Data Security, Data Administration and Database Administration

University of California, Berkeley

School of Information

INFO 257: Database Management

Announcements

- Questions?
- Assignment 4 Released (Project)
- Lecture Slides Upload
- Group Meeting after lecture

Lecture Outline

- Review
  - Database Administration: Security
- Database Administration: Disasters, Backup and Recovery
- Database Administration: Roles

Lecture Outline

- Database Administration: Data Integrity and Security
- Database Administration: Disasters, Backup and Recovery
- Database Administration: Roles

Data Integrity

- Intrarecord integrity (enforcing constraints on contents of fields, etc.)
- Referential Integrity (enforcing the validity of references between records in the database)
- Concurrency control (ensuring the validity of database updates in a shared multiuser environment)

Integrity Constraints (review)

- The constraints we wish to impose in order to protect the database from becoming inconsistent.
- Five types
  - Required data
  - attribute domain constraints
  - entity integrity
  - referential integrity
  - enterprise constraints

Referential Integrity

- Ensures that dependent relationships in the data are maintained. In Oracle, for example:
- CREATE TABLE table-name (

  attr1 attr-type PRIMARY KEY,

  attr2 attr-type NOT NULL,

…, attrM attr-type REFERENCES owner.tablename(attrname) *ON DELETE CASCADE*, …

Concurrency Control

- The goal is to support access by multiple users to the same data, at the same time
- It must assure that the transactions are *serializable* and that they are *isolated*
- It is intended to handle several problems in an uncontrolled system
- Specifically:
  - Lost updates
  - Inconsistent data states during access
  - Uncompleted (or committed) changes to data

Concurrency Control: Locking

- Locking levels
  - Database
  - Table
  - Block or page
  - Record
  - Field
- Types
  - Shared (S locks)
  - Exclusive (X locks)

Transaction Control in ORACLE

- Transactions are sequences of SQL statements that ORACLE treats as a "logical unit of work"
  - From the user's point of view a private copy of the database is created for the duration of the transaction
- Transactions are started with SET TRANSACTION, followed by the SQL statements
- Any changes made by the SQL are made permanent by COMMIT
- Part or all of a transaction can be undone using ROLLBACK

Transactions in MySQL

- START TRANSACTION or BEGIN starts a transaction block (disables autocommit)
- COMMIT or ROLLBACK will commit the transaction block or return to state before the block was started
- MySQL may use different underlying database engines – the InnoDB engine also supports SAVEPOINT and ROLLBACK TO SAVEPOINT

- NOTE: This syntax can be used in any of MySQL's database engines - but it only **WORKS** when using the InnoDB engine (which can be set up when the tables are created)

Transactions in MySQL (5.0+)

- START TRANSACTION [WITH CONSISTENT SNAPSHOT] | BEGIN [WORK]
- COMMIT [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
- ROLLBACK [WORK] [AND [NO] CHAIN] [[NO] RELEASE]
- SET AUTOCOMMIT = {0 | 1}

- The START TRANSACTION and BEGIN statement begin a new transaction. COMMIT commits the current transaction, making its changes permanent. ROLLBACK rolls back the current transaction, canceling its changes. The SET AUTOCOMMIT statement disables or enables the default autocommit mode for the current connection

Versioning

- Newer 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

Use of Versioning…

Better performance than locking

Database Security

- Views or restricted subschemas
- Authorization rules to identify users and the actions they can perform
- User-defined procedures (with rule systems or triggers) to define additional constraints or limitations in using the database
- Encryption to encode sensitive data
- Authentication schemes to positively identify a person attempting to gain access to the database

Views

- A subset of the database presented to some set of users
  - SQL:

    CREATE VIEW viewname AS SELECT field1, field2, field3,…, FROM table1, table2 WHERE <where clause>;
  - Note: “queries” in Access function as views

Restricted Views

- Main relation has the form:

Name	C_name	Dept	C_dept	Prof	C_prof	*TC*
J Smith	S	Dept1	S	Cryptography	TS	TS
M Doe	U	Dept2	S	IT Security	S	S
R Jones	U	Dept3	U	Secretary	U	U

U = unclassified : S = Secret : TS = Top Secret

Restricted Views

S-view of the data

NAME	Dept	Prof
J Smith	Dept1	---
M Doe	Dept2	IT Security
R Jones	Dept3	Secretary

U-view of the data

NAME	Dept	Prof
M Doe	---	---
R Jones	Dept3	Secretary

Authorization Rules

- Most current DBMS permit the DBA to define “access permissions” on a table by table basis (at least) using the GRANT and REVOKE SQL commands
- Some systems permit finer grained authorization (most use GRANT and REVOKE on variant views

MySQL Backup Types

- Physical (Raw) Versus Logical Backups
  - Physical (or Raw) Backups
    - Physical backups consist of raw copies of the directories and files that store database contents. This type of backup is suitable for large, important databases that need to be recovered quickly when problems occur.
  - Logical Backups
    - Logical backups save information represented as logical database structure (CREATE DATABASE, CREATE TABLE statements) and content (INSERT statements or delimited-text files). This type of backup is suitable for smaller amounts of data where you might edit the data values or table structure, or recreate the data on a different machine architecture.

