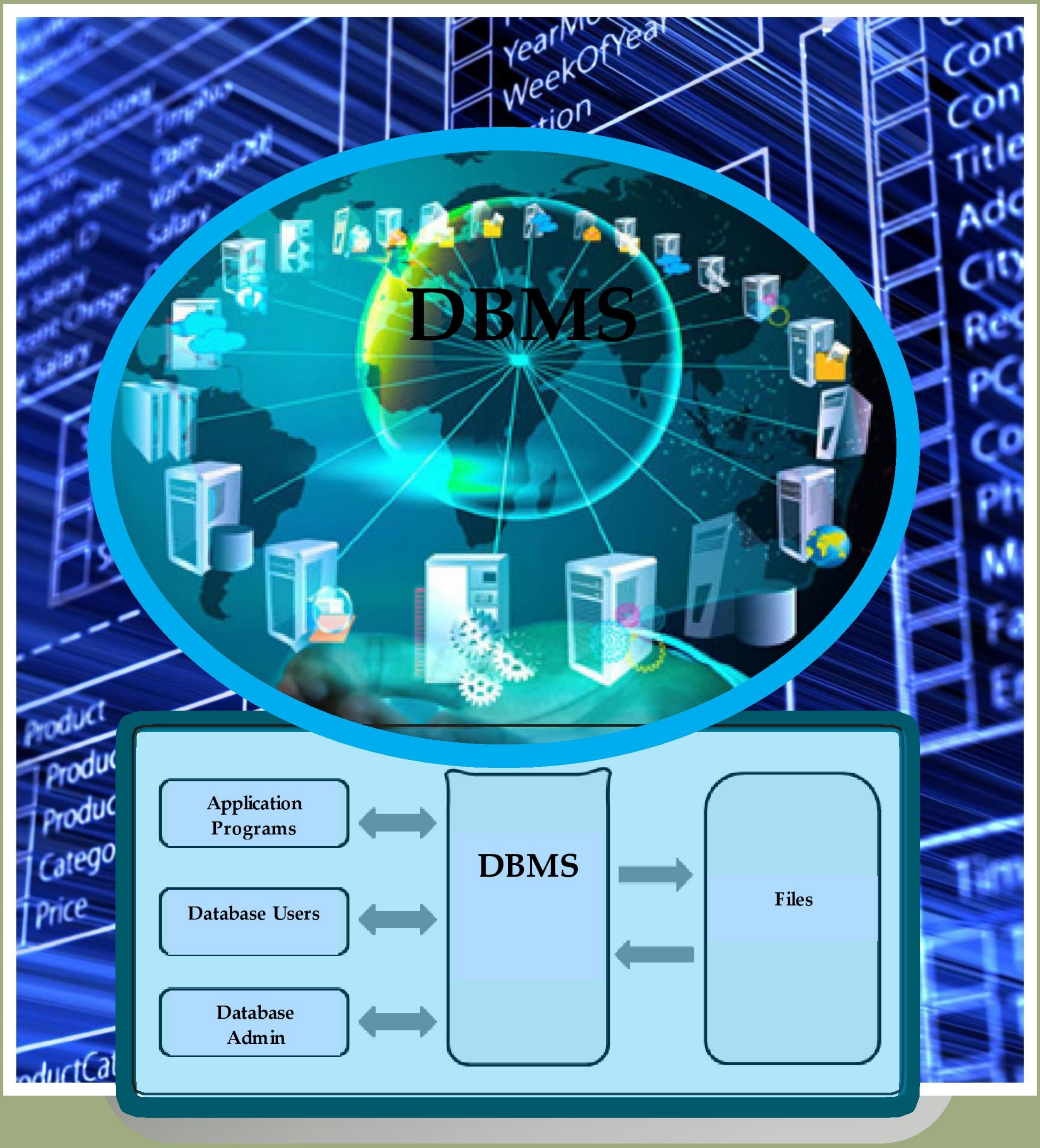
ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Computer and Information
Sciences (SOCIS)

BCS-092
Introduction of Databases



Database System – Basic Concepts and Models

1

ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi
National Open University
School of Computer and
Information Sciences

BCS-092
Introduction to
Databases

Block

1

DATABASE SYSTEM – BASIC CONCEPTS AND MODELS

Course Coordinator (for the year 2021)

Mr. Akshay Kumar, Associate Professor
SOCIS, IGNOU, New Delhi-110068

In all the Units of this Block, Check Your Progress and Answers to Check Your Progress Questions are written and added in 2021 BY

Mr. Akshay Kumar, Associate Professor
SOCIS, IGNOU, New Delhi-110068

PRODUCTION

Mr. Tilak Raj
Asst. Registrar (Pub.)
MPDD, IGNOU, New Delhi

Mr. Yashpal
Section Officer (Pub.)
MPDD, IGNOU, New Delhi

January, 2021

Laser Typesetting : Akashdeep Printers, 20-Ansari Road, Daryaganj, New Delhi-110002

IGNOU is one of the participants for the development of courses of this Programme

Copyright

This course has been developed as part of the collaborative advanced ICT course development project of the Commonwealth of Learning (COL). COL is an intergovernmental organization created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

The Open University of Tanzania (OUT) is a fully fledged, autonomous and accredited public University. It offers its certificate, diploma, degree and postgraduate courses through the open and distance learning system which includes various means of communication such as face-to-face, broadcasting, telecasting, correspondence, seminars, e-learning as well as a blended mode. The OUT's academic programmes are quality-assured and as centrally regulated by the Tanzania Commission for Universities (TCU).

For the avoidance of doubt, by applying this licence the Commonwealth of Learning does not waive any privileges or immunities from claims that it may be entitled to assert, nor does the Commonwealth of Learning submit itself to the jurisdiction, courts, legal processes or laws of any jurisdiction. The ideas and opinions expressed in this publication are those of the author/s; they are not necessarily those of *Commonwealth of Learning* and do not commit the organisation.

The Open University of Tanzania
P. O. Box 23409,
Kinondoni,
Dar Es Salaam,
Tanzania
Phone: +255 22 2668992
Fax: +255 22 2668756
Email: hod.ict@out.ac.tz
Website: www.out.ac.tz

Commonwealth of Learning
4710 Kingsway, Suite 2500,
Burnaby V5H 4M2,
British Columbia,
Canada
Phone: +1 604 775 8200
Fax: +1 604 775 8210
Email: info@col.org
Website: www.col.org

ACKNOWLEDGEMENTS By AUTHORS

The Open University of Tanzania (OUT), Faculty of Science, Technology and Environmental Studies, (FSTES) and the ICT Department wish to thank those below for their contribution to the production of this course material and video lectures:

Authors:
Ms. Grace W. Mbwete (Assistant Lecturer, ICT Department, OUT)
Mr. Elia A. Lukwaro (Tutorial Assistant, ICT Department, OUT)

Copy Editor:
Mr. Justin Kimaro (Principal Editor, IEMT, OUT)

Reviewers:
Ms. Lilian C. Mutalemwa (Assistant Lecturer, ICT Department, OUT)
Ms. Regina Monyemangene (Instructional Designer, IEMT, OUT)

Video Production:
Mr. Othman Mwinchoum (Multimedia studio, IEMT, OUT)
Ms. Grace W. Mbwete (Assistant Lecturer, ICT Department, OUT)
Ms. Lilian C. Mutalemwa (Assistant Lecturer, ICT Department, OUT)

This course has been developed with the support of the ***Commonwealth of Learning, Canada.***

COURSE INTRODUCTION

The course content

The course is broken down into units. Each unit comprising:

- An introduction to the unit content.
- Unit outcomes.
- New terminology.
- Core content of the unit with a variety of learning activities.
- A unit summary.
- Review questions, case studies and assignments, as applicable.
- Reference and Further Reading.
- Attribution.
- Links to video lectures for each unit are provided at the end of this course material

Resources

For those interested in learning more on this subject, we provide you with a list of additional resources at the end of each unit, these may be books, articles or websites. Links to video lectures for each unit are provided at the end of this course material.

Course overview

This course will equip students with ICT skills which include theory and practical of fundamental concepts of database management systems, data modelling techniques, data normalization, Structured Query Language (SQL) and finally, database development process. At the end of this course, you should be able to take a real world case scenario and create a database that adhere to a user requirements and required functions.

Course Overview Video

https://tinyurl.com/h4ktsab

The course aims are:

- To introduce learners to the requisite theory and practice of database technology and the applications of the technology in generic and specific domains.
- To enable the learners to learn and apply methodologies for conceptual, logical and physical database design.
- To enable the learner to acquire skills in solving business problems using the fundamentals of database modeling, enterprise analysis and design.
- To introduce learners to implementation and management issues as well as database definition and manipulation languages.

