CMPS1134
Fundamentals of Computing

Database Systems 2

Computer Science: An Overview
Eleventh Edition
J. Glenn Brookshear
Chapter 9

Chapter 9: Database Systems

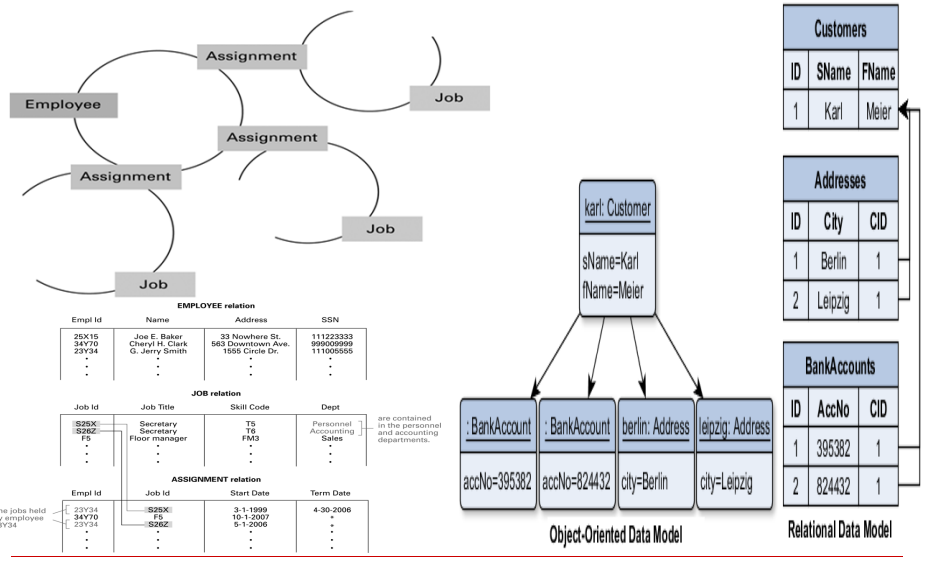
- Object-Oriented Databases
- Maintaining Database Integrity
- Traditional File Structures
- Data Mining
- Social Impact of Database Technology

Object-oriented Databases

- **Object-oriented Database:** A database constructed by applying the object-oriented paradigm
  - Each entity stored as a persistent object
  - Relationships indicated by links between objects
  - DBMS maintains inter-object links

Object-oriented Databases

The associations between objects in an object-oriented database (Fig 9.13)



Object-oriented Databases

Advantages of Object-oriented Databases

- Matches design paradigm of object-oriented applications. Avoid issues of trying to match imperative style relational databases with OOP applications
- Intelligence can be built into attribute handlers i.e. objects can contain methods
- Can handle exotic data types
  - Example: multimedia encapsulated in objects

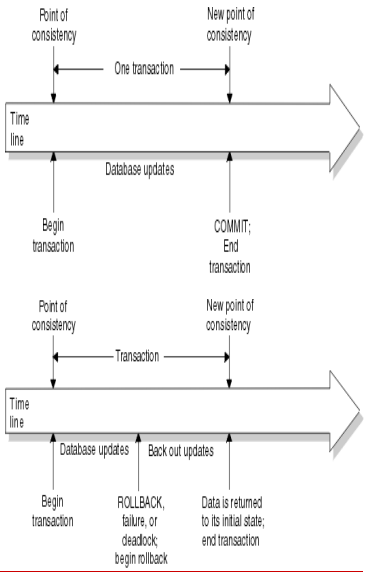
Maintaining Database Integrity

- Database systems for **personal** use are relatively simple systems
  - Loss or corruption would be inconvenient rather than disastrous
- Large, multi-user, **commercial** database systems are complex systems that are core to the operations of organizations
  - The cost of incorrect or loss data can be enormous and can have devastating consequences.
  - A major role of the DBMS is to maintain the **Database Integrity**

Maintaining Database Integrity

Commit/ Rollback Protocol

- **Transaction:** A sequence of operations that must all happen together
  - Example: transferring money between bank accounts
- **Transaction log:** A non-volatile record of each transaction's activities, built before the transaction is allowed to execute
  - **Commit point:** The point at which a transaction has been recorded in the log
  - **Roll-back:** The process of undoing a transaction



Maintaining Database Integrity

Locking

- Simultaneous access problems
  - Incorrect summary problem
  - Lost update problem
- **Locking** = preventing others from accessing data being used by a transaction
  - **Shared** lock: used when reading data
  - **Exclusive** lock: used when altering data