From: http://dev.mysql.com/doc/refman/5.1/en/backup-types.html

Logical Backups

- Logical backup tools include the **mysqldump** program and the **SELECT ... INTO OUTFILE** statement. These work for any storage engine, even MEMORY.
- To restore logical backups, SQL-format dump files can be processed using the **mysql** client. To load delimited-text files, use the **LOAD DATA INFILE** statement or the **mysqlimport** client.

Physical Backups

- Physical backup tools include file system-level commands (such as cp, scp, tar, rsync), **mysqlhotcopy** for MyISAM tables, **ibbackup** for InnoDB tables, or **START BACKUP** for NDB tables.
- For restore, files copied at the file system level or with **mysqlhotcopy** can be copied back to their original locations with file system commands; **ibbackup** restores InnoDB tables, and **ndb_restore** restores NDB tables.

Lecture Outline

- Review
  - Database Administration: Security
- Database Administration: Disasters, Backup and Recovery
- Database Administration: Roles and Functions

Disasters come in many forms…

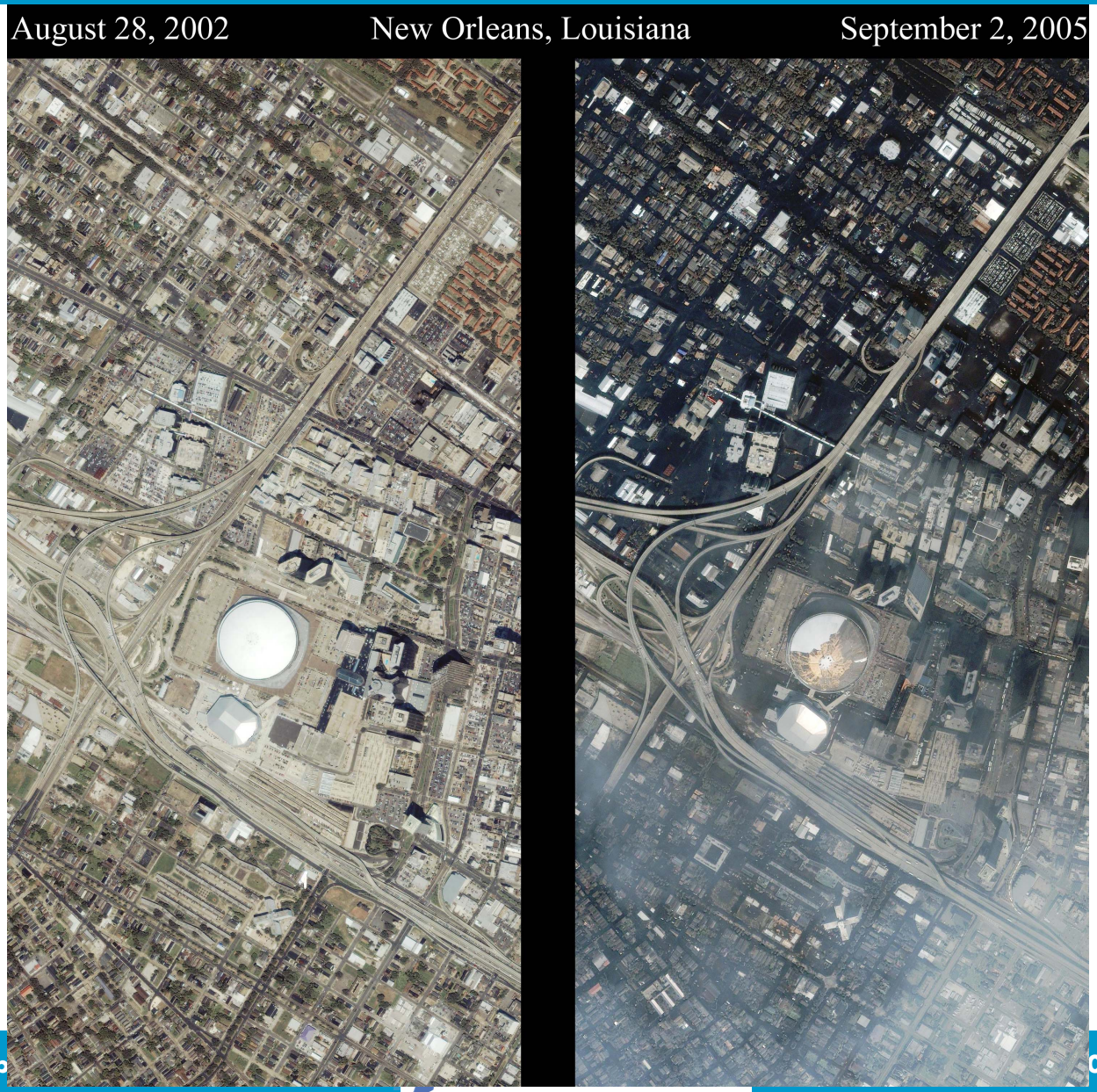
Pre Flood (Sept. 1992)