Course outcomes

Upon completion of Introduction to Databases, learners should be able to:

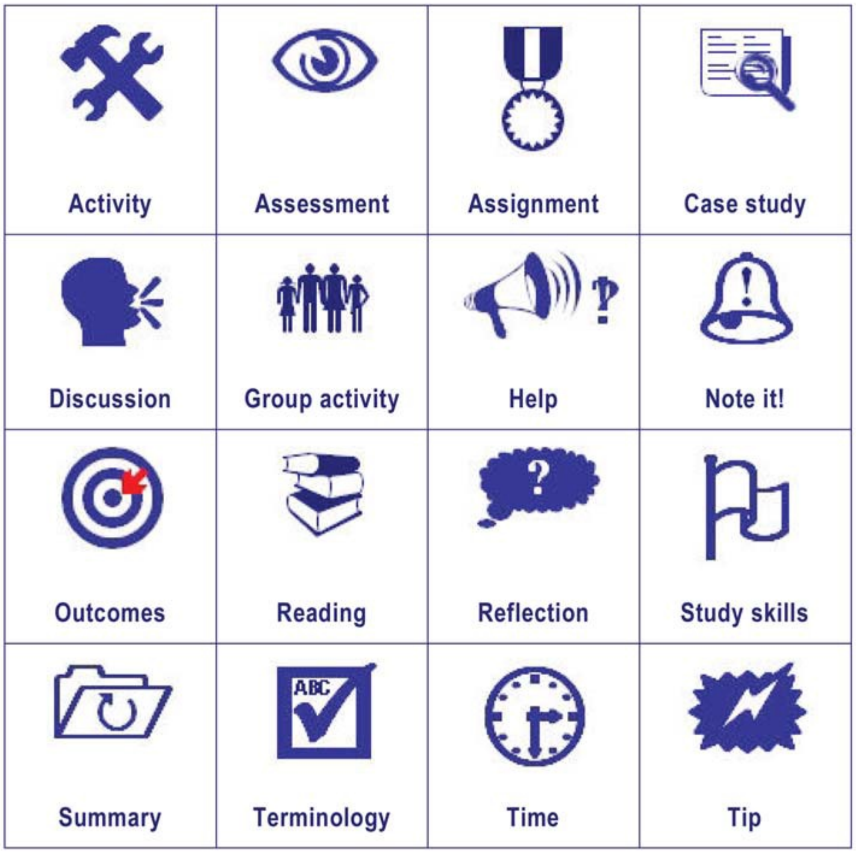
- *Explain the basic definition and characteristics of databases.*
- *Compare and contrast different roles of database end users.*
- *Differentiate between various data models.*
- *Design physical database schema for given user requirements.*
- *Develop an entity-relationship model based on user requirements.*
- *Normalize a set of relational schema to a third normal form.*
- *Describe and use data definition language commands (SQL).*
- *Describe and use data manipulation commands (SQL).*
- *Describe how to join tables for use in SQL.*
- *Design, Implement and Test a database according to given user requirements.*

Getting around this Course Material

Margin icons

While working through this Course Material you will notice the frequent use of margin icons. These icons serve to "signpost" a particular piece of text, a new task or change in activity; they have been included to help you to find your way around this Course Material.

A complete icon set is shown below. We suggest that you familiarize yourself with the icons and their meaning before starting your study.



BLOCK INTRODUCTION

This Block provides an introduction to the basic concepts of the database management systems. This Block consists of 5 Units.

Unit 1 introduces various terminologies related to database management system. It also introduces the functions, characteristics and database users. This Unit also provides a classification of DBMSs.

Unit 2 discusses a brief history of the origin of the database approach and its importance.

Unit 3 introduces the basic modeling for the database architecture, especially the three layer architecture of the database.

Unit 4 introduces the basic database models

Unit 5 provides an introduction to relational model, explaining the fundamental concepts and properties of table in relational model.

UNIT 1 INTRODUCTION TO DATABASE SYSTEMS

Structure

1.1 INTRODUCTION

In the present time, database systems are used to electronically store all the data of an organization. Databases allow data sharing and integrate data of an organization. In this unit, we are going to be introduced to the basic concepts of database systems. This is an introductory unit where we are going to learn about the definition of a database, its characteristics, classification, functions and the user types which are involved in using the database.

1.2 OBJECTIVES

Upon completion of this unit you should be able to:

- Describe what a database is and how it functions.
- Outline various properties of a database.

- Explain the roles of different database end users.
- Compare and contrast the different databases based on their classification.

1.3 TERMINOLOGIES

Database : Organised set of data in a computer.

DBMS : Database Management System, a software used to manage database.

Data : Raw facts.

Information : Processed data.

Centralized DB : Centralized Database is a database which is located and managed in one single site.

DBA : Database Administrator - performs all activities related to maintaining a successful database environment.

1.4 DATABASES: DEFINITION AND CHARACTERISTICS

This section introduces you to some of the basic definitions and characteristic of database system. The next subsection defines the term database.

1.4.1 What is Database?

A database is a shared collection of related data used to support the activities of a particular organization. A database can be viewed as a repository of data that is defined once and then accessed by various users as shown in Figure 1.1.

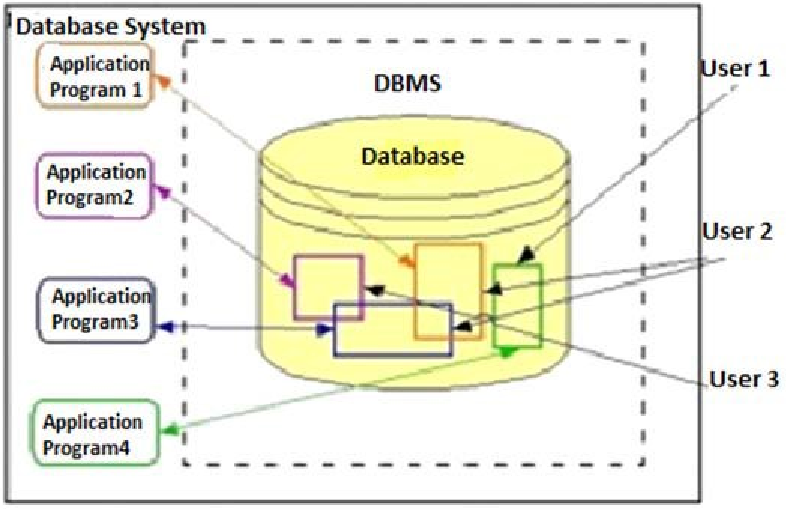


Figure 1.1: A database is a repository of data, by A. Watt.

It may be noted in figure 1.1 that database is a vast collection of data, on which large number of application programs can be developed; and a large number of users may be able to use the data of a database. Please also note that these application programs or user may be allowed to use only part data of database.

1.4.2 Characteristics of a Database

There are a number of characteristics that distinguish the database approach with the file-based approach. The following are the characteristics (features) and benefits of the database systems:-

i) **Self-Describing Nature of a Database System**

A database system is referred to as *self-describing* because it not only contains the database itself, but also metadata which defines and describes the data and relationships between tables in the database. This information is used by the DBMS software or database users, if needed. This separation of data and information about the data makes a database system totally different from the traditional file-based system in which the data definition is part of the application programs.

ii) **Insulation between Program and Data**

In the file based system, the structure of the data files is defined in the application programs so if a user wants to change the structure of a file, all the programs that access that file might need to be changed as well. On the other hand, in the database approach, the structure of data is stored as metadata in the system catalog and not in the programs. Therefore, one change is all that is needed.

iii) **Support multiple views of data**

A view is a subset of the database which is defined and dedicated for particular users of the system. Multiple users in the system might have different views of the system. Each view might contain only the data of interest to a user or a group of users (Refer to Figure 1.1).

iv) **Multiuser system**

A multiuser database system must allow multiple users access to the database at the same time. As a result, the multiuser DBMS must have concurrency control strategies to ensure several users access to the same data item at the same time, and to do so in a manner that the data will always be correct.

v) **Control Data Redundancy**

In the Database approach, ideally each data item is stored in only one place in the database. In some cases redundancy still exists so as to improve system performance, but such redundancy is controlled and kept to minimum.

vi) **Data Sharing**

The integration of the whole data in an organization leads to the ability to produce more information from a given amount of data.

vii) **Enforcing Integrity Constraints**

DBMSs should provide capabilities to define and enforce certain constraints such as data type and data uniqueness, to ensure the correctness of data. These constraints are also known as data integrity constraints.

viii) **Restricting Unauthorized Access**

Not all users of the system have the same accessing privileges. DBMSs should provide a security subsystem to create and control the user accounts.

ix) **Data Independence**

System data (Meta Data) descriptions are separated from the application programs. Changes to the data structure is handled by the DBMS and not embedded in the program.

x) **Transaction Processing**

The DBMS must include concurrency control subsystems to ensure that several users trying to update the same data do so in a controlled manner. The results of any updates to the database must maintain consistency and validity.

xi) **Providing backup and recovery facilities**

If the computer system fails in the middle of a complex update process, the recovery subsystem is responsible for making sure that the database is restored to the stage it was in before the process started executing.

Check Your Progress 1

Q1. What is meta-data in a database?

....

....

....

Q2. When is the concurrency control needed in a database?

....

....

....

Q3. What is meant by data redundancy?

....

....

....

1.5 FUNCTIONS OF A DATABASE

Today, databases are widely being used by different organizations whereby application depends on their specific requirements and business needs. In summary, this section will outline functions of a database so that there is greater understanding of why an individual or an organization would want to use a database management system. These functions includes but not limited to:

- To organize and store data.

- To provide facilitation for analysis and modeling.
- To process given data and turn it into information
- To explore data using exploratory techniques
- To support organizational activities

1.6 DATABASE USAGE AND ENVIRONMENT

Database usage is almost everywhere in different aspects of everyday life. The following are some of the applications in which databases are used in carrying out organizational activities:

- Banking - assist in carrying out banks' customer's financial transactions.
- Airlines - provide flexible means of flight scheduling and reservations.
- Universities - platform for registration and grades recording for students.
- Sales - storing customer, products and purchases details.
- Manufacturing - organizing data on production, inventory, orders and the whole supply chain.

1.7 DATABASE USERS

These are people whose jobs require access to a database for querying, updating and generating reports. An end user might be one of the following:

Application User

The application user uses the existing application programs to perform their daily tasks.

Sophisticated user

Sophisticated users are those who have their own way of accessing the database. This means they do not use the application program provided in the system. Instead, they might define their own application or describe their need directly by using query languages. These specialized users maintain their personal databases by using ready-made program packages that provide easy-to-use menu driven commands, such as MS Access.

