MODULE ONE **LIS 3010:** APPLICATION OF INFORMATION COMMUNICATION TECHNOLOGIES (ICTs) IN INFORMATION MANAGEMENT

COPYRIGHT ISSUES HERE

Acknowledgment

IDE would like to thank **Mr. Tuesday Bwalya** from the Department of Library & Information Science for writing the module.

TABLE OF CONTENTS

Introduction

This module will explain to you the meaning, features and components of any information systems. You will further learn how to create modern databases using MySQL database management systems. Most modern information systems are anchored by MySQL database management systems. In this regard, you will learn about the theories of database designing. Further, you will learn about data modeling using the *Entity-Relationship model*. You will also learn how to transform data models into real database. This module will also show you how to install server software called *Apache* and MySQL database management system that will run on Apache server. After the installation of the software, you will learn how to use SQL commands to create a database and its tables in MySQL database management system; store and retrieve data from the tables. In addition, you will learn how to backup and restore MySQL database.

This Module has **six** units. In unit **One**, the module defines major concepts such as information systems, data, information, knowledge and wisdom/intelligence. Unit one will also explain among other things; components of an information system, types of information systems and approaches to designing information systems.

In unit **Two**, you will learn the theories of database designing which will cover among other things; meaning and uses of databases, history and types of databases.

In Unit **Three**, you will learn how to create data models for databases using Entity-Relationship Model, and how to transform data models into real database designs.

Unit **Four** explains the history of Structured Query Language (SQL), the language used to design modern databases. This language is used by server based database management systems such as MySQL. It will further explain the types of SQL commands and types of data to be used in MySQL databases.

In Unit **Five**, you will learn how to install and test run Apache server and MySQL database management system.

Unit **six** will show you how to use SQL commands (Statements) to create a database, tables of a database, store and retrieve data from the tables. You will further learn how to modify the tables, and database using SQL commands. Finally, you will learn how to backup data from MySQL database and restore it.

Aims

The aims of this module are **five** fold:

Explain the meaning, components and types of information systems. Further, it aims at explaining the history and theories of database design and data modeling using the Entity-Relationship Model. This module also aims at showing you how to install Apache server and MySQL database management system, design a database using SQL commands, backup and restore MySQL database.

Objectives

Upon completion of this module, you should be able to perform the following:

- Explain the meaning of an information system, components, historical development and approaches to designing information systems.
- Explain the theories of database designing
- Explain the history and types of SQL
- Install Apache server and MySQL database management system
- Create a database using Structured Query Language (SQL)
- Query and manipulate the database tables using SQL statements (commands)
- Backup a MySQL database and restore it.

Time frame

You will require **three** months to finish this module

Study Skills

For you to understand the contents of this module, you need to begin with the first unit. Units are interrelated so do not skip one unit or else you will not understand the contents of this module.

Need help?

For any help, please contact the Institute's learner support officer, **Mr. Mwewa** on **0976656105**

Assessment

Continuous assessment for LIS 3010 course will be 60%. You will require to do **three** practical assignments and **one** test during residential school; all carrying equal marks of **15%**. The final examination will carry only **40%**.

Resources needed

For you to do this course, you need a computer with not less than 512 Mb of RAM and 1GHz processor speed. A computer with better specifications will be desirable.

In addition, you will need text books and articles in the areas of information systems and MySQL database management systems.

UNIT 1: Information systems

Introduction

In unit one, you will learn the meaning of the term *information system, data, information, knowledge* and *wisdom/intelligence.* This unit also explains components of an information system, historical development of information systems and types of information systems that are found at various levels of an organization. Further, you will learn various approaches to information systems development and how to manage them.

Aims

This unit seeks to explain and define the major concepts regarding information systems development and components that form an information system, historical development and various types of information systems. It also explains the major approaches used in information systems development and how information systems are managed.

Objectives

At the end of this unit, you should be able to do the following:

- Define the terms such as information systems, data, information, knowledge and wisdom/intelligence
- Explain components of an information system
- Explain the historical development of information systems
- Know various information systems that are found at different levels of an organization
- Know various approaches used in developing information systems
- Explain the historical development of information systems
- Know how to manage information systems

Equipment needed

No equipment is needed to cover this unit.