Peak Flood (Sept. 1993)

Post Flood (Oct. 1993)

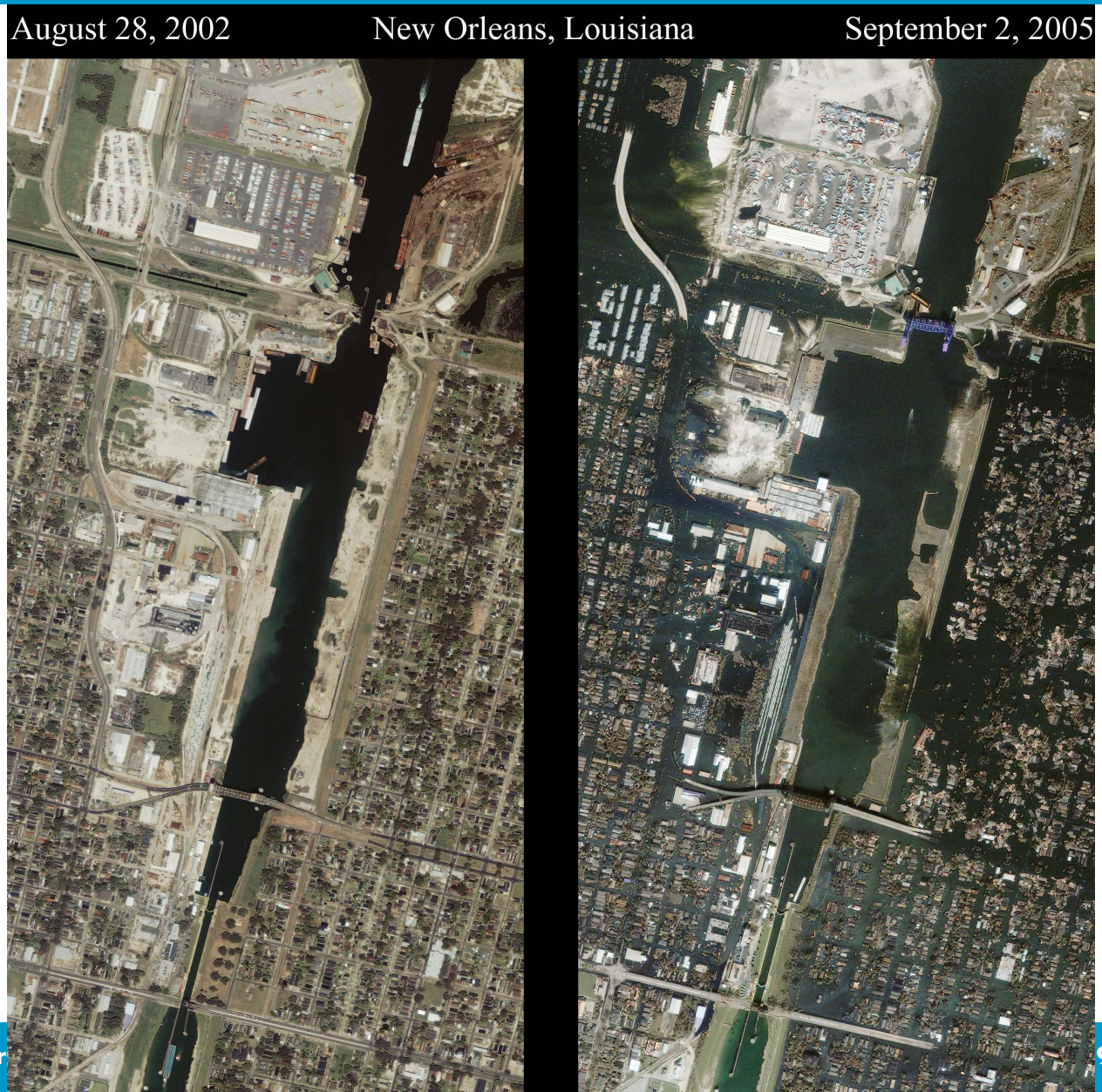
La Crosse, Wisc 2001

Katrina



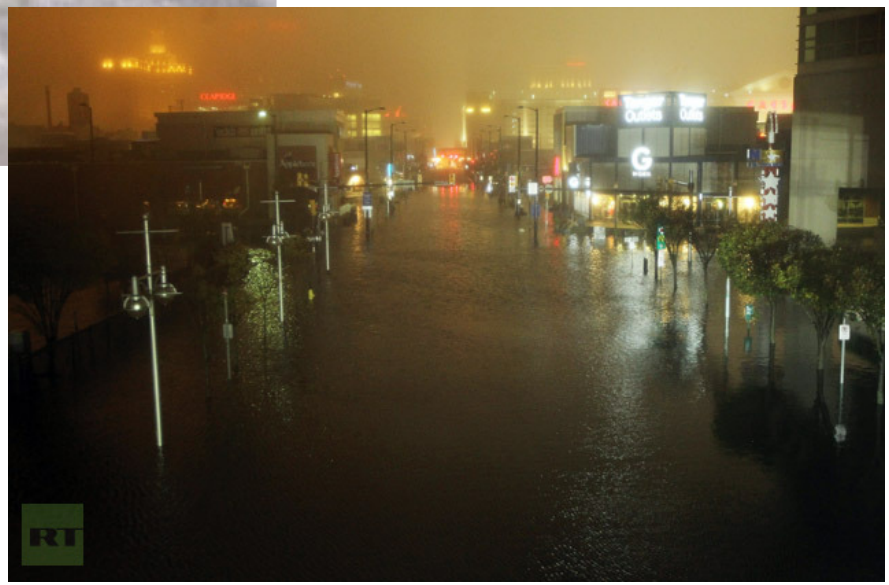
Katrina

Katrina

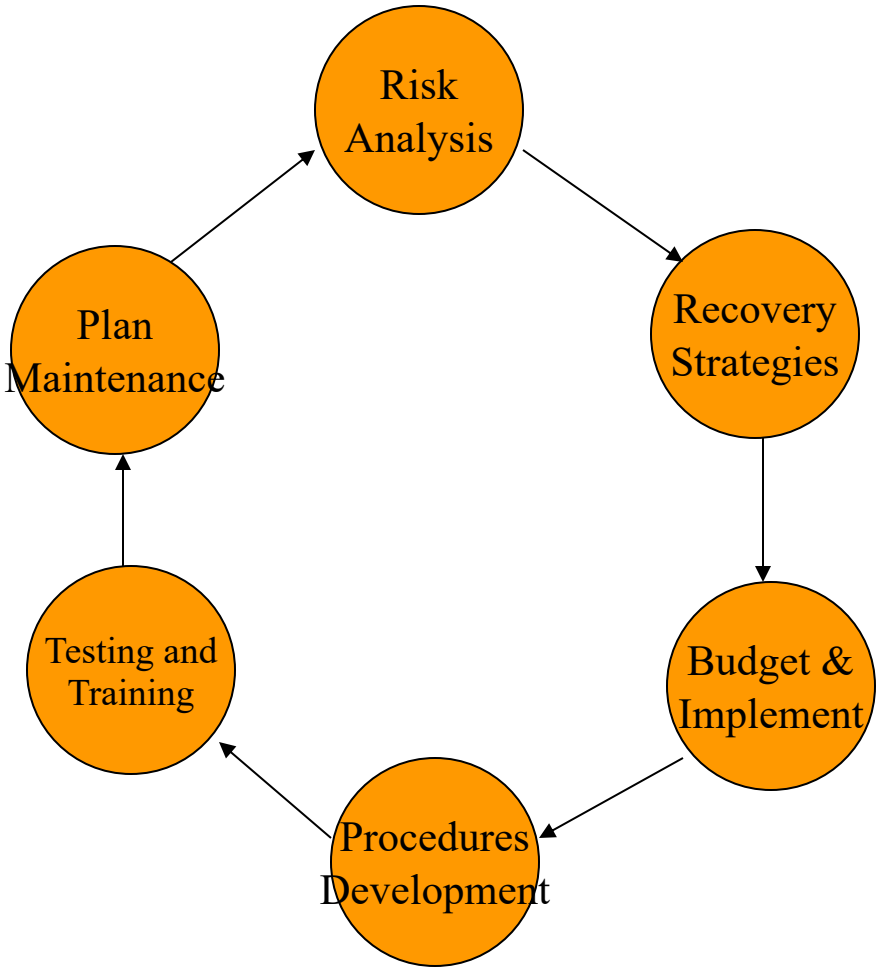


Katrina

Hurricane Sandy in N.J. & N.Y.



Disaster Recovery Planning



From Toigo "Disaster Recovery Planning"

Threats to Assets and Functions

- Water
- Fire
- Power Failure
- Mechanical breakdown or software failure