Application Programmers

Application Programmers are technical personnel who implement specific application programs to access to the stored data. This kind of user needs to be familiar with the DBMSs to accomplish their task.

Database Administrators

Database Administrator(s) can be a one person or a group of people in the organization who is responsible for authorizing the access to the database, monitoring its use and managing all the resource to support the use of the whole database system.

Check Your Progress 2

Q1. How can a database be used to convert data into information?

....

....

....

....

....

Q2. What are the applications of database in a University like IGNOU?

....

....

....

....

....

Q3. You have developed a database driven mobile application. What kind of user of database are you?

....

....

....

....

....

1.8 CLASSIFICATION OF DBMSS

Database management systems can be classified based on several criteria, such as the data model, user numbers and database distribution as explained in this section.

1.8.1 Data Model

The most popular data model in use today is the relational data model. Well-known DBMSs like Oracle, MS SQL Server, DB2 and MySQL support this model. Other traditional models, such as hierarchical data models and network data models are still used in industry mainly on mainframe platforms. However, they are not commonly used due to their complexity. These are all referred to as traditional models because they preceded the relational model.

In recent years, the newer *object-oriented data models* were introduced. This model is a database management system in which information is represented in the form of objects as used in object-oriented programming. Object-oriented databases are different from relational databases which are table-oriented. Object-oriented database management systems (OODBMS) combine database capabilities with object-oriented programming language capabilities.

The object-oriented models have not caught on as expected so are not in widespread use. Some examples of object-oriented DBMSs are O2, ObjectStore and Jasmine.

1.8.2 Number of Users

A DBMS can be classification based on the number of users it supports. It can be a single-user database system, which supports one user at a time, or a *multiuser database system*, which supports multiple users concurrently.

1.8.3 Distribution

There are four main distribution systems for database systems and these, in turn, can be used to classify the DBMS.

i) **Centralized systems**

With a *centralized database system,* the DBMS and database are stored at a single site that is used by several other systems too. This is illustrated in Figure 1.2.

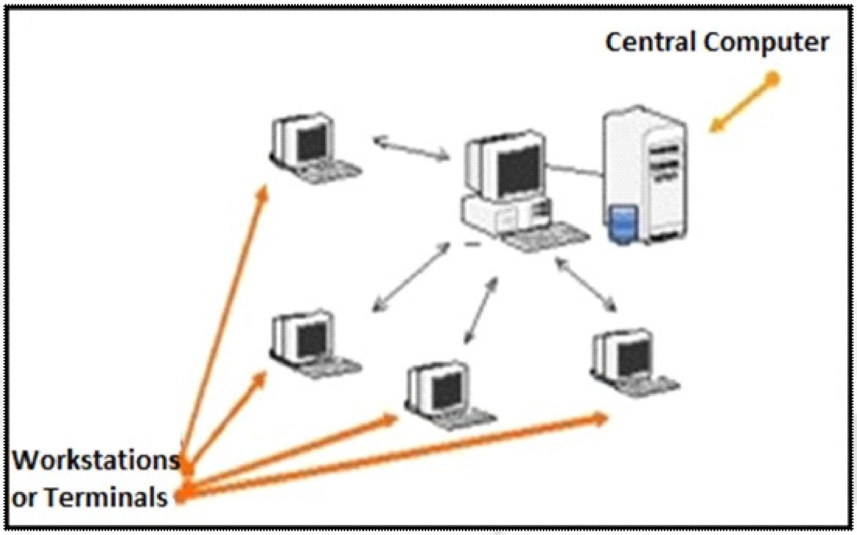


Figure 1.2: A centralized database system, by A. Watt

Please note that a database on a central computer can be accessed by many workstations, which may be either directly connected to central computer or may be connected over a computer network.

In the early 1980s, many Canadian libraries used the GEAC 8000 to convert their manual card catalogues to machine-readable centralized catalogue systems. Each book catalogue had a barcode field similar to those on supermarket products.

ii) **Distributed database system**

In a *distributed database system*, the actual database and the DBMS software are distributed from various sites that are connected by a computer network as shown in Figure 1.3. Thus, on each site a DBMS may be used.

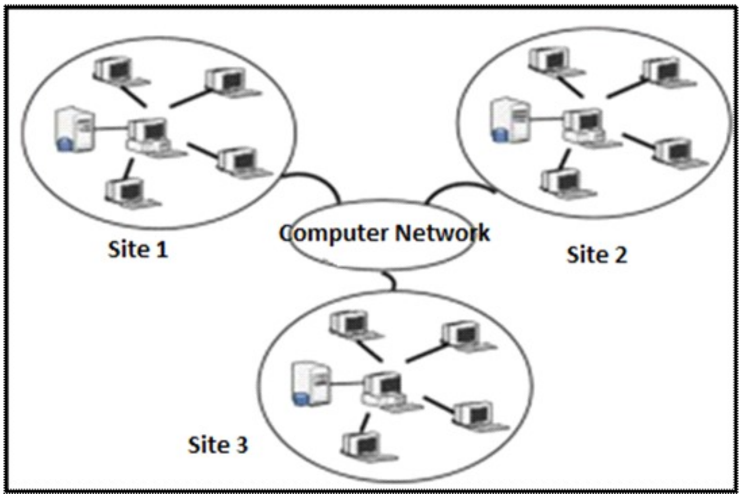


Figure 1.3: A distributed database system

iii) **Homogeneous distributed database systems**

Homogeneous distributed database systems use the same DBMS software from multiple sites. Data exchange between these various sites can be handled easily. For example, library information systems by the same vendor, such as Geac Computer Corporation, use the same DBMS software which allows easy data exchange between the various Geac library sites.

iv) **Heterogeneous distributed database systems**

In a heterogeneous distributed database system, different sites might use different DBMS software, but there is additional common software to support data exchange between these sites. For example, the various library database systems use the same machine-readable cataloguing (MARC) format to support library record data exchange.

1.8.4 Licensing

This applies in two different ways - which is either a database management system is open-source or proprietary. Open-source means that computer software with its source code is made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose. On the other hand, proprietary DBMSs creators and owners have all the rights to sell, modify or distribute.

1.8.5 Generic or Special Purpose

Generic DBMSs usage is not specific for any area of application i.e. they can be used in any application environment. However, special purpose databases are types of databases which specifically design for a special use.

Check Your Progress 3

Q1. What is a Relational Model?

....

....

....

Q2. What is the difference between centralized and distributed database systems?

....

....

....

Q3. What is open source software?

....

....

....

1.9 VIDEO LECTURE

https://tinyurl.com/gm2j93g

1.10 ACTIVITY

Activity 1.0

Introduction to Databases Exercise

Motivation: To become conversant with basic features of selected DBMS.

Resources: Internet access and Unit 1 learning material.

What to do:

- List down five open-source DBMSs and their application environment.
- List down ten proprietary DBMSs and their application environment.
- List down at least three single-user DBMSs.

How to do it:

By using Internet search engine the student will have to attempt this activity.

Feedback: The learner should discuss their work during the counselling session

1.11 SUMMARY

In this unit you learned about the difference between data and information, database management systems, their properties, and usage and application environment. Finally, you have also learned about classification of database management systems. You must also watch the supported video as this explain these concepts. This unit introduces you to these topics; however, you may go through the further readings for more details these topics.

1.12 ANSWERS

Check Your Progress 1

Ans 1 : Meta data is data about data. It contains the structure of data, data types of data and other details on data. You may refer to further readings for more details.

Ans 2 : (i) If two users or applications are trying to access the same data at the same time.

(ii) If two transaction are trying to access the same data at the same time.

Ans 3 : Redundancy refers to duplicity of data.

Check Your Progress 2

Ans 1 : You can query or process data of database to generate information.

Ans 2 : Student registration, fee payment, list of study centers/Regions, Programme structure, result storage, attendance management etc.

Ans 3 : Application programmer.

Check Your Progress 3

Ans 1 : It represents data in the form of tables; it allows enforcement of integrity constraints on the data; and also allows many data manipulation operations.

Ans 2 : In centralised database system data is stored on a central site, whereas in distributed database data is stored in several sites. Such distribution of data can lead to many issues, which can be studied from further readings.

Ans 3 : Open source software allows using code to study, change and distribute freely.

1.13 REVIEW QUESTIONS BY AUTHORS

1. Give detailed classification of databases. What
2. is the purpose of managing information?
3. Discuss the uses of databases in a business environment.

4. Why would you choose a database system instead of simply storing data in operating system files? When would it make sense not to use a database system?

5. What are the duties of a DBA? Describe the major difference between DBA responsibilities and application programmer duties in database environment.

6. List four types of users for database system in university applications, banking industry and hospitality industry; giving details for each role.

7. What are the properties of a database?

8. Give five database application environments, detailing each use for that particular application.

9. Describe four features (functions) you would expect to find in a DBMS.

1.14 FURTHER READING

1. Eng, N., & Watt, A. (2013). Database design. Retrieved July 08, 2016, from BC Open TextBooks, https://opentextbc.ca/dbdesign/chapter/chapter-3-characteristics-and-benefits-of-a-database/

Download this book for free at http://open.bccampus.ca

2. Eng, N., & Watt, A. (2013). Database design - 2nd edition. Retrieved July 08, 2016, from BC Open TextBooks, https://opentextbc.ca/dbdesign01/chapter/chapter-6-classification-of-database-systems/

Download this book for free at http://open.bccampus.ca

3. Classification of database systems. (2015, October 19). Retrieved May 27, 2016, from Open TextBooks for Hongkong, http://www.opentextbooks.org.hk/ditatopic/30720

4. Characteristics and benefits of a database. (2015, October 16). Retrieved May 27, 2016, from Open TextBooks for Hongkong, http://www.opentextbooks.org.hk/ditatopic/30646

1.15 ATTRIBUTION

The content of this unit of the Learning Material - Introduction to Databases (including its images, unless otherwise noted) is a derivative copy of materials from the book Database Design by Adrienne Watt and Nelson Eng licensed under Creative Commons Attribution 4.0 International License.

