Advanced Databases

Dr. Rabah Mokhtari

Department of Computer Science,
UNIVERSITY OF M'SILA

November 10, 2019

Content

1 Introduction

2 Database Management System

3 History of Database Models and Management

Introduction

Fundamental concepts

General definition of a Database

- A database is a collection of related data.
- This collection of data is stored for later retrieval and processing.
- Data can be recorded or stored in physical drives.
- Data is used to reference known facts like, for example, person names, phone numbers and product prices.

Data VS Information

- **Data** is collection of raw facts. It simply exists and has no significance beyond its existence.
- **Data** is always need to be processed.
- For example, age of students blood group of students, are generally considered as “data”

Fundamental concepts

Data VS Information

- After processing, organization and presenting data in a given situation so to make it useful, it becomes Information.
- For example, height of a particular student.

File-based Systems Vs Database Management Systems

Before the advent of database management, data was stored in data file. Systems used data file are called file-based or non-database systems.

File-based system

A nondatabase (or file-based) system is a program or a collection of programs responsible for data file managing and processing

Database Management Systems (DBMS)

DBMS is a collection of programs that enable to create and maintain a database

Database Approach

Characteristics of the Database Approach

1. Data Abstraction: A data model is used to hide storage details and present the users with a conceptual view of the database.
2. Support of multiple views of the data: the database is populated for several users using different views according to the interest of each user.
3. Concurrency : Allowing a set of concurrent users to retrieve from and to update the database.

Database Management System

Definition

DBMS

- Database Management System (DBMS) is a software system that is used to define, create, manipulate, and maintain databases.
- Popular DBMSs include Microsoft Access, Oracle, DB2, MySQL, and Microsoft SQL Server.

Classification of DBMS users

DBMS users

1. Database Administrator (DBA).
2. Database Designer.
3. End Users.
4. Application Programmers.

DBA

The DBA is the person or the group of person in charge of performing database administration tasks to manage both the database and the use of the DBMS.

Database designer

The database designer is responsible for defining the detailed database design, including data structure, relationships between data, constraints, ...

Classification of DBMS users

End users

- End Users are the people who interact directly with the database through utilities.
- They are authorized to query and update all or part of the database using standards of Queries to respond quickly to other user requests.

Application Programmers

An application programmer interact with the database from programs written in high-level programming languages such as C++ and Java.

Advantages of Database Approach

Advantages

1. Getting more information from the same amount of data.
2. Data integrity and security.
3. Controlling Data Redundancy and Inconsistency.
4. Data independence.
5. Increasing programmer productivity.

Disadvantages of Database Approach

Disadvantages

1. Cost of stuff training.
2. Cost of Hardware & Software.
3. Cost of Data Conversion.
4. More difficult recovery.

History of Database Models and Management

Overview

Good reference

- To gain insight into the present state of database models and architectures, we need to present a historical introduction of the evolution of database management systems.
- A good historical presentation of evolution of Data Processing and Database Management Systems may be found in the book titled *Inside the Database Object Model* written by Burleson, D. K. (1998).

Flat File Processing

Flat file

- The early computers were very large and cumbersome to maintain.
- Computers were perfect for performing repetitive tasks as payroll calculations
- Organizations soon began to see that high volume for repetitive data storage and processing were ideal applications for these computers.
- In that time, many of "database" systems were really nothing more than a loosely coupled collection of files.

Flat file for data storage

- These were called "flat files" because data was stored as fixed length records in flat-files in a **linear** fashion.
- It was necessary to read the file from front-to-end until retrieving the desired record.
- There are no structures for indexing or recognizing relationships between records.

Flat File Processing

Flat file for data storage

- Flat files are said to be **non-keyed** files because records was always retrieved in the same order.
- New record will be stored at the **end** of file.

Early access methods

- An access method defines the technique that is used to store and retrieve data.
- QSAM and BSAM were often used to describe physical **sequential** files in IBM mainframe environment.
- QSAM stands for *Queued Sequential Access Method* (pronounced “quesam“)
- BSAM *Basic Sequential Access Method* (pronounced ”bee-sam“)

Basic Direct Access Method (DBAM)

BDAM

- After it became possible to store data on disk, organizations struggled to bypass the linear nature of flat file organization.
- Each block and record can be identified by a **unique** disk address.
- BDAM (pronounced ”bee-damn“) method uses hashing algorithm to determine the disk address of the target stored record.
- BDAM algorithm will compute the target location (address) of the record based on its **symbolic key**.
- BDAM provides much faster access and retrieval of records.
- A direct access file is sometimes called a **”keyed“** file.