Other Resources

You need books that discuss information systems. These books include the listed below:

Haag. S, C. M and Philips. A (2007). *Management of Information Systems for the Information Age*. Beijing: China Machine Press

Laudon, K.C and Laudon, J.P (2006). *Management of Information Systems*. 6th Ed. Beijing: Pearson Education

Malaga, R.A (2006). *Information Systems Technology*. Beijing: Publishing House of Electronics Industry

McNurlin, C and Sprague, R.H(2008). *Information Systems Management in Practice*. Xian: Xian Jiaotong University Press

Scholtiz, M (2006). *Application of Executive information systems in South Africa***.** [Online]Available:http://libserv5.tut.ac.za:7780/pls/eres/wpg_docload.download_file?p_filename=F2058583165/mariusscholtz.pdf. [Accessed on 16/11/2013]

Time Frame

You need **One** week to finish studying this unit.

1.1 Introduction

Every organization has an information system that collects, stores data, and analyzes such data to create information which is disseminated for among other purposes; decision making. These systems have stipulated rules and regulations that are followed so that information collected meets the set standards. Some systems are manual (meaning that they are not computerized) while others are automated or computerized. For example, if you go to the University of Zambia Clinic, there is a system to get information from a patient. The system is manual as it requires the registry officer pulling and taking the patient's card to the medical officer who sometimes may request the laboratory to undertake tests for specified ailments by filling in a form which a patient takes to the laboratory technicians. The results from the laboratory technicians are written on the same form and given to a patient to take to the medical officer for him/her to a make a precise decision of the disease a patient is suffering from.

At our International Airport (Kenneth Kaunda Airport), there is an information system which captures information about persons entering the country and it is linked to the Ministry of Home Affairs Information System that has information about Zambian nationals and other nationals as it is in turn linked to other National Information Systems. The system is computerized as you are not required to fill immigration forms as you are about to land. The immigration officer will just ask for your passport and swipe it on the machine or ask you to place your thumb on a machine; all your biodata will be displayed. The immigration officers will check whether or not you are a prohibited immigrant. The officer will enter information additional about you if needed.

Similarly, the University of Zambia Library has a computerized information system to management its collection and users. Once first year students are enrolled in the university; the library collects information about them and stored it in the system. When these students want to borrow the library material, the library staff at the circulation desk will retrieve the information of the library user from the system and make a decision whether or not to lend such a user a book.

In addition, the University of Zambia has a student registration system. This information system collects and stores information about the entire registration process of a student. When a student pays tuition fees, the system collects that information from ZANACO Bank Information System,

and updates the student's records to allow her/him to register for courses. When the student registers for the courses such information is captured and stored in the system.

Banks also do have information systems. These are called Transactional Information Systems which include among other subsystems; Automatic Teller Machines (ATMs) and, Online Electronic Banking Systems. These provide self-service banking services to the banks' customers as a way of improving efficiency and scale up productivity. It is also vital to mention that these systems do capture vital data which banks use in their operations.

What is clear from these examples is that all Information Systems collect and store data/information which is used for decision making and other uses such as tracking down the movement of an entity on which the information is kept e.g. terrorists. It is also clear that in any system, there are some components such as human resources (people), machine hardware (computers, cameras), software and networks