Download this book for free at http://open.bccampus.ca

The following material was written by Grace Mbwete:

1. Introduction

2. Database Usage and Environment

3. Activity1.0

4. Review questions

UNIT 2 DATABASE HISTROY

Structure

2.1 INTRODUCTION

Databases have existed for many years as computer technology was progressing between storage, memory, processor and networks. Database technology in itself has evolved between different eras of navigational, relational and post relational models. This unit will give us a background of why database approach came into existence and the limitation of the file-based systems which were in use before the advent of databases.

2.2 OBJECTIVES

Upon completion of this unit you should be able to:

- *Explain* how database technology has evolved.
- *Describe* the limitations of file-based systems.
- *Identify* different application areas of database

2.3 TERMINOLOGIES

Data : Single fact or a piece of information.

File-based systems : An application program designed to manipulate data files.

DBMS : A powerful software tool that allows you to store, manipulate and retrieve data in a variety of ways.

Database approach : Allows the management of large amounts of organizational information.

Concurrency : The ability of the database to allow multiple users access to the same record without adversely affecting transaction processing.

Data inconsistency : A situation where various copies of the same data are conflicting.

Data isolation : A property that determines when and how changes made by one operation become visible to other concurrent users and systems.

2.4 BEFORE THE ADVENT OF DATABASES

This section explains the file based systems, which were used prior to database systems.

2.4.1 File-based Systems

One way to keep information on a computer is to store it in permanent files. A company system has a number of application programs and each of these is designed to manipulate data files. These application programs are written as per the request of the users in the organization. New applications are added to the system as the need arises. The system just described is called the *file-based system*.

Consider a traditional banking system that uses the file-based system to manage the organization's data shown in Figure 2.1. As we can see, there are different departments in a bank. Each has its own applications that manage and manipulate different data files. For banking systems, the programs may be used to debit or credit an account, find the balance of an account, add a new mortgage loan and generate monthly statements.

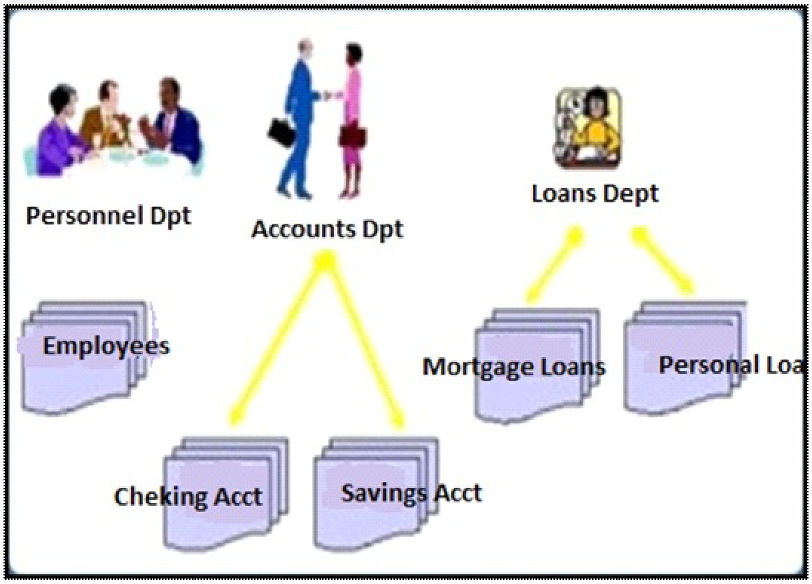


Figure 2.1: Example of a file-based system used by banks to manage customers' data, by A. Watt.

2.4.2 Limitations of the file-based system

Using the file-based system to keep organizational information has a number of disadvantages. Listed below are five examples.

i) **Data redundancy**

Often, within an organization, files and applications are created by different programmers from various departments over a long period of time. This can lead to *data redundancy*, a situation that occurs in a database when a field needs to be updated in more than one table. This practice can lead to several problems such as:

- ***Inconsistency in data format***

The same information being kept in several different places (files) *Data inconsistency*, a situation where various copies of the same data are conflicting, wastes storage space and duplicates effort.

ii) **Data isolation**

Data isolation is a property that determines when and how changes made by one operation become visible to other concurrent users and systems. This issue occurs in a concurrency situation. This is a problem because it is difficult for new applications to retrieve the appropriate data, which might be stored in various files.

iii) **Integrity problems**

A problem with data integrity is another limitation of using a file-based system. It refers to the maintenance and assurance that the data in a database are correct and consistent. Factors to consider when addressing this issue are:

- Data values must satisfy certain consistency constraints that are specified in the application programs.
- It is difficult to make changes to the application programs in order to enforce new constraints.

iv) **Security problems**

Security can be a problem with a file-based approach because:

- There are constraints regarding accessing privileges.
- Application requirements are added to the system in an ad-hoc manner so it is difficult to enforce constraints.

v) **Concurrency access**

Concurrency is the ability of the database to allow multiple users access to the same record without adversely affecting transaction processing. A file-based system must manage, or prevent, concurrency by the application programs. Typically, in a file-based system, when an application opens a file, that file is locked. This means that no one else has access to the file at the same time. In database systems, concurrency is managed thus allowing multiple users access to the same record. This is an important difference between database and file-based systems.

Check Your Progress 1

Q1. How is file based system different from database system?

....

....

....

....

Q2. What is Data isolation and data integrity?

....

....

....

....

Q3. Are file based systems more secure than Database approach? Justify, your answer.

....

....

....

....

2.5 DATABASE APPROACH

The difficulties that arise from using the file-based system have prompted the development of a new approach in managing large amounts of organizational information called the *database approach*. Databases and database technology play an important role in most areas where computers are used, including business, education and medicine. To understand the fundamentals of database systems, we will start by introducing some basic concepts in this area.

2.5.1 Role of Database in Business

Everyone uses a database in some way, even if it is just to store information about their friends and family. That data might be written down or stored in a computer by using a word-processing program or it could be saved in a spreadsheet. However, the best way to store data is by using *database management software*. This is a powerful software tool that allows you to store, manipulate and retrieve data in different ways.

Most companies keep track of customer information by storing it in a database. This data may include customers, employees, products, orders or anything else that assists the business with its operations.

2.5.2 Data

Data are factual information such as measurements or statistics about objects and concepts. We use data for discussions or as part of a calculation. Data can be a person, a place, an event, an action or any one of a number of things. A single fact is an element of data, or a *data element.*

If data are information and information is what we are in the business of working with, you can start to see where you might be storing it. Data can be stored in:

- Filing cabinets
- Spreadsheets
- Folders
- Ledgers
- Lists
- Piles of papers on your desk

All of these items store information, and so too does a database. Because of the mechanical nature of databases, they have terrific power to manage and process the information they hold. This can make the information they house much more useful for your work.

Check Your Progress 2

Q1. Can word processing can be categorized as database approach?

....

....

....

....

Q2. What are the basic features of database management?

....

....

....

....

Q3. Which tools is most useful for processing data to create information?

....

....

....

....

2.6 ACTIVITY

Activity 2.0

Data History Activity

Motivation: To differentiate the use of database application and spreadsheets.

Resources: Unit 2 learning materials, selected DBMS software and spreadsheet software.

What to do: Answer the following questions:

1. List down and describe each characteristic of the database approach.
2. What are the major parts of a database system?
3. What is data redundancy, and which characteristics of the file system can lead to it?
4. What are some disadvantages of database systems?
5. List some of the functions of a DBMS.
6. Why is a spreadsheet NOT a database?

Duration: Expect to spend about 2 hours on this activity.

Feedback: The learner should discuss their work during the counselling session.

2.7 SUMMARY

In this unit you learned about the evolution of database technology -- starting from the usage of file-based systems and the eventual advent of databases. This unit also, gave an overview of meaning of data and the role of databases in a business.

2.8 ANSWERS

Check Your Progress 1

Ans 1. A file based system can keep a copy of data in its local file and any updates on data may not be available immediately to other users. Database applications, in general, allow controlled redundancy and shares data consistently among various applications.

Ans 2. Data Isolation determines when the changes made by one program/ user would be visible to other programs/users. Data integrity ensures the correctness of data.

Ans 3. No, as security in file based system is to be enforced on multiple copies of data, therefore, it is difficult to implement.

Check Your Progress 2

Ans 1. No, typically you create multiple files with duplicate data.

Ans 2. Database management allows you to store, manipulate and retrieve data easily.

Ans 3. DBMS has powerful features to process data into information.

2.9 REVIEW QUESTIONS BY AUTHOR

1. Compare and contrast between data and information.
2. Explain the differences between file and record.
3. Explain the role and significance of databases in any society.
4. Why is it important to avoid data redundancy?
5. Select and briefly explain three disadvantages of the file based approach that is resolved by the database approach.
6. Explain the difference between data and information.
7. One of the advantages of database systems over file-based systems is said to be the concurrency control. Explain this term and why it is an advantage.