BDAM Strorage and retrieval

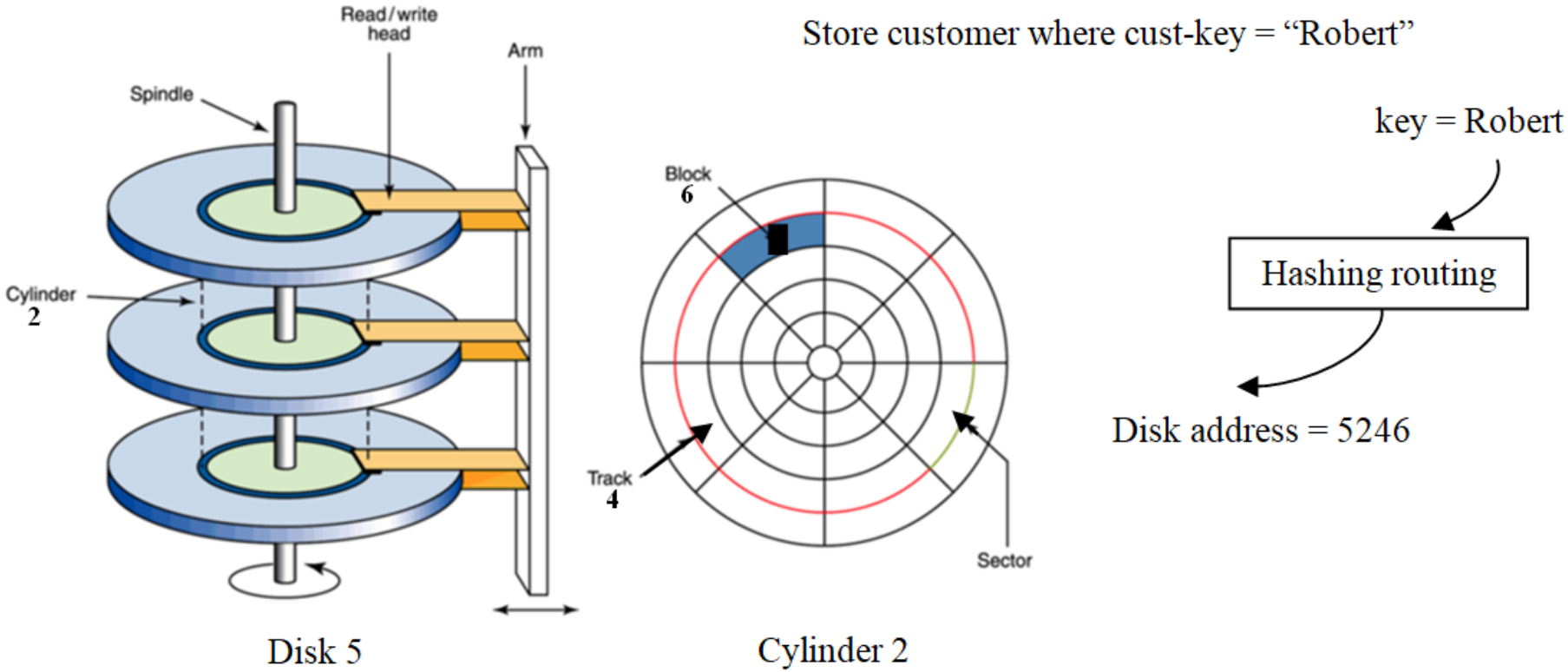

Figure: Hashed file storage

The Indexed Sequential Access Method (ISAM)

BDAM Problem

BDAM method provides presents a good solution for fast storage and retrieval of data. But the high cost of disk storage made it a very expensive proposition.

Flat files with indexes

- Flat files present a good solution for low cost of data storage space.
- How to speed data storage and retrieval on flat files ?
- New methods used flat files with **indexes**.
- These new methods became a very popular alternative to BDAM.
- Examples, ISAM (Indexed Sequential Access Method) and VSAM (Virtual Sequential Access Method) of IBM.

The Indexed Sequential Access Method (ISAM)

Computer Indexes are like indexes used in book

- Book indexes allow you to find what you want quickly.
- Computer index can speed the **retrieval of information**.