- Accidental or deliberate destruction of hardware or software
  - By hackers, disgruntled employees, industrial saboteurs, terrorists, or others

Threats

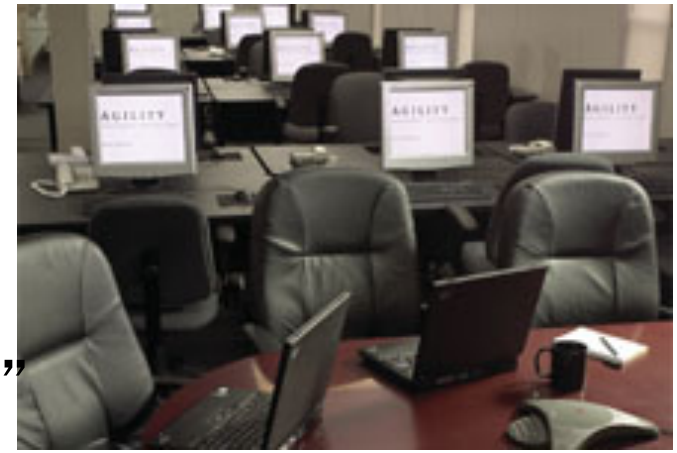
- Between 1967 and 1978 fire and water damage accounted for 62% of all data processing disasters in the U.S.
- The *water* damage was sometimes caused by fighting *fires*
- More recently improvements in fire suppression (e.g., Halon) for DP centers has meant that water is the primary danger to DP centers