2.10 FURTHER READING

1. Watt, A., &Eng, N. (2013). Database Design - 2nd Edition. Retrieved May 27, 2016, from BC Open TextBooks, https://opentextbc.ca/dbdesign/chapter/chapter-1-before-the-advent-of-database-systems/

Download this book for free at http://open.bccampus.ca

2. Before the advent of database systems. (2016, January 19). Retrieved May 23, 2016, from Open TextBooks for Hongkong, http://www.opentextbooks.org.hk/ditatopic/30633

Download this book for free at http://open.bccampus.ca

3. Open, T. (2016). ICTs: E-government. Retrieved May 27, 2016, from OpenLearn - The Open University, http://www.open.edu/openlearn/science-maths-technology/computing-and-ict/information-and-communication-technologies/icts-e-government/content-section-3.2

Download this book for free at http://open.bccampus.ca

2.11 ATTRIBUTION

The content of this unit of the Learning Material - Introduction to Databases (including its images, unless otherwise noted) is a derivative copy of materials from the book Database Design by Adrienne Watt and Nelson Eng licensed **under Creative Commons Attribution 4.0 International License**.

Download this book for free at http://open.bccampus.ca

The following material was written by Grace Mbwete:

1. Introduction
2. Database Usage and Environment
3. Activity2.0
4. Summary
5. Review questions

UNIT 3 DATA MODELLING

Structure

3.1 INTRODUCTION

In the previous unit, the basic terminology related to DBMS has been explained. In addition, the advantages of using DBMS have also been given. Database Architecture plays a significant role in ensuring proper creation and management of a database. Data which is stored in a database can be seen from different levels. In this unit you will be introduced to the levels of database architecture and learn how to apply them in order to describe the structure of the database.

3.2 OBJECTIVES

On completion of this unit you will be able to:

- *Explain* the basic concepts of data modelling.
- *Identify* different levels of database architecture as viewed by users.
- Explain the concept of data independence and its importance in a database system.
- *Compare and contrast* between logical and physical data independence.

3.3 TERMINOLOGIES

Conceptual Model : The logical structure of the entire database.

Data independence : The immunity of user applications to changes made in the definition and organization of data in application program designed to manipulate data files.

Data model : A collection of concepts or notations for describing data, data relationships, data semantics and data constraints.

Database logical design : Defines a database in a data model of a specific DBMS is the first step in the process of database design.

Database physical design : Defines the internal database storage structure, file organization and indexing techniques.

Physical Data independence : The immunity of the internal model to changes in the physical model.

Logical Data Independence : The ability to change the logical schema without changing the external schema.

3.4 DATA MODELLING

One of the key requirements for a good database is to identify the data that would be stored in it. A good way of doing it is through data modelling.

3.4.1 Introduction to Data Modelling

Data modeling is the first step in the process of database design. This step is sometimes considered to be a high-level and abstract design phase, also referred to as conceptual design. The aim of this phase is to describe:

- The data contained in the database (e.g., entities: students, lecturers, courses, subjects)
- The relationships between data items (e.g., students are supervised by lecturers; lecturers teach courses)
- The constraints on data (e.g., student number has exactly eight digits; a subject has four or six units of credit only)

In the second step, the data items, the relationships and the constraints are all expressed using the concepts provided by the high-level data model. Because these concepts do not include the implementation details, the result of the data modeling process is a (semi) formal representation of the database structure. This result is quite easy to understand so it is used as reference to make sure that all the user's requirements are met.

The third step is database design. During this step, we might have two sub-steps: one called *database logical design* which defines a database in a data model of a specific DBMS, and another is called *database physical design* which defines the internal database storage structure, file organization or indexing techniques. Finally, a database is implemented, which includes implementing the database operations and user interfaces.

In the database design phases, data are represented using a certain *data model*. The data model is a collection of concepts or notations for describing data, data relationships, and data semantics and data constraints. Most data models also include a set of basic operations for manipulating data in the database.

3.4.2 Degree of Data Abstraction

In this section, we will look at the database design process in terms of specificity. Database design starts at a high level and proceeds to an ever-increasing level of detail. For example, when building a home, you start with how many bedrooms and bathrooms the home will contain -- whether it will be on one level or multiple levels, etc. The next step is to get an architect to design the home from a more structured perspective. This level gets more detailed with respect to actual room sizes, how the home will be wired, where the plumbing fixtures will be placed, etc. The last step is to hire a contractor to build the home. That's looking at the design from a high level of abstraction to an increasing level of detail.

The database design is very much like that. It starts with users identifying the business rules; then the database designers and analysts create the database design; and then the database administrator and application developers implement the design using a DBMS.

Check Your Progress 1

Q1. What is the role of data modelling?

....

....

....

....

Q2. What will be the data model of a system that allows registration by students into courses?

....

....

....

....

Q3. What is meant by degree of abstraction?

....

....

....

....

3.5 DATABASE ARCHITECTURE LEVELS (VIEWS)

As it was detailed above, database architecture provides a base for the database to have capability, reliability, effectiveness and efficiency in meeting user requirements. In addition, a correct database architecture should provide data independence and ease of design. In order for the design to meet these requirements, sophisticated database architecture is used, providing a number of levels of data abstraction or data definition. The levels (view) are listed as follows:

3.5.1 External Level

External level or model represents the end user's view of the database. It contains multiple different external views where each view represents a specific business unit of an organisation. This is a view which is closely related to the real world as perceived by each user. This model is aimed at facilitating application program development and assisting database designer in identifying data for each user requirement.

3.5.2 Conceptual Level

Conceptual level provides flexible data-structuring capabilities and a "community view": the logical structure of the entire database. This level has the following details:

i. It contains data stored in the database

ii. It shows relationships among data including:

- Constraints
- Semantic information (e.g., business rules)
- Security and integrity information

iii. It considers a database as a collection of entities (objects) of various kinds.

iv. It forms the basis for identification and high-level description of main data objects.

3.5.3 Internal Level

The Internal level adapts the conceptual level design to a specific DBMS which makes it software-dependent. The three best-known examples of internal view are the relational data model, the network data model and the hierarchical data model. This internal level:

i. considers a database as a collection of fixed-size records

ii. are closer to the physical level or file structure

iii. are representations of the database as seen by the DBMS.

iv. requires the designer to match the conceptual model's characteristics and constraints to those of the selected implementation model.

v. involve mapping the entities in the conceptual model to the tables in the relational model.

3.5.4 Physical Level

Physical level is the physical representation of the database and the lowest level of abstractions. Physical level is specifically based on:

i. How the data is stored. It deals with :

- Run-time performance
- Storage utilization and compression
- File organization and access methods
- Data encryption

ii. The physical level is managed by the operating system (OS)

iii. Provide concepts that describe the details of how data are stored in the computer's memory.

iv. Software and hardware dependent

Check Your Progress 2

Q1. What is a conceptual level in Database Architecture?

....

....

....

....

Q2. Which of the levels in Data Architecture are software dependent?

....

....

....

....

Q3. You want to make your database more efficient, which level would you modify.

....

....

....

....

3.6 DATA ABSTRACTION LAYER

In a pictorial view, you can see how the different models work together. Let's look at this from the highest level -- i.e. the external model.

The external model is the end user's view of the data. Typically a database is an enterprise system that serves the needs of multiple departments. However, one department is not interested in seeing other departments' data (e.g., the human resources (HR) department does not care to view the sales department's data). Therefore, one user's view will differ from another user.

The external model requires that the designer subdivide a set of requirements and constraints into functional modules that can be examined within the framework of their external models (e.g., human resources versus sales).

As a data designer, you need to understand all the data so that you can build an enterprise-wide database. Based on the needs of various departments, the conceptual model is the first model created.

At this stage, the conceptual model is independent of both software and hardware. It does not depend on the DBMS software used to implement the model. It does not depend on the hardware used in the implementation of the model. Changes in either hardware or DBMS software have no effect on the database design at the conceptual level.

Once a DBMS is selected, you can then implement it. This is the internal model. Here you create all the tables, constraints, keys, rules, etc. This is often referred to as the *logical design*. The physical model is simply the way the data is stored on disk. Each database vendor has its own way of storing the data. Figure 3.1 illustrates the data abstraction layers.

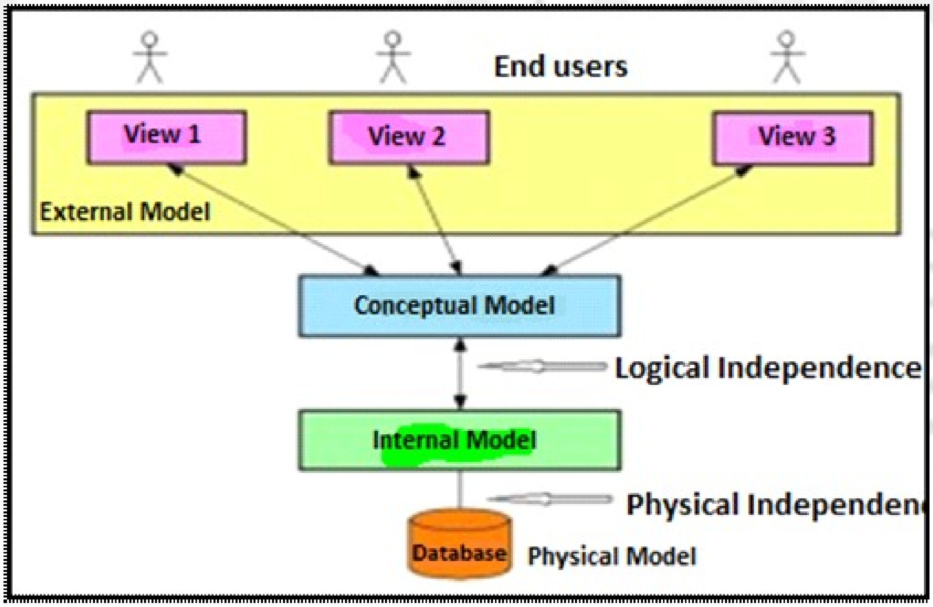


Figure 3.1: Data abstraction layers, by A. Watt

3.7 SCHEMA

A *schema* is an overall description of a database, and it is usually represented by the *entity relationship diagram (ERD)*. There are many sub-schemas that represent external models and thus display external views of the data. Below is a list of items to consider during the design process of a database.

i. External schemas: there are multiples of these.

ii. Multiple sub-schemas: these display multiple external views of the data.

iii. Conceptual schema: there is only one. These schemas include data items, relationships and constrain one of this.

