

Data Administrator

Causes of Poor Data Utilization

- Multiple definitions of the same data entity and inconsistent representations of the same data elements in separate database, which makes linking data across different.
- Missing key data elements, which makes existing data useless.
- Low levels of data quality due to inappropriate sources of data or timing of data transfers from one system to another.
- Not knowing what data exist, where to find them, and what they really mean.

Data Administration

- A high-level function that is responsible for the overall management 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.

Functions of Data and Database Administration

- There are 6 stages in the life cycle of a typical database system:
 - ◆ Database planning
 - ◆ Database analysis
 - ◆ Database design
 - ◆ Database implementation
 - ◆ Operation and maintenance
 - ◆ Growth and change

Database Planning

- This develops a strategic plan for database development that supports the overall organizational business plan.
- This usually is the responsibility of top management.

Database Analysis

- The process of analysis is concerned with identifying data entities currently used by the organization and their relationships.

Database Design

- This develops a strategic plan for database development that supports the overall organization business plan.
- This usually is the responsibility of top management.

Database Implementation

- Developing customized software routines
- Distributing the system software and data
- Preparing system documentation

Operation and Maintenance

- This is a process to update the database to keep it current.

Growth and Change

- Data administrators must plan for change, such as adding new record types, accommodating growth.
- They must monitor the performance of the database and take corrective actions whenever necessary.

Database Management System

- A Database Management System (DBMS) is a software application system that is used to create, maintain, and to provide controlled access to user databases.

Components of DBMS

- DBMS Engine
- Interface Subsystem
- Information Repository Dictionary Subsystem
- Performance Management Subsystem
- Backup and Recovery Subsystem
- Application Development Subsystem
- Security Management Subsystem

DBMS Engine

- The central components of a DBMS
- Provides access to the repository and the database and coordinates all of the other functional elements of the DBMS.

Interface Subsystem

- Provides facilities for users and applications to access the various components of the DBMS.
- Most DBMS products provide a range of languages and other interfaces.
- The system is used by programmers and by users with little or no programming experience.

Information Repository Dictionary Subsystem

- This is also known as the Data Dictionary
- Used to manage and control access to the repository.

Performance Management Subsystem

- Provides facilities to optimize DBMS performance.
 - ◆ Query optimization:
 - ◆ Structuring SQL queries to minimize response time.
 - ◆ DBMS reorganization:
 - ◆ Maintaining statistics on database usage and taking actions such as database reorganization, creating indexes.

Backup and Recovery Subsystem

- Logging transactions and database changes, periodically making backup copies of the database, and recovering the database in the event of some type of failure.

Application Development Subsystem

- Allow end users and programmers to develop complete database applications.

Security Management Subsystem

- Protect and control access to the database and repository.

Single-user vs. Multi-user Systems

- Single-User Environment
 - ◆ A DBMS is single-user if at most one user at a time can use the system.
- Multi-User Environment
 - ◆ A DBMS is multi-user if many users can use the system concurrently.

Transaction

- The execution of a program that accesses or changes the contents of the database is called a **Transaction**.

Why Concurrency Control is Needed?

- If the concurrent transactions is not controlled, it may lead to problems such as an inconsistent database.
 - ◆ The Lost Update Problem
 - ◆ The Temporary Update Problem
 - ◆ Inconsistent Analysis Problem

The Lost Update Problem

Transaction A	Time	Transaction B
1. Read account balance (Balance = \$1,000)	t1	1. Read account balance (Balance = \$1,000)
2. Update record (withdraw \$200 and the balance is \$800)	t2	2. Update record (withdraw \$300 and the balance is \$700)
	t3	
	t4	ERROR!

Transaction A's update is lost at time t4, because transaction B overwrites without even looking at it. This means that the effect of B's update has been lost due to interference between the transactions.

The Temporary Update Problem

Transaction 1 (T1)	Transaction 2 (T2)
Read item (X) $X = X - N$ Write item (X)	Read-item (X) $X = X + M$ Write-item (X)
Rollback	

Transaction T1 fails and must change the value of X back to its old value; but meanwhile, T2 has read the "temporary" incorrect value of X

The Inconsistent Analysis Problem

Transaction 1 (T1)	Transaction 2 (T2)
read_item(X) $X = X - N$ write_item(X)	sum = 0 read_item(A) sum = sum + A . . . read_item(X) sum = sum + X read_item(Y) sum = sum + Y
read_item(Y) $Y = Y + N$ write_item(Y)	

T2 reads X after N is subtracted and reads Y before N is added, so a wrong summary is resulted

Basic Approaches to Concurrency Control

- Concurrency control is concerned with preventing loss of data integrity due to interference between users in a multi-user environment.
- Two basic approaches to concurrency control:
 - ◆ A Pessimistic Approach
 - ◆ An Optimistic Approach

Pessimistic Approach - Locking

- The most common type of concurrency control mechanism.
- Any data that is retrieved by a user for updating must be locked, or denied to other user, until the update is completed.
- Locking data is most like checking a book out of the library.
- It is unavailable to other until it is returned by the borrower.