Traditional File Structures

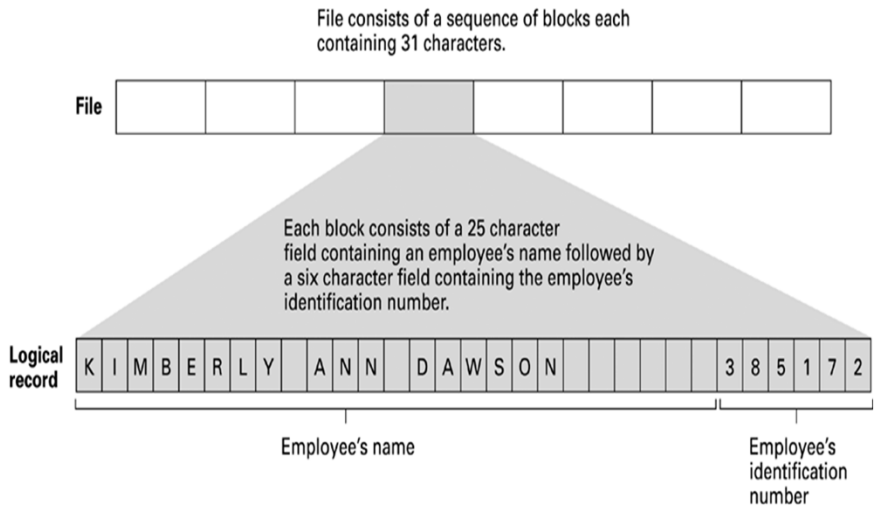
- These structures :
  - Represent the historical beginning of data storage and data retrieval systems from which current database technology has evolved
  - Many of the techniques developed for these structures are important tools in the construction of today's massive, complex databases – such as:
    - **Indexing**
    - **Hashing**

Sequential Files

- **Sequential file:** A file whose contents can only be read in order
  - Reader must be able to detect end-of-file (EOF)
  - Data can be stored in logical records, sorted by a key field
    - Greatly increases the speed of batch updates

Traditional File Structures

The structure of a simple employee file implemented as a text file (Fig 9.14)



Traditional File Structures

Procedure for merging 2 sequential files (Fig 9.15)

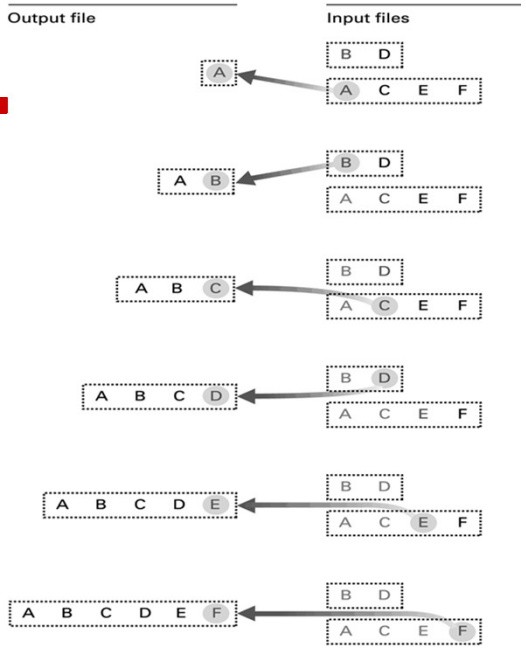
```
procedure MergeFiles (InputFileA, InputFileB, OutputFile)

if (both input files at EOF) then (Stop, with OutputFile empty)
if (InputFileA not at EOF) then (Declare its first record to be its current record)
if (InputFileB not at EOF) then (Declare its first record to be its current record)
while (neither input file at EOF) do
    (Put the current record with the "smaller" key field value in OutputFile;
     if (that current record is the last record in its corresponding input file)
         then (Declare that input file to be at EOF)
          else (Declare the next record in that input file to be the file's current record)
    )
Starting with the current record in the input file that is not at EOF,
     copy the remaining records to OutputFile.
```


Traditional File Structures

Applying the merge algorithm (Fig 9.16)

Letters
are used to represent
entire records.

The particular letter
indicates the value
of the record's
key field.