3.8 DATA INDEPENDENCE

Data independence refers to the immunity of user applications to changes made in the definition and organization of data. Data abstractions expose only those items that are important or pertinent to the user. Complexity is hidden from the database user. Data independence and operation independence together form -- logical and physical data independence.

3.8.1 Logical data independence

A *logical schema* is a conceptual design of the database done on paper or a whiteboard -- much like architectural drawings for a house. The ability to change the logical schema, without changing the external schema or user view, is called *logical data independence*. For example, the addition or removal of new entities, attributes or relationships to this *conceptual schema* should be possible without having to change existing external schemas or rewrite existing application programs. In other words, changes to the logical schema (e.g., alterations to the structure of the database like adding a column or other tables) should not affect the function of the application (external views).

3.8.2 Physical data independence

Physical data independence refers to the immunity of the internal model to changes in the physical model. The logical schema stays unchanged even though changes are made to file organization or storage structures, storage devices or indexing strategy.

Physical data independence deals with hiding the details of the storage structure from user applications. The applications should not be involved with these issues, since there is no difference in the operation carried out against the data.

Check Your Progress 3

Q1. What is logical design?

....

....

....

....

Q2. How many conceptual schema & external schemas can be created for a single databases system?

....

....

....

....

Q3. What is logical data independence and physical data independence?

....

....

....

....

3.9 VIDEO LECTURE

https://tinyurl.com/zhng6at

3.10 ACTIVITY

Activity 3.0

Database Architecture and Data Languages Activity

Aim: To develop knowledge of database architecture.

Resources: Unit 3 learning materials and word processor software.

What to do:

- Draw the 'three-tier' database architecture. Discuss the functions of each layer, and define the necessary tools involved.
- Draw 'client-server architecture' which is commonly used to implement a database system.

Duration: Expect to spend about 1 hour on this activity

Feedback: The learner should discuss their work during the counselling session

3.11 SUMMARY

In this unit you learned on the basic feature of database architecture. The content included data abstraction layers which included internal view, external view, physical view and conceptual view. In addition, you have also learned about the concept of data independence and the difference between logical and physical data independence.

3.12 ANSWERS

Check Your Progress 1

Ans 1. Data Modelling describes the data of database, Relationships and constraints.

Ans 2. Entities: Student, Courses

Relationships: Student *registers* for a course.

Constraints: Student_ID would be unique, qualifications and control number are also needed. Each course will be identified by unique course code etc.

Ans 3. Degree of abstraction refers to the level of details being provided for database design. It may start at high level data model to low level implementation details.

Check Your Progress 2

Ans 1. Conceptual level defines:

- Data that is to be stored
- Business rules or constraints
- Security and integrity information

Ans 2. Internal Level

Ans 3. Physical level.

Check Your Progress 3

Ans 1. Logical design includes design of tables, keys, constraints and business rules.

Ans 2. One conceptual schema, but multiple external schemas.

Ans 3. The ability to change logical schema without changing external schema is logical data independence.

The ability to change physical schema without changing logical schema is physical data independence.

3.13 REVIEW QUESTIONS BY AUTHORS

1. Describe the purpose of a conceptual design.
2. How is a conceptual design different from a logical design?
3. Compare and contrast between external view and conceptual view.
4. Compare and contrast between internal view and a physical view.
5. Which level of the database architecture does the database administrator work with?
6. Which level of the database architecture does the application programmer work with?

7. What is logical data independence and why is it important?

8. What is physical data independence?

9. Explain the difference between external, internal, and conceptual schemas. How are these different schema layers related to the concepts of logical and physical data independence?

10. Describe in details the three-tier database architecture.

3.14 FURTHER READING

1. Watt, A., & Eng, N. (2013). Database Design - 2nd Edition. Retrieved May 27, 2016, from BC Open TextBooks, https://opentextbc.ca/dbdesign/chapter/chapter-5-data-modelling/

Download this book for free at http://open.bccampus.ca

2. Data Modelling. (2015, October 19). Retrieved May 27, 2016, from Open TextBooks for Hongkong, http://www.opentextbooks.org.hk/ditatopic/30674

3.15 ATTRIBUTION

The content of this unit of the Learning Material - Introduction to Databases (including its images, unless otherwise noted) is a derivative copy of materials from the book Database Design by Adrienne Watt and Nelson Eng licensed under **Creative Commons Attribution 4.0 International License.**

Download this book for free at http://open.bccampus.ca

The following material was written by Grace Mbwete:

1. Introduction

2. Activity 3.0

3. Summary

4. Review questions

UNIT 4 DATA MODELS

Structure

4.1 INTRODUCTION

Data Models provide the logical structure of data and fundamentally determines the manner in which data can be stored, organized, and manipulated. Therefore, in this unit, we will have an opportunity of learning the overview of database models, their various categories and structure for each.

4.2 OBJECTIVES

Upon completion of this unit you should be able to:

- *Explain* the basic concepts behind data models.
- *Categorize different types of data models.*
- *Describe* the basic differences between relational, hierarchical, object-oriented and network database models.

4.3 TERMINOLOGIES

Data Model : Logical structure of data.

Hierarchical Model : A data model which represents data as a hierarchical tree structure.

Instance : A record within a table.

Network Model : A data model which represents data as record types.

Relation : Another term for a table.

4.4 OVERVIEW

A database model defines the way the data would be structured. This section defines a categorisation of such models.

4.4.1 What is a Database Model?

A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized, and manipulated.

4.4.2 Categories of Database models

Database model can be categorized as high-level conceptual data models and record-based data models.

i) **High-level Conceptual Data Models**

High-level conceptual data models provide concepts for presenting data in ways that are close to the way people perceive data. A typical example is the entity relationship model which uses main concepts like entities, attributes and relationships. An entity represents a real-world object such as an employee or a project. The entity has attributes that represent properties such as an employee's name, address and birth-date. A relationship represents an association among entities -- for example, an employee *works on* many projects. A relationship *works on* exists between the employee and each project. These models are discussed in Block 2 of this course.

ii) **Record-based Logical Data Models**

Record-based logical data models provide concepts that users can understand but are not too far from the way data is stored in the computer. Three well-known data models of this type are relational data models, network data models and hierarchical data models.

Check Your Progress 1

Q1. What is a database model?

....

....

....

....

Q2. Give an example of high level conceptual database model.

....

....

....

....

Q3. Which database model defines the storage structure?

....

....

....

....

4.5 TYPES OF DATABASE MODELS

The following are the most common database used in the industry. Each database model has its own characteristics which defines their application environment.

4.5.1 Hierarchical database Model

A hierarchical database model is a data model in which the data is organized into a tree-like structure. The data is stored as records which are connected to one another through links. A record is a collection of fields, with each field containing only one value. The entity type of a record defines which fields the record contains. A record in the hierarchical database model corresponds to a row (or tuple) in the relational database model and an entity type corresponds to a table (or relation)

Figure 4.1 illustrates hierarchical model for a 'university 'database where there are parent and child entities showing the hierarchy structure of the model.

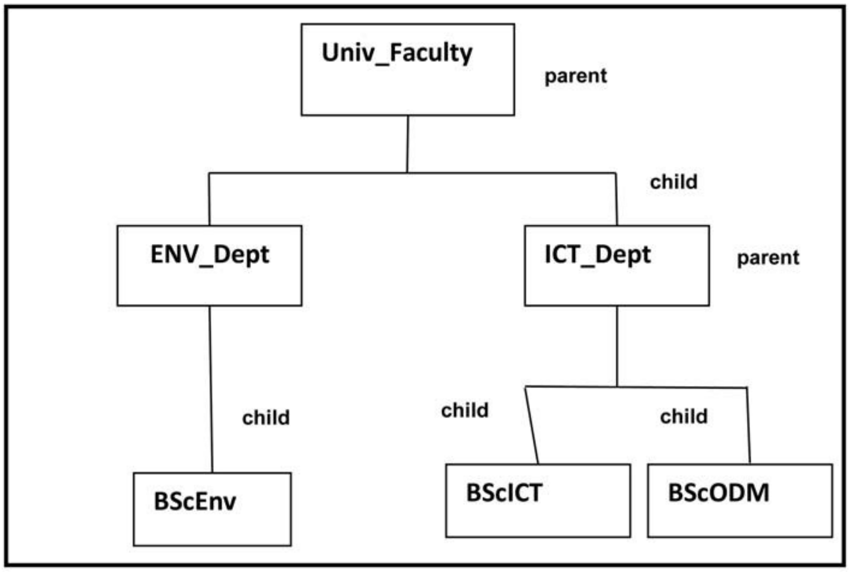


Figure 4.1: Example of a hierarchical model, by G. Mbwete (2016)

4.5.2 Network Database Model

The network model is a database model conceived as a flexible way of representing objects and their relationships. Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy or lattice. Figure 4.2 shows an example of a '*chain store*' database in a form of network database model.