Kinds of Records

- Class I: VITAL
  - Essential, irreplaceable or necessary to recovery
- Class II: IMPORTANT
  - Essential or important, but reproducible with difficulty or at extra expense
- Class III: USEFUL
  - Records whose loss would be inconvenient, but which are replaceable
- Class IV: NONESSENTIAL
  - Records which upon examination are found to be no longer necessary

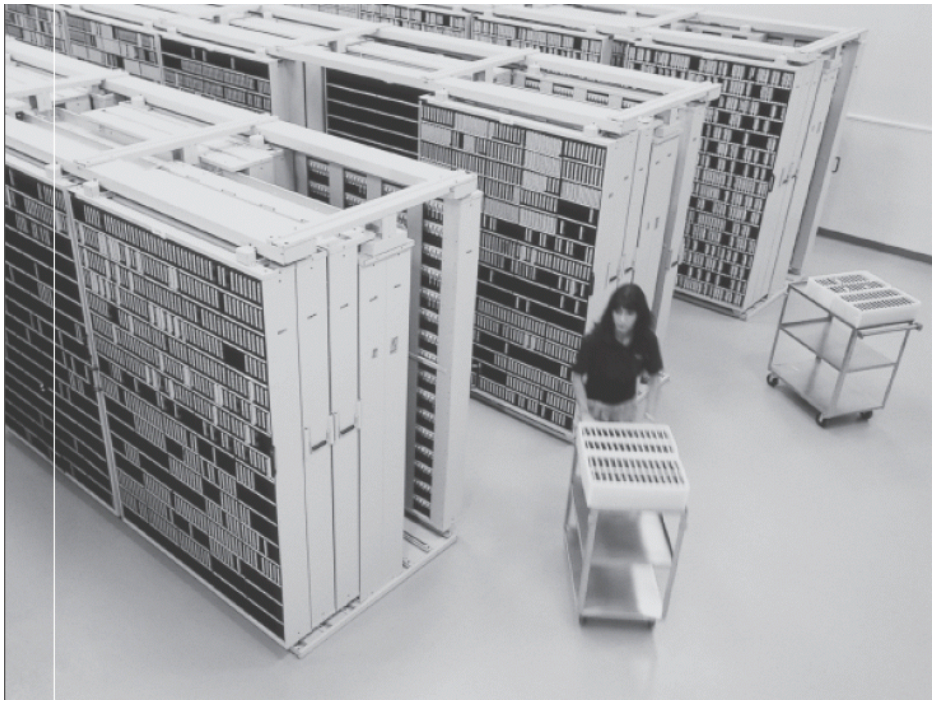
Offsite Storage of Data

- Early offsite storage facilities were often intended to survive atomic explosions
- PRISM International directory
  - PRISM = Professional Records and Information Services Management
  - http://www.prismintl.org/
- Mirror sites (Hot sites)



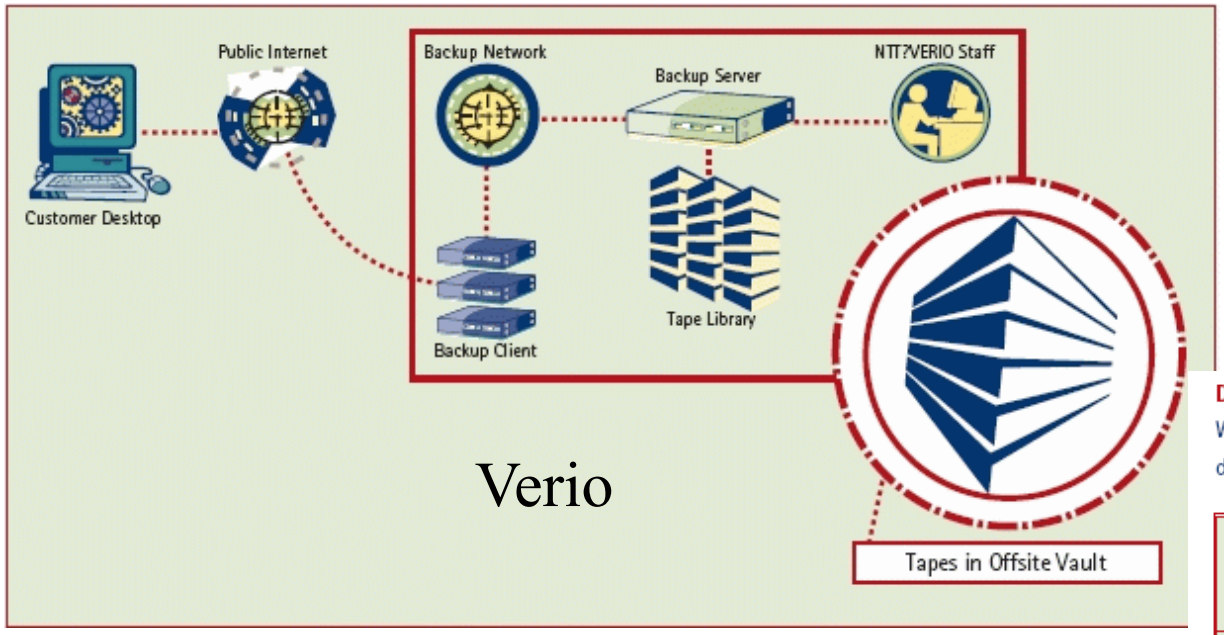
Agility "Hotsuite"