Shared Locks

- Also called **S Locks**, or **Read Locks**
- Allow other transaction to read (but not update) a record (or other resource)
- A transaction should place a shared lock on a record when it will only read (but not update) that record.
- With a shared lock, it prevents another user from placing an exclusive lock on that record.

Exclusive Locks

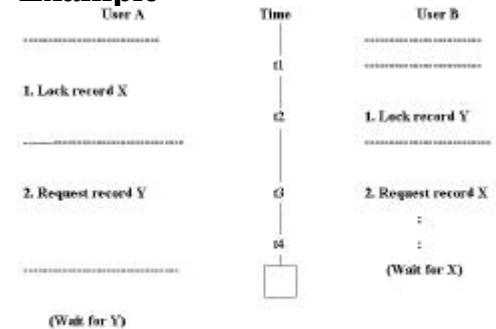
- Also called X locks, or write locks
- Prevent another transaction from reading (and therefore updating) a record until it is unlocked.
- A transaction should place an exclusive lock on a record when it is about to update that record.
- With an exclusive lock, it prevents another user from placing any type of locking on that record.

	Shared Lock(S lock)	Exclusive Lock (X lock)
Shared Lock	True	False
Exclusive Lock	False	True

Pessimistic Approach - Managing Deadlock

- Locking (say at the record level) solves the problem of erroneous updates but may lead to another, called deadlock.
- Occurs when two (or more) transaction have locked a common resource and each must wait for the other to unlock the resource.

Example



(Wait for Y)

User A has locked record X and user B has locked record Y. Then User A requests record Y and user B requests record X. Both requests are denied, since the requested records are already locked. Thus, unless the DBMS intervenes, both users will wait indefinitely.

Managing Deadlock

- Deadlock Prevention
 - ◆ When deadlock prevention is employed, user programs must lock all records they will require at the beginning of a transaction (rather than one at a time).
- Deadlock Resolution
 - ◆ This allows deadlocks to occur but build mechanisms into the DBMS for detecting and breaking the deadlocks.

Pessimistic Approach - Optimistic Approach (Versioning)

- Most of the time users do not want the same record, or only want to read the record.
- With Versioning, there is no form of locking.
- Each transaction is treated as a view of the database as when the transaction starts.
- When transaction modifies a record, the DBMS creates a new record version instead of overwriting the old record.
- If there is no conflict, this user's changes are used to update the central database.
- The main advantage of versioning over locking is performance improvement as read-only transactions can run concurrently with updating transaction.

Example

- User A reads the record containing the account balance, successfully withdraws \$200 and the new balance \$800 is posted the account with a COMMIT statement.
- Meanwhile, user B has also read the account record and requested a withdrawal. This is posted to her local version of the account record.
- Therefore, when the transaction attempts to COMMIT, it discovers the update conflict and her transaction is aborted. The transaction can be restarted later with the correct balance of \$800.

Types of Failures

- Computer Failure (System Crash)
 - ◆ A hardware or software error occurs in the computer system during transaction execution.
 - ◆ If the hardware crashes, the contents of the computer internal memory may be lost.
- A Transaction or System Error
 - ◆ Some operation in the transaction may cause it to fail, such as integer overflow or division by zero.

Types of Failures (cont')

- Disk Failure
 - ◆ Some disk blocks may lose their data because of a read or write malfunction or because of a disk read/write head crash.
 - ◆ This may happen during a read or write operation of the transaction.
- Physical Problems and Catastrophes
 - ◆ This is an endless list that includes power or air conditioning failure, fire, theft, sabotage, overwriting disks or tapes by mistake etc.

Database Recovery

- Database recovery means restoring a database quickly and accurately after loss and damage.
- The basic recovery facilities includes:
 - ◆ Backup Facility
 - ◆ Journalizing Facilities
 - ◆ Checkpoint Facility
 - ◆ Recovery Manager

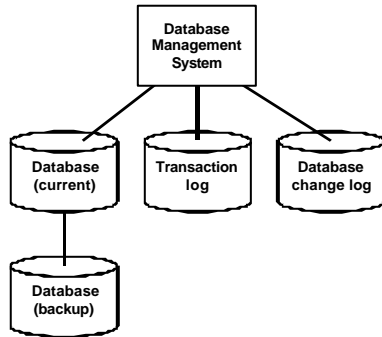
Backup Facility

- Provide periodic backup copies of the entire database.
- The copy should be stored in a secured location where it is protected from loss or damaged.