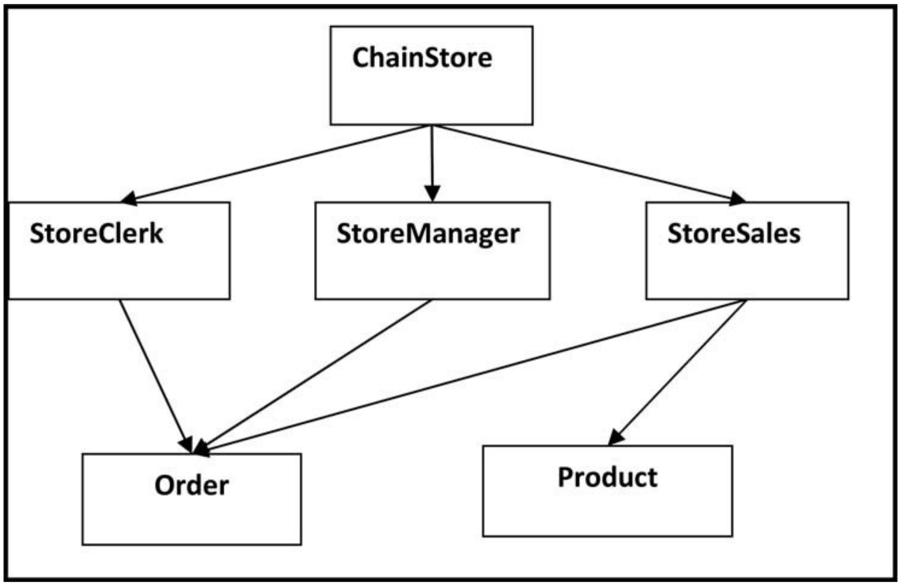


Figure 4.2: Example of a Network model, by G. Mbwete

4.5.3 Relational Database Model

The *relational model* represents data as relations, or tables. This is the most common model used in the database industry today. A relational database is based on the relational model developed by E.F. Codd in 1970s. It is explained in the next unit in details.

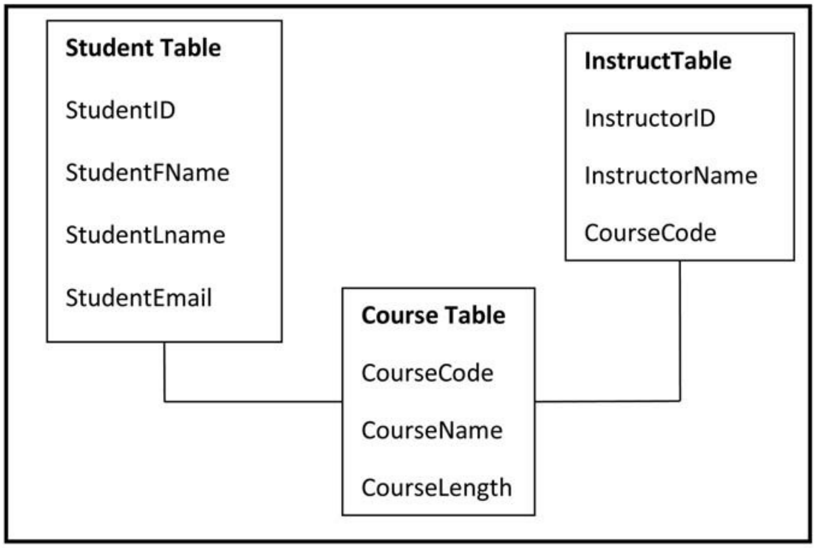


Figure 4.3: Illustration of a relational database model, by G. Mbwete

4.5.4 Object Oriented Database Model

Object-oriented database model conceptualize a database as a collection of objects, or reusable software elements, with associated features and methods. Multimedia and hypertext database are one the features of object-oriented model. This is also known as post-relational database model, since it incorporates tables, but isn't limited to tables; but more features as well. Figure 4.4 illustrates an object 'customer' with its associated attributes.

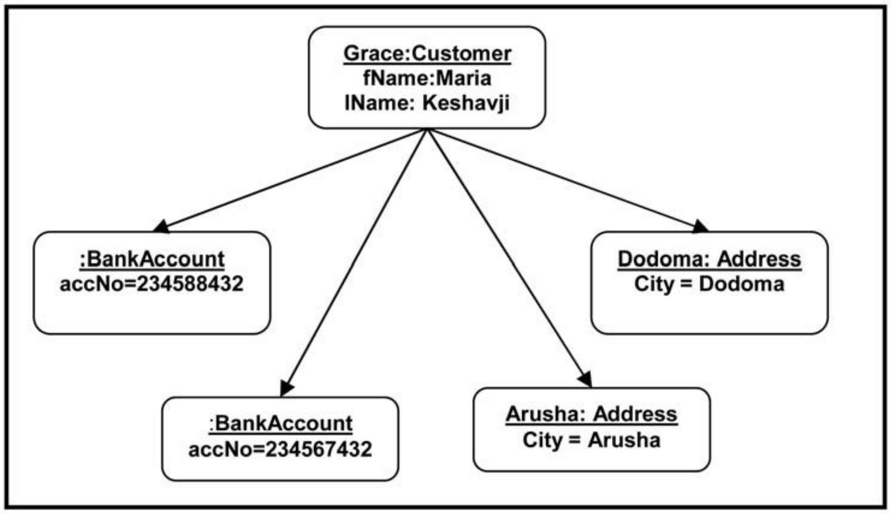


Figure 4.4: Illustration of a object-oriented data model, by G. Mbwete

Check Your Progress 2

Q1. Differentiate between hierarchical model and network model.

....

....

....

....

Q2. In what kind of database applications, object oriented models can be used?

....

....

....

....

Q3. Which is the most used database models?

....

....

....

....

4.6 ACTIVITY

Activity 4.0

Database Model Exercises

Aim: Compare and contrast between different database models.

Resources: Unit 4 learning materials and selected DBMS software.

What to do: Answer the following questions:

1. Explain the basic features of relational database model.
2. Are there any emerging database models apart from relational databases? Justify your answer.
3. What is object-relational database model? How common is model being used in the IT industry today?

Duration: Expect to spend about 2 hours on this activity.

Feedback: The learner should discuss their work during the counselling session

4.7 SUMMARY

In this unit you learned about database models and their categories. In addition, you have also learned about different types of database models which have formed the basis of databases in use in the industry.

4.8 ANSWERS

Check Your Progress 1

Ans 1. It defines a logical structure of a database. It determines how data would be stored, organized and manipulated.

Ans 2. Entity-Relationship model.

Ans 3. Record-based logical data model.

Check Your Progress 2

Ans 1. Hierarchical model represents a hierarchy, whereas network model represents a complex network of elements. Links in a network model need not be a lattice as the case for hierarchical model.

Ans 2. Hypertext and multimedia based databases.

Ans 3. Relational database model.

4.9 FURTHER READING

1. Eng, N., & Watt, A. (2013). Database design - 2nd edition. Retrieved May 23, 2016, from BC Open TextBooks, https://opentextbc.ca/dbdesign01/chapter/chapter-4-types-of-database-models/

Download this book for free at http://open.bccampus.ca

2. Types of Database Models. (2015, October 19). Retrieved May 23, 2016, from Open TextBooks for Hongkong, http://www.opentextbooks.org.hk/ditatopic/30671

4.10 ATTRIBUTION

The content of this unit of the Learning Material - Introduction to Databases (including its images, unless otherwise noted) is a derivative copy of materials from the book Database Design by Adrienne Watt and Nelson Eng licensed under **Creative Commons Attribution 4.0 International License.**

Download this book for free at http://open.bccampus.ca

The following material was written by Grace Mbwete:

1. Introduction
2. Activity4.0
3. Summary

UNIT 5 RELATIONAL DATA MODEL

Structure

5.1 INTRODUCTION

The relational data model provides techniques of structuring how data can be perceived. Relational Data Model provides conceptual view of data, which is not provided by the E-R model. It usually represents the relations, integrity constraints and other details of a database. The relational data model assumes entities as tables and allows operations to be performed on them. In this unit, you will learn about relations.

5.2 OBJECTIVES

Upon completion of this unit you should be able to:

- *Describe the fundamental concepts and notations in relational data model.*
- *Explain* the various types of relationships in the Relational Data Model.
- *Describe* the basic properties of a table.

5.3 TERMINOLOGIES

Relation : Subset of the Cartesian product of a list of domains characterized by a name. Technical name for table or file.

SQL : Structured Query Language (SQL): the standard database access language.

Domain : The original sets of atomic values used to model data; A domain contains a set of acceptable values that a column is allowed to have.

Atomic value : Each value in the domain is indivisible as far as the relational model is concerned.

Table : *The same as relation or an entity or instance.*

5.4 OVERVIEW

The relational data model was introduced by C. F. Codd in 1970. Currently, it is the most widely used data model. The relational model has provided the basis for:

i. Research on the theory of data/relationship/constraint

ii. Numerous database design methodologies

iii. The standard database access language called *structured query language (SQL)*

iv. Almost all modern commercial database management systems.

The relational data model describes the world as "a collection of inter-related relations (or tables)."

5.5 FUNDAMENTAL CONCEPTS IN RELATIONAL MODEL

This section explains some of the basic terminology used in the context of relational database management system.

5.5.1 Relation/Table

A *relation*, also known as a *table*, is a subset of the Cartesian product of a list of domains characterized by a name, and within a table, each row represents a group of related data values. A *row*, or record, is also known as a tuple. The column in a table is a field and is also referred to as an attribute. You can also think of it this way: an attribute is used to define the record or *tuple*. and a record or contains a set of attributes.