 13

Traditional File Structures

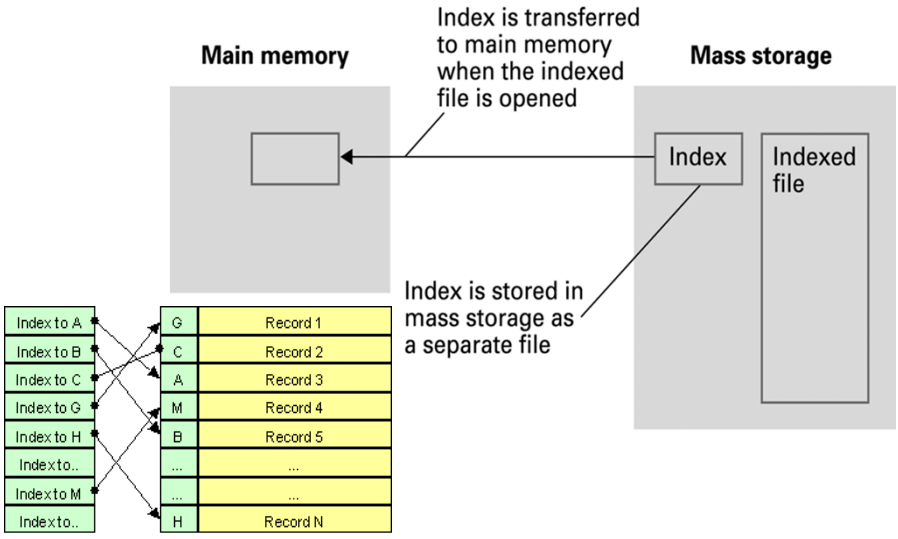
Indexed Files

- **Index:** A list of key values and the location of their associated records

 14

Traditional File Structures

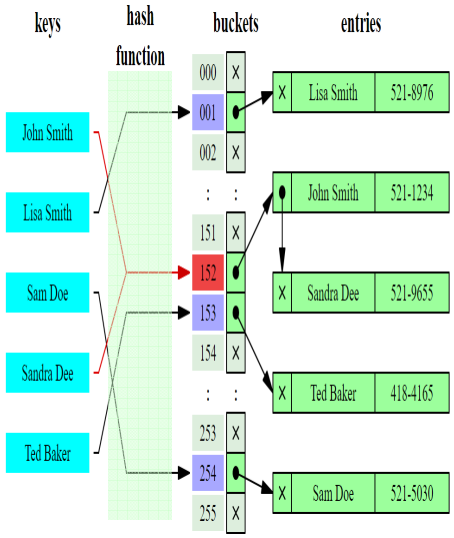
Opening an indexed file (Fig 9.17)



Traditional File Structures

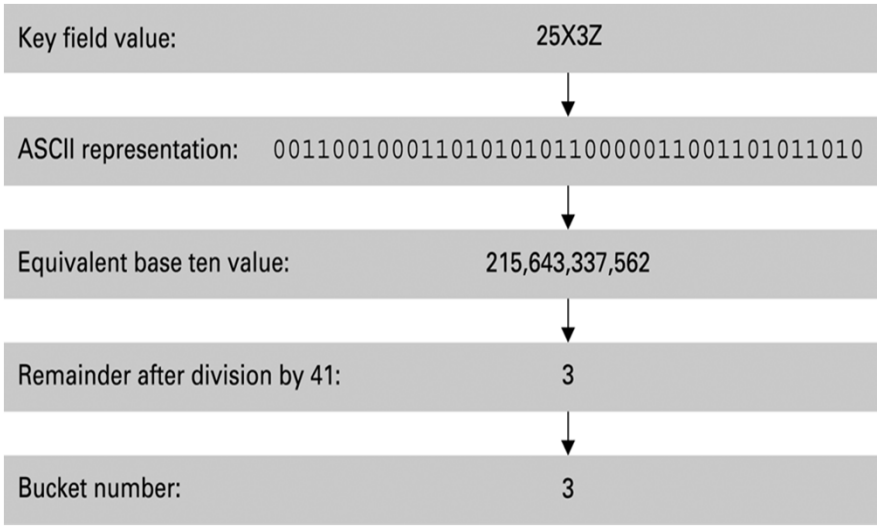
Hashing

- Each record has a **key** field
- The storage space is divided into **buckets**
- A **hash function** computes a bucket number for each key value
- Each record is stored in the bucket corresponding to the hash of its key