Index Structure

- In the simplest index structures, the index only contains two fields.
- One field is the **symbolic key** and the second field contains the **disk address** of record that contains the key value.
- In most file management systems, the index file kept as **completely separate file** from the master file

The Indexed Sequential Access Method (ISAM)

Record Storage and retrieval

- When a record is requested, the program will scan the index, locate the **symbolic key**, and then **retrieve** the record from the master file based on the disk address value.
- When new records are added to the end of the master file, the ISAM file system will automatically adjust the indexes to account for change.

The Indexed Sequential Access Method (ISAM)

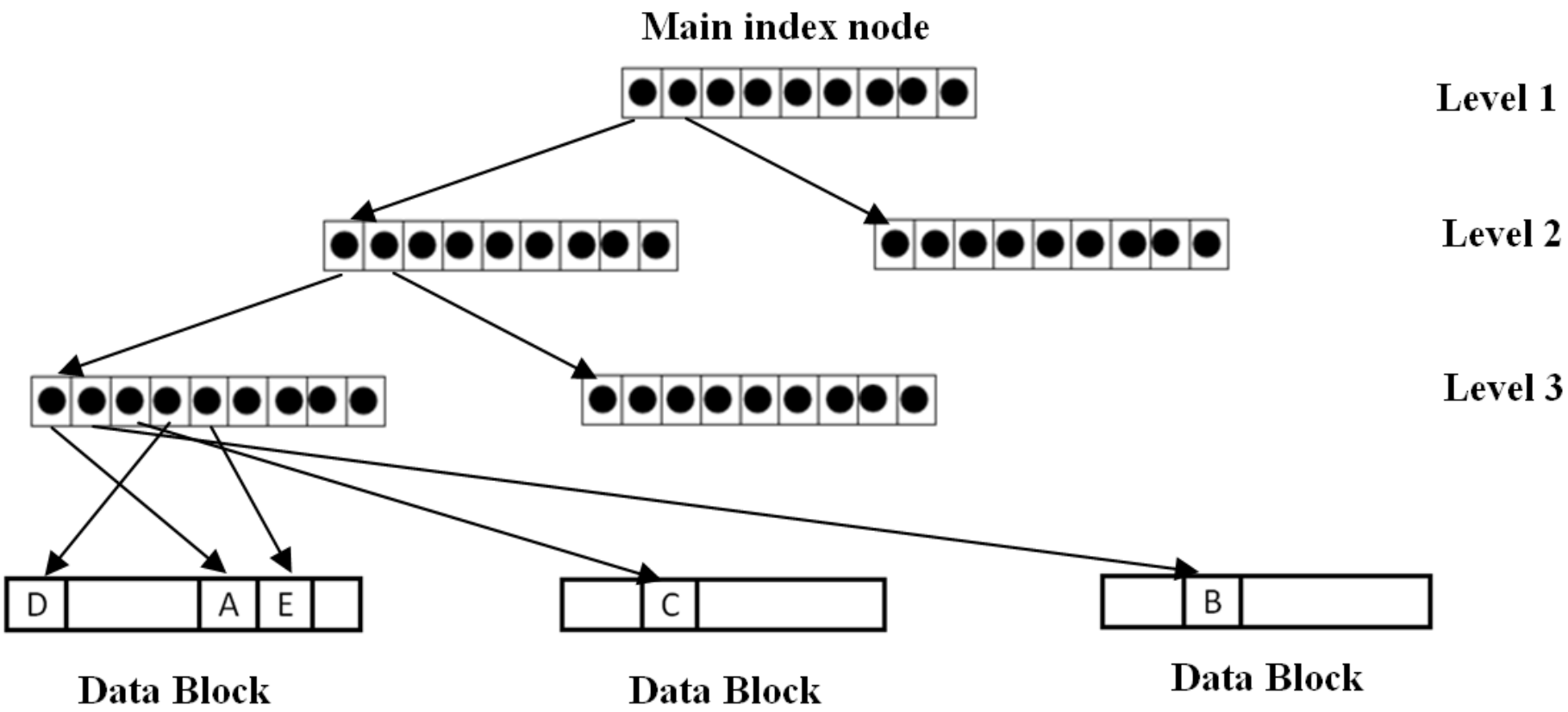

Figure: A simple index retrieval

Shortcomings of flat file databases

There were many problems and difficulties with flat-file database systems, such as

Shortcomings of flat file DB systems

- **Data sharing**
- **Duplication of data**
- **Data file structure modification**
- **Backup and recovery**

The Era of formal Database Management

Shortcomings of flat-file systems prompted the development of new approach : database approach.