The steps below outline the logic between a relation and its domains.

i. Given *n* domains, which are denoted by D1, D2, … Dn

ii. And *r* is a relation defined on these domains

iii. Then $r \subseteq D1 \times D2 \times \ldots \times Dn$

This is the formal definition of a relation, which is the foundation for the relational model. It is explained with the help of the following example:

Consider a relation r1 consisting of two domains D1 with values a and b and domain D2 with values c and d, then the cross product of D1×D2 will be as under:

D1×D2	
a	c
a	d
b	c
b	d

A relation can be any subset of D1×D2. However, databases in relational model are represented using tables. A database is composed of multiple tables and each table holds the data. Table 5.1 illustrates a relation known as 'Course' which contains details of a University 'course'.

Table 5.1: 'Course' relation

Course Code	CourseName	CourseInstructor
OIT217	Database Design	Miss Grace
OIT216	System Security	Mr Raitton
OIT218	Web Programming	Mr Ombeni

This table can be expressed as a relations Course (CourseCode, CourseName, CourseInstructor). The Course is the name of the relation, and CourseCode, CourseName and CourseInstructor are defined over specific domains. For example, CourseCode domain is the list of all the course codes of the specific University. It may be noted that CourseCode will take values from its specific domain only.

Check Your Progress 1

Q1. What is a relation?

....

....

....

....

Q2. Give one example of relation in terms of domains.

....

....

....

....

Q3. Determine the expected domain of Teacher from the following table:

Teacher	Subject
X	S1
X	S2
Y	S3
Z	S1
A	S4
X	S5

....

....

....

....

5.5.2 Column/Field/Attribute

A database stores pieces of information or facts in an organized way. Understanding how to use and get the most out of databases requires us to understand that method of organization.

The principal storage units are called columns or *fields* or *attributes*. These house the basic components of data into which your content can be broken down. When deciding which fields to create, you need to think generically about your information, for example, drawing out the common components of the information that you will store in the database and avoiding the specifics that distinguish one item from another.

Table 5.2 illustrates the attributes (characteristics) for 'Course' relation/table. This relation has four columns/field/attributes namely: Course_Code, Course_Name, Course_Instructor and Course_Unit which give characteristics of a 'course' table. In table 5.2 they are shown in bold. The rest are the values of these attributes, with each row representing values of the attributes for one specific course.

Table 5.2: Illustration of columns of a 'Course' table

Course_Code	**Course_Name**	**Course_Instructor**	**Course_Unit**
OIT217	Database Design	Miss Grace	2
OIT216	System Security	Mr Raitton	1
OIT218	Web Design	Mr Ombeni	3

5.5.3 Domain

A *domain* is the original sets of atomic values used to model data. By *atomic* value, we mean that each value in the domain is indivisible as far as the relational model is concerned. The following are examples:

i. The domain of date of birth has a set of numbers in any date format.

ii. The domain of Marital Status has a set of possibilities: Married, Single, Divorced.

iii. The domain of Shift has the set of all possible days: {Mon, Tue, Wed…}.

iv. The domain of Salary is the set of all floating-point numbers greater than 0 and less than 2,00,00,000.

v. The domain of First Name is the set of character strings that represents names of people.

In summary, a domain is a set of acceptable values that a column is allowed to contain. This is based on various properties and the data type for the column.

5.5.4 Records/Row/Tuples

Just as the content of any one document or item needs to be broken down into its constituent bits of data for storage in the fields, the link between them also needs to be available so that they can be reconstituted into their whole form. Records allow us to do this. Records contain fields that are related, such as a customer or an employee. As noted earlier, a tuple is another term used for record.

Records and fields form the basis of all databases. A simple table gives us the clearest picture of how records and fields work together in a database storage project. Table 5.3 below shows 'Course' relation which *'five'* tuples/records.

Table 5.3: Illustration of rows/tuples of 'Course' relation

Course_Code	**Course_Name**	**Course_Instructor**	**Course_Unit**
OIT217	Database Design	Ms Grace Mbwete	2
OIT216	System Security	Mr EliahLukwaro	1
OIT218	Web Programming	Mr Ombeni Mathias	3
OIT313	Network	Mrs Lilian Charles	4
OIT314	Programming	Mr Said Ally	2

5.5.5 Degree

The *degree* is the number of attributes/columns/fields in a table or relation. In the example below in table 5.4, for a 'Course' table, the degree is 4.

Table 5.4: Illustration of degree of a table - 'Course table'

Course_Code	Course_Name	Course_Instructor	Course_Unit
OIT217	Database Design	Miss Grace Mbwete	2
OIT216	System Security	Mr EliahLukwaro	1
OIT218	Web Programming	Mr Ombeni Mathias	3
OIT313	Network	Mrs Lilian Charles	4
OIT314	Programming	Mr Said Ally	2

5.6 PROPERTIES OF A TABLE

Tables in a relational data model have the following major characteristics:

i. A Table has a name that is distinct from all other tables in the database.

ii. There are no duplicate rows, each row is distinct.

iii. Entries in columns are atomic (no repeating groups or multivalued attributes)

iv. Entries from columns are from the same domain based on their data type:

- Number (numeric, integer, float, smallint,…)
- Character (string)

- Date
- Logical (true or false)

v. Operations combining different data types are disallowed.

vi. Each attribute has a unique/distinct name.

vii. The order of columns in a table is not important.

viii. The order of rows is not important.

ix. Relational model is also characterised by the operations that can be performed on relations, however, they are beyond the scope of this Unit.

Check Your Progress 2

Consider a University, which maintains the following information about its student:

Student id, name, date of birth, age, phone number.

Now answer the following questions for the structure as given above:

Q1. List the attributes that would be required for making a table.

....

....

....

....

Q2. Insert two records in the table, whose structure is created in Q1.

....

....

....

....

Q3. What is the degree of the relation given in Q1.

....

....

....

....

5.7 VIDEO LECTURE

https://tinyurl.com/jxqzavf

5.8 ACTIVITY

Activity 5.0

Relational Database Model Exercises

Aim: To become conversant with basic Relational Database Model.

Resources: Unit 5 learning materials.

What to do: Study table 5.5 below and answer the following questions:

1. Using correct terminology, identify and describe all the components in Table 5.5.
2. What is the possible domain for field Staff_Office?
3. How many fields are shown? What is the degree of the table?
4. How many tuples does the table have?

Table 5.5: Staff Table

Staff_ID	Staff_First Name	Staff_Last Name	Staff_Office	Staff_Dept
608	Grace	Mbwete	ODL-908	ICT
702	Elia	Ahidi	Block E-ICT	ICT
1002	Lillian	Charles	ODL- 806	Nutrition
2006	Cathy	Gerald	Block C- 202	P/Science
3117	Martin	Clemence	Bungo-101	Nutrition

Duration: Expect to spend about 1/2 hours on this activity.

Feedback: The learner should discuss their work during the counselling session

5.9 SUMMARY

In this unit you have learnt about Relational Data Model which was developed by C.F. Codd in the 1970s. The unit content had covered fundaments concepts behind Relational Database Model which includes details on the relations/tables and its properties (fields/columns, tuples/rows, degree and domain).

You may refer to further readings for more details on relational model.

5.10 ANSWERS

Check Your Progress 1

Ans 1. A relation is a proper subset of Cartesian product of domains.

Ans 2. Consider the two domains Course Name and Teacher:

Course Name
DBMS
ARCHITECTURE

Teacher
X
Y

The relation will be any subset of the cross product.

Course Name	Teacher
DBMS	X
DBMS	Y
ARCHITECTURE	X
ARCHITECTURE	Y

For example,

Course Name	Teacher
DBMS	Y
ARCHITECTURE	X

is a valid relation which represents that course name DBMS is taught by Teacher Y and ARCHITCTURE is taught by Teacher X.

Ans 3. Domain of Teacher is {X, Y, Z, A}

Check Your Progress 2

Ans 1. Student {Student_ id, name, dob, phone}

Please note as age can be computed from date of birth (dob), therefore, it has not been included as an attribute in the relation named Student.

Ans 2.

Student_id	Name	Dob	phone
S01	Mohan	22-02-1991	9123456780
S02	Hanif	22-05-1993	1234567890

Ans 3. 4

5.11 REVIEW QUESTIONS BY AUTHORS

1. Define the term relational database schema (the schema of a whole relational database).
2. Define the term relation schema key. List the properties of a relation schema key.
3. What is the difference between a relation and a relation schema, and what is their relationship?

4. Define the structure of a relational database schema.

With examples, please explain the following terms:

- Relation
- Domain
- Attribute
- Attribute domain
- Relation instance
- Relation cardinality
- Relation degree

5.12 FURTHER READINGS

1. The Relational Data Model. (2015, October 19). Retrieved May 27, 2016, from Open TextBooks for Hongkong, http://www.opentextbooks.org.hk/ditatopic/30728

2. Eng, N., & Watt, A. (2013). Database design - 2nd edition. Retrieved May 27, 2016, from BC Open TextBooks, https://opentextbc.ca/dbdesign01/chapter/chapter-7-the-relational-data-model/

Download this book for free at http://open.bccampus.ca

3. Lecture notes for 15.564: Information Technology I. Retrieved April 15, 2016, from MIT OCW, http://ocw.mit.edu/courses/sloan-school-of-management/15-564-information-technology-i-spring-2003/lecture-notes/lec6.pdf

5.13 ATTRIBUTION

The content of this unit of the Learning Material - Introduction to Databases (including its images, unless otherwise noted) is a derivative copy of materials from the book Database Design by Adrienne Watt and Nelson Eng licensed under **Creative Commons Attribution 4.0 International License.**

Download this book for free at http://open.bccampus.ca

The following material was written by Grace Mbwete:

1. Introduction
2. Activity4.0
3. Summary