database objects

• Information Repository
  – Stores metadata describing data and data processing resources

• Information Repository Dictionary System (IRDS)
  – Software tool managing/controlling access to information repository
Figure 12-16 Three components of the repository system architecture

- **Information Model**: A schema of the repository information
- **Repository Engine**: Software that manages the repository objects
- **Repository Database**: Where repository objects are stored

- Objects
- Relationships
- Extensible Types
- Version & Configuration Management
Database Performance Tuning

- **DBMS Installation**
  - Setting installation parameters

- **Memory Usage**
  - Set cache levels
  - Choose background processes

- **Input/Output (I/O) Contention**
  - Use striping
  - Distribution of heavily accessed files

- **CPU Usage**
  - Monitor CPU load

- **Application tuning**
  - Modification of SQL code in applications
Data Availability

• Downtime is expensive
• How to ensure availability
  – Hardware failures—provide redundancy for fault tolerance
  – Loss of data—database mirroring
  – Maintenance downtime—automated and nondisruptive maintenance utilities
  – Network problems—careful traffic monitoring, firewalls, and routers