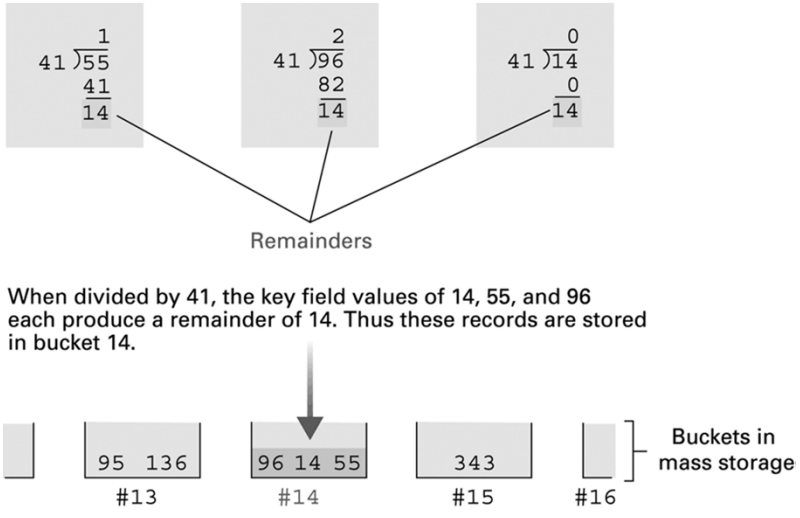
Traditional File Structures

Hashing the key field value 25X3Z to one of 41 buckets (Fig 9.18)



Traditional File Structures

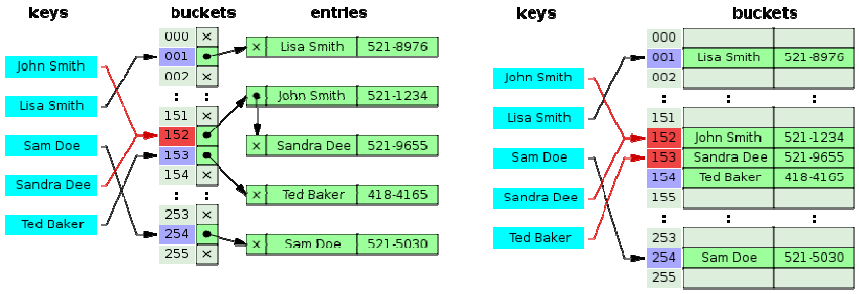
Rudiments of a hashing system (Fig 9.19)



Traditional File Structures

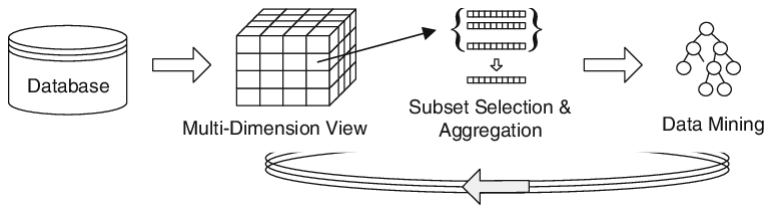
Collisions in Hashing

- **Collision:** The case of two keys hashing to the same bucket
  - Major problem when table is over 75% full
  - Solution: increase number of buckets and rehash all data or implement **open** (separate chaining) or **closed** (open addressing) hashing.



Data Mining

- **Data Mining:** The area of computer science that deals with discovering patterns in collections of data.
- Important tool in numerous areas including marketing, inventory management, quality control, fraud detection, and investment analysis.
- **Data warehouse:** A static data collection to be mined
  - **Data cube:** Data presented from many perspectives to enable mining



Data Mining

Data Mining Strategies

- **Class description**
  Identifying properties that characterize a given group of data items e.g. people who buy small economical vehicles
- **Class discrimination**
  Identifying properties that divide two groups e.g. used vs. new car buyers
- **Cluster analysis**
  Find properties of data items that lead to the discovery of groupings e.g. discovering age groups that watch a particular movie
- **Association analysis**
  Looking for links between data groups e.g. people that buy chips also buy beer and soda
- **Outlier analysis**
  Identify data entries that do not comply to the norm e.g. identify potential terrorists by recognizing unusual behavior
- **Sequential pattern analysis**
  Identify patterns of behavior over time e.g. climate change

Social Impact of Database Technology

- Problems
  - Massive amounts of personal data are being collected
    - Often without knowledge or meaningful consent of affected people
  - Data merging produces new, more invasive information
  - Errors are widely disseminated and hard to correct
- Remedies
  - Existing legal remedies often difficult to apply
  - Negative publicity may be more effective i.e. public opinion

Chapter 9: Topics Covered

- Object-Oriented Databases
- Maintaining Database Integrity
- Traditional File Structures
- Data Mining
- Social Impact of Database Technology