Journalizing Facilities

- Maintain an audit of transactions and database changes.
- There are transaction log and database change log.
 - ◆ Transaction log contains a record of the essential data for each transaction that is processed against the database.
 - ◆ Database change log contains before- and after-images of records that have been modified by transactions.

Journalizing Facilities



DB212 © Peter Lo 2004

41

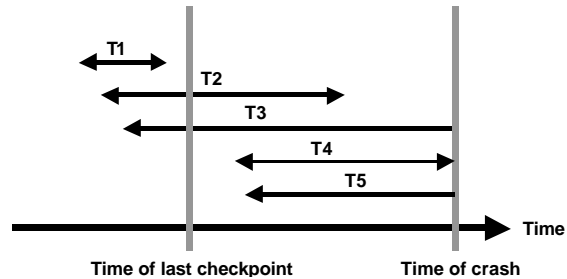
Checkpoint Facility

- The DBMS periodically suspends all processing and synchronizes its files and journals
- Checkpoints should be taken frequently.
- When failures do occurs, it is often possible to resume processing from the most recent checkpoint.
- Only a few minutes of processing work must be repeated.

DB212 © Peter Lo 2004

42

Example



DB212 © Peter Lo 2004

43

Explanation of Example

- T1 was completed before the last checkpoint, so it will not be listed in the checkpoint log record and will have no records in the log subsequent to the last checkpoint.
- T2 was currently active at the time of the last checkpoint so it will also have a COMMIT or ABORT log record in the log file subsequent to the last checkpoint.
- T3 is also listed in the checkpoint record, but it has not completed by the time of the failure, so it has no COMMIT or ABORT record in the log.
- T4 was executed fully between the time of the last checkpoint and the crash, so it has both a BEGIN TRANSACTION and a COMMIT or ABORT record in the log, subsequent to the last check-point record.
- T5 was was begun after the checkpoint, but not completed. It therefore has a BEGIN TRANSACTION, but no COMMIT or ABORT record, in the log subsequent to the last checkpoint.

DB212 © Peter Lo 2004

44

Explanation of Example

- Therefore, at the time of crash, transaction T3 and T5 effects have to be undone, since they are incomplete transaction. Transactions of type T1 has no problems, since they are known to have completed and their updates are known to have been consolidated on the databases at the time of the last checkpoint. Transaction of type T2 and T4 normally present no problem but it is not known whether all the necessary updates have been carried out on the database (some changed pages may still be in the buffers and consequently been lost). Thus the system will have to check whether a complete updates are done. If not, all the updates are undone, else if completed (commit), all updates are redone.
- In short, this means redoing the effects of a transaction which had committed before the crash, but after the last checkpoint; as well as undoing the effects of the incomplete transactions at the point of crash.

Recovery Manager

- Allows the DBMS to restore the database to a correct condition and restart processing transactions.

Database Security

- Protection of the database against accidental or intentional loss, destruction or misuse.
- Data administration uses several facilities provided by data management software in carrying out these functions.
 - ◆ Views or subschemas
 - ◆ Authorization Rules
 - ◆ User-defined Procedures
 - ◆ Encryption Procedures
 - ◆ Authentication Schemas

Views or Subschemas

- Help to restrict user views of the database.
 - ◆ Example:

```
CREATE VIEW ITEM-ORDER
  AS SELECT ITEM-NAME, ORDER-NO
  FROM ITEM, ORDER
  WHERE ITEM.ORDER-NO = ORDER.ORDER-NO;
```

Authorization Rules

- Identify users and restrict the actions they may take against the database.
 - ◆ Example
 - ◆ Using of password.

User-defined Procedures

- Defines additional constraints or limitations in using the database.
 - ◆ Example
 - ◆ User implements their password logging in their own PC.

Encryption Procedures

- encodes data in an unrecognizable form.
 - ◆ Example,
 - ◆ In the electronic funds transfer systems. The encryption procedures should also include decoding facility.

Authentication schemas

- Positively identify a person attempting to gain access to a database.

Review Questions

1. Contrast the following terms:
 - a) data administration vs. database administration
 - b) deadlock prevention vs. deadlock resolution
 - c) optimistic concurrency control vs. pessimistic concurrency control
 - d) shared locks vs. exclusive locks
2. Describe the DBMS facilities that are required for database backup and recovery.

Review Questions

3. For each of the situations describe below, indicate which of the following security measures is most important appropriate:
 1. authorization rules
 2. encryption
 3. authentication schemes
- a. A national brokerage firm uses a simple password system to protect its database but finds it needs a more comprehensive system to grant different privileges (such as read versus create or update) to different users.

Review Questions

- b. A manufacturing firm uses a simple password system to protect its database but finds it needs a more comprehensive system to grant different privileges (such as read versus create or update) to different users.
- c. A university has experienced considerable difficulty with unauthorized users who access files and databases by appropriating passwords from legitimated users.