Offsite Storage Providers



Iron Mountain

Offsite backup providers



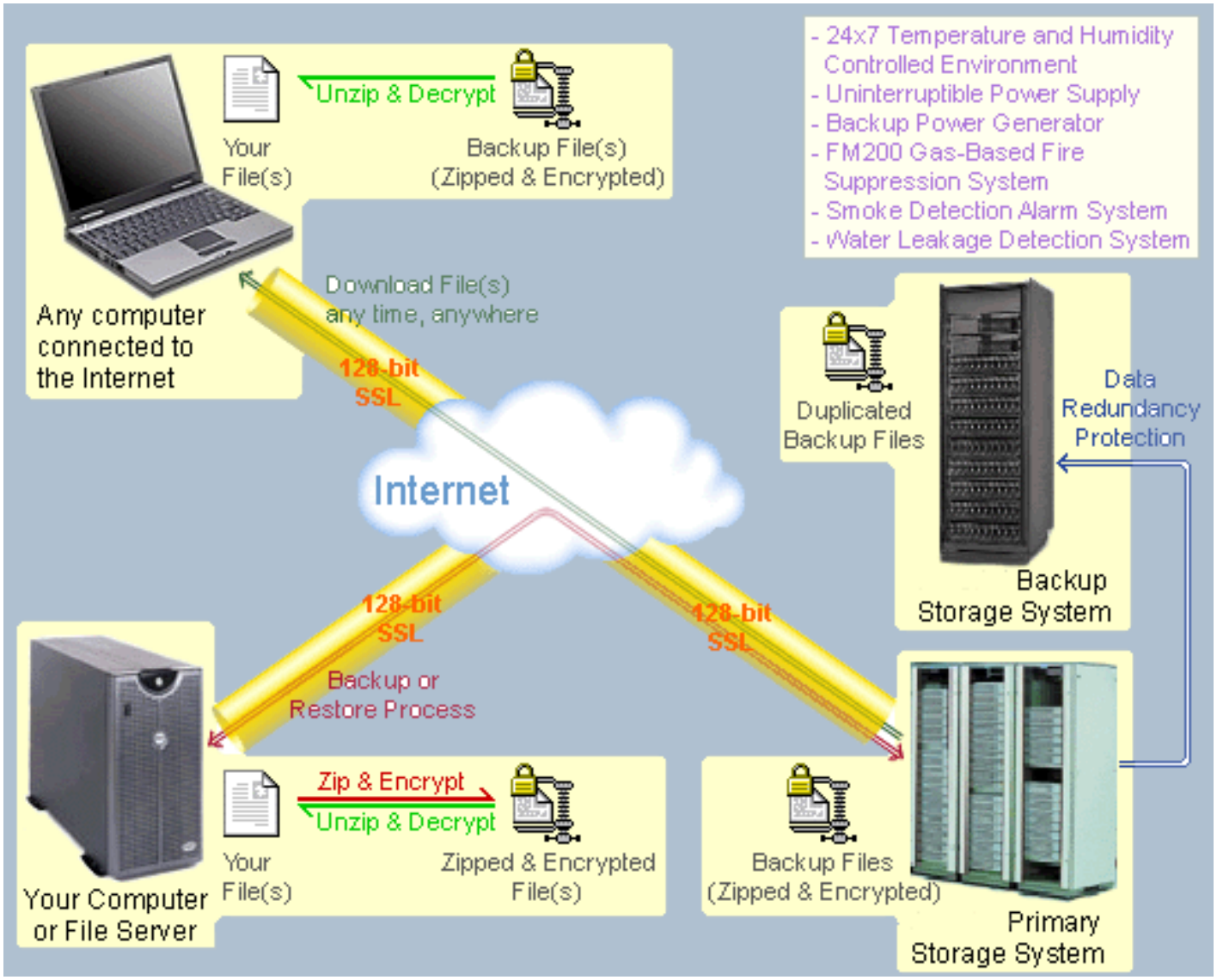
DATABASE BACKUP

We offer three database backup options to meet your needs for database availability:

OFFLINE DATABASE BACKUP	This service provides a "cold" backup option for all database types. The database (DBMS) platform is halted during the backup process and cannot be accessed until completed.
ONLINE DATABASE BACKUP	Online Database Backup offers "hot" database backups for Microsoft DBMS platforms and "warm" backups for all other database (DBMS) platforms. During an Online Database Backup, the database continues to function and serve requests, but will experience some degree of performance degradation. Six packages are available to backup from 35 GB to more than 800GB per month. Custom billing plans are available for backups over 1 TB per month.
MIRRORED ONLINE DATABASE BACKUP	Mirrored Online Database Backup offers "active online" database backups for Microsoft DBMS platforms and "hot" backups for all other database (DBMS) platforms. During a Mirrored Online Database Backup, the production database continues to function and serve requests, and best of all, will not experience any form of performance degradation. Six packages are available to backup from 35 GB to more than 800 GB per month. Custom billing plans are available for backups over 1 TB per month.