DBMS features

1. Recovery of incomplete transactions (rollback)
2. A mechanism for recovering transactions after disk failure (roll forward)
3. Internal tools for management of relationships between data.
4. Locking and concurrency tools for simultaneous multiple user access.
5. A common access language that can be embedded in procedural code.

The Era of formal Database Managements

The database management systems of 1990s also provide:

1990s DBMS features

1. Internal support for maintaining business rules (referential integrity).
2. Read consistency for long-running queries.
3. Support for distributed updates (two-phase commit).
4. Attaching behaviors to data (methods).

Early Database Management

IMS

- In the late 1950s, IBM developed a prototype computer database to demonstrate that data could be stored, retrieved, and updated in a structured format.
- This database became known as the revolutionary Information Management System, or **IMS**.
- **IMS** allows data access by numerous programs in different languages.
- **IMS** was designed to support the multiuser needs of larger organizations.
- IMSs manage **hierarchical database model**.

Important

The IMS database has concurrent control, and a full backup and recovery mechanism

Hierarchical Database Model

Introducing of hierarchical databases

- The hierarchical database model was first introduced as IMS.
- The model was released in **1960**.
- The hierarchical database model used **pointers** to logically **link** related data items.
- In 1998, IMS still enjoys a large following among users with large databases and high-volume **transaction** requirements.

Hierarchical Database Model

A hierarchical database is very well suited to modeling **relationships** that are **naturally hierarchical** (see the University database bellow).

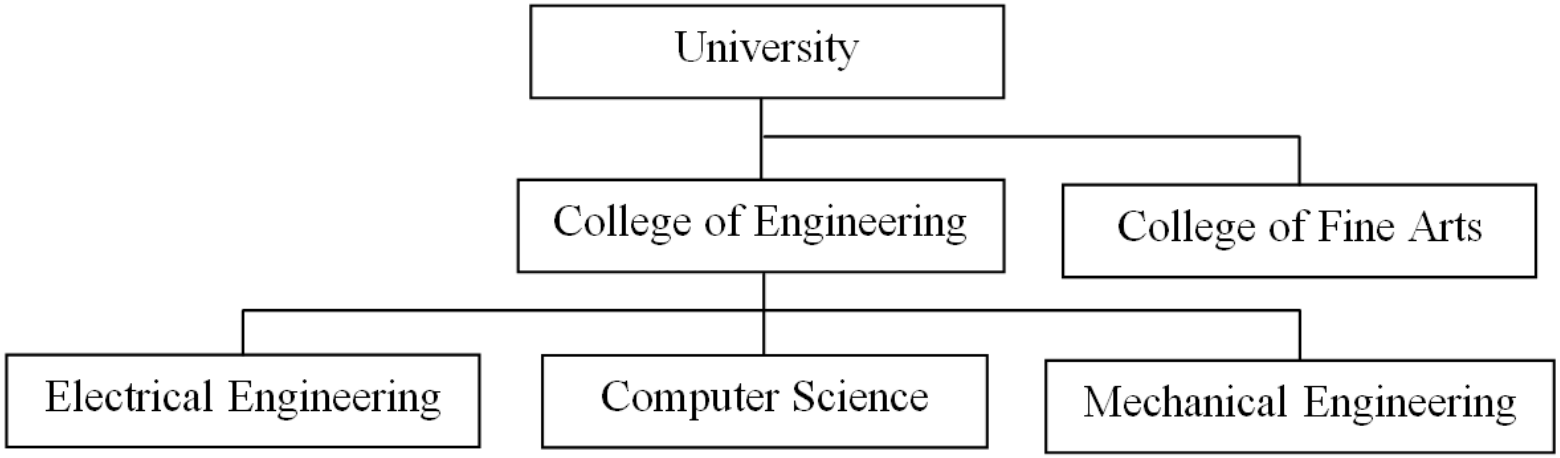

Figure: A simple hierarchical chart

Hierarchical Database Model

Segment type

- The nodes in a hierarchical Database are called **Segment Types**.
- A segment type is simply **user-defined** category of data, and each segment contains **fields**.
- Each segment has a **key field**, and the key field is used to retrieve the data from the segment.
- The segment types are composed of descending **one-to-many** relationships.

Eg. University record fields

UNIVERSITY information might include the UNIVERSITY name, the mailing address, and phone number.

Hierarchical Database Model

Eg. One-To-Many relationship

The **College of Engineering** has 3 departments, **Electrical Engineering**, **Computer Science**, and **Mechanical Engineering**.

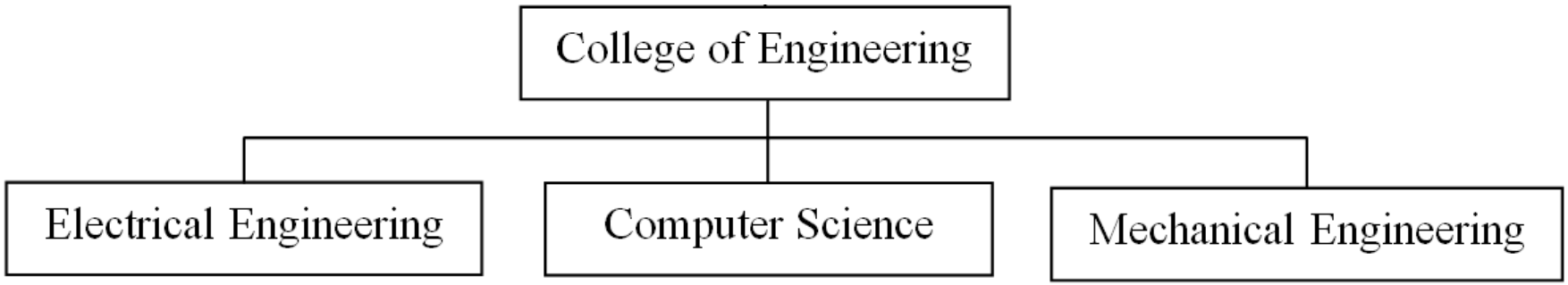

Figure: One-To-Many relationship example

Discussion of the hierarchical DB

Advantages

1. A large base with a proven technology that has been around for decades.
2. The ease of using a hierarchy or tree structure.
3. The speed of the system (exceeding 2000 transactions per second).

Disadvantages

Some disadvantages of hierarchical databases are because of rigid rules in relationships :

1. **Insertion** and **deletion** can become very complex.
2. **Access** to child segment can only be done through the parent (root) segment.
3. IMS databases are not suitable for **many-to-many** and **recursive many-to-many** like **BOM** (Bill-Of-Material) relationships.
4. **Cost of stuff training –** To become proficient in IMS you need months of training and experience.

CODASYL Network Model

CODASYL Definition

- CODASYL stands for *Conference on Data Systems Language* was a group of organization of volunteer representatives of computer manufacturers formed in **1959** to guide the development of Data Systems Languages standards.
- The first CODASYL DBMS specifications were not produced until **1969**.
- After some revisions, the first real CODASYL DBTG specifications of the CODASYL standard approach were in **1971**.

CODASYL DBMS and COBOL language

- CODASYL was responsible for standardization of **CODASYL Network DBMS** and **COBOL** language.
- **CODASYL Network Model** is known also as **Navigational Model**.
- CODASYL Databases used a network linked-list structure, so to access and retrieve data, the program will navigate through the specified database using explicit procedures of the DML (Data Manipulation Language) language.

CODASYL Network Model

CODASYL Data languages

The Data Base Task Group (DBTG) defined three languages:

1. **Device Media Control Language (DMCL)**
2. **Data Manipulation Language (DML)** The DML defined commands used to navigate through the linked-list structures hat comprised the database. CODASYL DML verbs included ”GET“, ”FIND“, ”STORE“, ”MODIFY“, and ”DELETE“.
3. **Data Definition Language (DDL)** used to define the **Logical Structure** of the database.

DBA and DBM Users

Two main users in CODASYL Databases:

1. Data Base Administrator (DBA) functions included: data structure or schema, data integrity, security, and authorization.
2. Data Base Manager (DBM) functions included: operation, backup/recovery, performance, statistics, and auditing.

CODASYL Network Network Model

CODASYL Database storage and retrieval methods

- CODASYL Model uses BDAM method to store and retrieve records to and from the network linked-list database.
- BDAM provides a faster access and retrieval of records.

Bachman diagram

Charles W. Bachman[1] developed a ”diagram“ that represented the data structures as required by CODASYL Model. This diagram method becomes known as the Bachman diagram.