Backup and Offsite Backup

Found on the Web…
This is typical of services that provide offsite backup for computers or DP centers



Lecture Outline

- Review
  - Database Administration: Security
- Database Administration: Disasters, Backup and Recovery
- Database Administration: Roles

Today

- Traditional and Current Data Administration
- Traditional and Current Database Administration

Changes in Traditional Roles

- This is being driven by rapid changes in
  - Technology
  - Platforms (e.g., Micro vs. Mainframe vs. Server vs. Cloud)
  - Organizational Structure
- We will focus on the core functions and tasks of these roles (traditional or current)

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, standards
- Planning
- Data conflict (ownership) resolution
- Managing the information repository
- Internal marketing of DA concepts

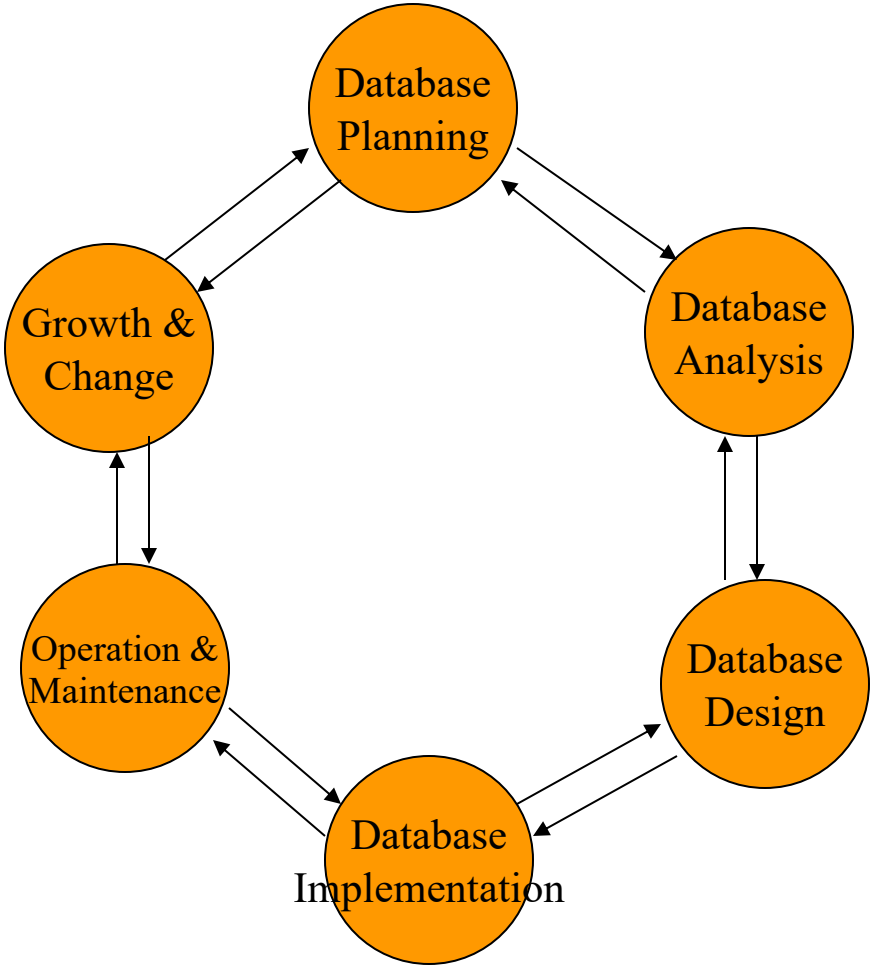
Traditional Database Administration Functions