[1] Charles Bachman is best known for his invention of the first random access database management system, the Integrated Data Store (IDS) and he won the ACM's 1973 A.M. Turing Award for his outstanding contributions to database technology

CODASYL Network Model

Record in Bachman diagram

- The Bachman diagram describes the physical constructs of all record types in the database.
- A CODASYL **record type** corresponds to **segment type** in the hierarchical model.

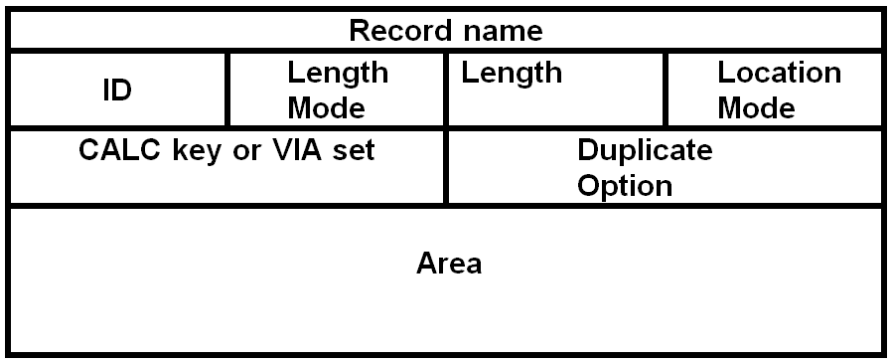

Figure: Bachman diagram layout of record structure

The rectangles of Bachman diagram are subdivided into four rows.

1. The top row of the box contains the record name.
2. Row two contains the numeric identification ID number, the length mode (fixed or variable), the length of records, and the location mode (CALC, or VIA).
3. Row three contains for CALC, the field serving as the CALC key, and for VIA SET, the set name.
4. Row four contains the era designated.

CODASYL Network Model

CODASYL Model Set Type

The set type is shown by Bachman arrow pointing from the owner record type to the member record type.

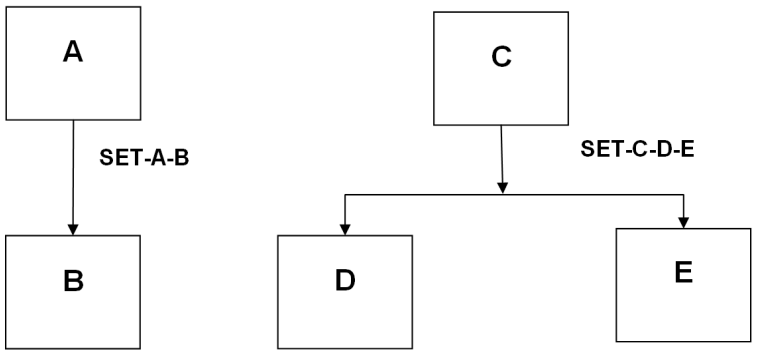

Figure: A set type of one owner record type and one or more member record types

CODASYL Network Model

CODASYL Database example

CODASYL Network Model

Shortcomings of CODASYL Model

1. Network databases, very much like hierarchical databases, are very difficult to navigate.
2. The Data Manipulation Language (DML), designed for complex navigation, was a skill that required months of training.
3. Implementing structural changes was extremely difficult since data relationships are ”hard-linked“ with embedded pointers.

Eventually, the CODASYL approach lost its popularity as simpler, easier-towork-with systems came on the market.

Relational Database Model

Founder of the Relational Model

- **Edgar Codd** was primarily involved in the development of hard disk systems at IBM in San Jose, California.
- He was unhappy with the navigational model of the CODASYL approach and the IMS model.
- He wrote a series of papers, in **1970**, outlining novel ways to construct databases.
- His ideas eventually evolved into a paper titled, *A Relational Model of Data for Large Shared Data Banks* (ACM, 1970).

Relational Database Model

Mathematical theory for Relational Model

The relational model is based on a collection of mathematical principles drawn primarily from **set theory** and **predicate logic**.

Data structures

- Records of data are stored in **tables** (also referred to as **entities**).
- Each **table** has **rows** (also referred to as **records** or **tuples**).
- Each Table has also **columns** (also referred to as **attributes**).
- **Tables** basically correspond to **segment type** in hierarchical and **record types** in the network models.
- **Tables** are independent, unlike hierarchical and network models that are pointer connected.
- **Tables** can contain only one type of record, and each record has a fixed number of fields that are explicitly named.
- Data **redundancy** is reduced via a process called **normalization**.