- 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

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

Database System Life Cycle



Note: this is a different version of this life cycle than discussed previously

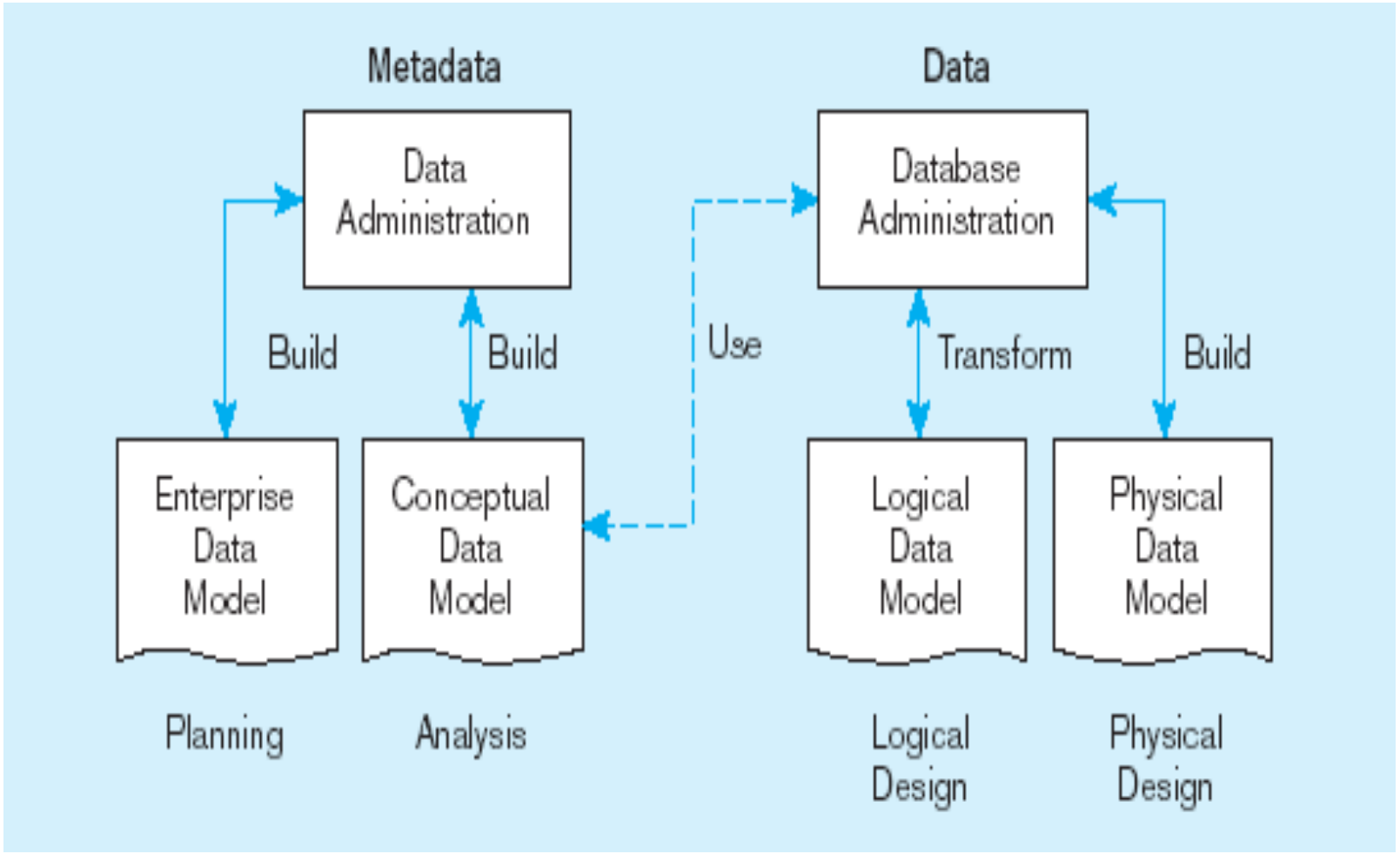
Database Planning

- Development of a strategic plan for database development that supports the overall organization's business plan
- DA supports top management in development of this plan
- The result of this stage is an *enterprise data model*

Database Design

- Purpose of the design phase is the development of the logical database design that will serve the needs of the organization and the physical design implementing the logical design
- In relational systems the outcome is normalized relations, and the data definition for a particular database systems (including indexes, etc.)

Roles for design process



Database Implementation

- Database design gives you an empty database
- Load data into the database structure
- Convert existing data sets and applications to use the new database
  - May need programs, conversion utilities to convert old data to new formats.
- Outcome is the actual database with its data

Database Implementation DA & DBA functions

- Specify database access policies (DA & DBA)
- Establish Security controls (DBA)
- Supervise Database loading (DBA)
- Specify test procedures (DBA)
- Develop application programming standards (DBA)
- Establish procedures for backup and recovery (DBA)
- Conduct User training (DA & DBA)

Operation and Maintenance 1: Operations

- Users are responsible for updating the database, DA and DBA are responsible for developing procedures that ensure the integrity and security of the database during the update process.
- Specific responsibility for data collection, editing and verification must be assigned
- Quality assurance must be practiced to protect and audit the database quality.

Operation and Maintenance 2: Maintenance

- The ongoing process of updating the database to keep it current
  - adding new records
  - deleting obsolete records
  - changing data values in particular records
  - modifying relation structures (e.g. adding new fields)
- Privacy, security, access control must be in place.
- Recovery and Backup procedures must be established and used

Operation and Maintenance: DA & DBA functions

- Monitor database performance (DBA)
- Tune and reorganize databases (DBA)
- Enforce standards and procedures (DBA)
- Support users (DA & DBA)

Data Warehouse Administration

- New role, 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

Growth & Change

- Change is a way of life
  - Applications, data requirements, reports, etc. will all change as new needs and requirements are found
  - The Database and applications and will need to be modified to meet the needs of changes to the organization and the environment
  - Database performance should be monitored to maintain a high level of system performance

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