Relational Database Model

Primary and Foreign keys

- A field or several fields that make a unique identifier in a table called **primary key**.
- A primary **key uniquely** identifies a row in a table.
- The **foreign key** allows you to join two or more tables together by using a primary key in one table with a **non key** field in another table.

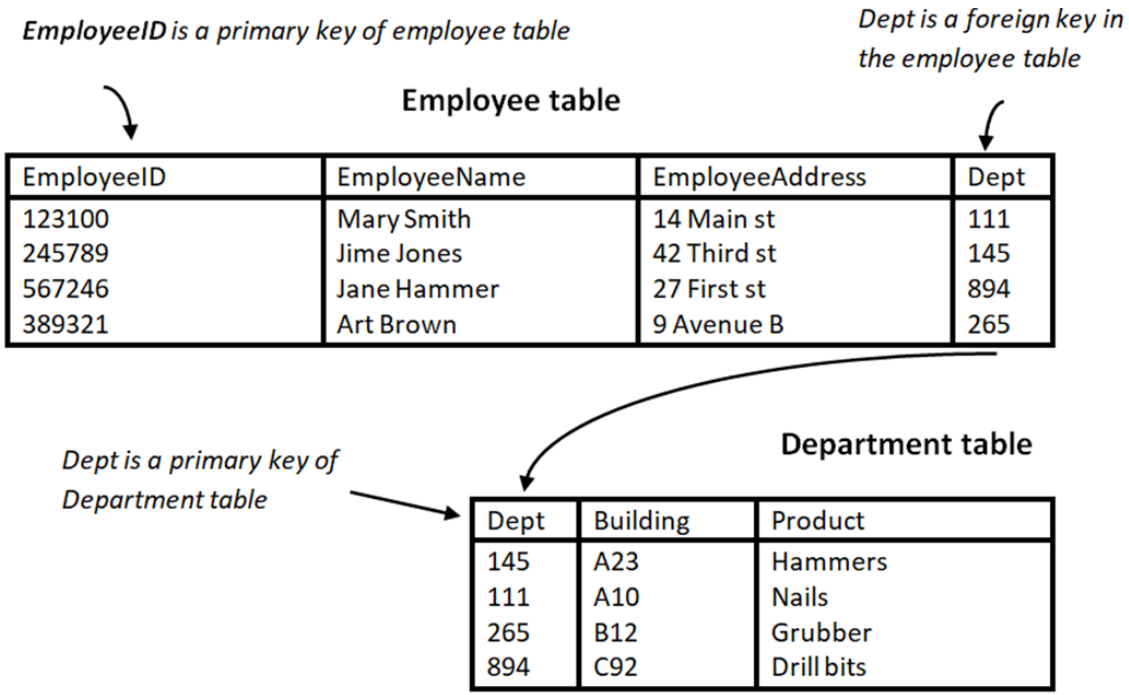

Figure: A simple relational chart

Relational Database Model

Relational databases made the following improvement over hierarchical and CODASYL network databases

Relational vs Hierarchical and and Network

1. **Simplicity** : The concept of tables with rows and columns is extremely simple and easy to understand.
2. **Data independence** : Data independence is because tables are not hard linked to one another. Data independence is ability to modify data structure (in this, case, tables) without affecting existing programs.
3. **Declarative data access** : Rather than having dozens of CODASYL DML commands,the relational model introduced a declarative language called Structured Query Language (SQL), also known as **sequel**.

Object-Oriented Database Model

Origins

- The term of Object-Oriented, abbreviated OO, has its origins in OO programming language.
- The main idea the OO paradigm is to couple the data with behavior.
- The first procedural language to do this task is **Simula** language, which was proposed in the late **1960s**.
- In **1970**, Xerox PARC developed **Smalltalk**, which was one of the first OO languages to explicitly incorporate additional OO concepts.

OO Databases Management

- Object in data management sense was first developed by Xerox Corporations Palo Research Center (PARC) in the early **1970s**.
- OO databases management approach emphasizes a more natural representation of the data.
- In the late 1990s environments, the data models are more demanding. They need to handle audio, video, text, graphics, etc which need more flexible storage format than hierarchical, network, and relational databases can provide.

References