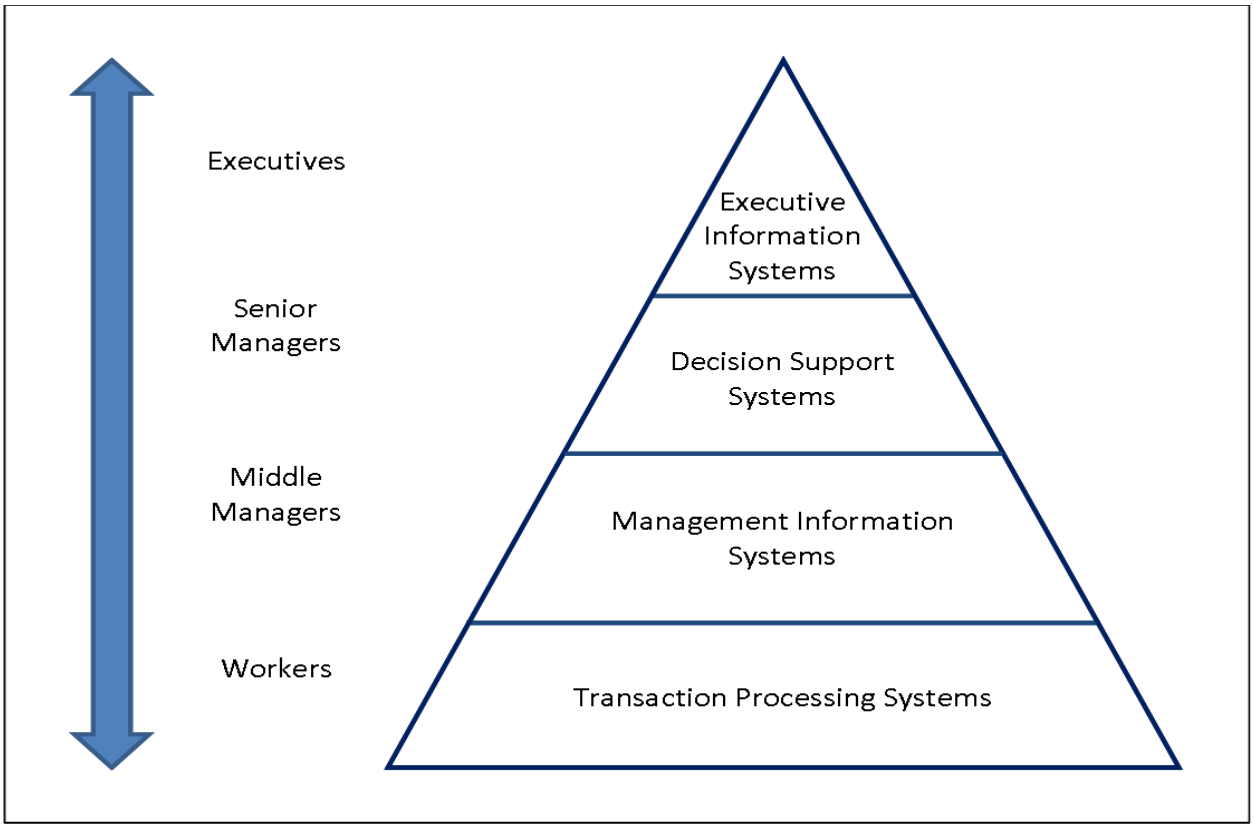
1.2 Definition of concepts

We have been using terms such as information systems, data and information without defining them. At this point, it is imperative to professionally define key concepts.

a) Information system

Many scholars have defined information system in different ways. Kenneth and Jane (2006, pp7) define an information system as a set of interrelated components that collect (or retrieve), process, store, and distribute information to support decision making and control in an organization. In addition to support decision making, coordination, and control, an information system can help workers analyze problems, visualize complex subjects and create new products. According to Ross (2006, pp9), an information system is a set of interrelated information technologies that work together to process, store, retrieve, collect and distribute information. Examples of systems include: Data warehouses, Enterprise resource planning, Search engines, Geographic information systems, Global information systems, office automation and Integrated Library Management systems. Ross (2006) further postulates that information systems are found at various levels and within functional areas of an organization. These levels of the organization are strategic, management, knowledge and operational levels.

Figure 1: *Types of information systems in organizations*



Source: McNurlin, C and Sprague, R.H (2008)

b) Data

It is true to say that some information systems collect and store data. So what is data? According to Ross (2006), data are raw facts. It is data that is not contextualized. These facts could be in form of figures, words or statements in text. Examples of data in form of figures include **40˚c** and **23cm.** Examples of text data include **coughing heavily**, **nice** and **slow to run**. Data could also take the form of sound. Let say the sound produced by a male frog, the **croak** sound and the **Neh (I'm hungry)** sound produced by the baby. These figures, statements, words and various sounds have no meaning in themselves until they are put into context.

c) Information

Information therefore is data that has been contextualized. Data that has been processed (meaning added to it). If someone says today's weather is 40˚c or says the man is slow to run. This becomes information. We can also say the frog is croaking or the baby is crying (neh). This too is information. In these examples, we can see that meaning has been added to data.

i. Value of information

Any information received or collected should possess certain features for it to have value. There are six qualities of information namely;

- *Accuracy*-information should be correct and factual for it to be relied upon. Consider the source of such information.
- *Timeliness*-information should be current not outdated. Consider when such information was released. This is so because information changes rapidly.
- *Accessibility*-information needed should be made available or accessible. It should not take long for one to access it.
- *Engagement*-information should be able to affect a decision/ influence decision making.
- *Application*-information should be relevant to the current context. It should be relevant to the decisions you want to make.
- *Rarity*- information should not be previously known or non-confidential. It should not be in the public domain. For purchase of shares on the stock markets, you can get insiders information although it is illegal to do so.

d. Knowledge

Knowledge is the understanding and awareness that one gets after using or interpreting information. For the example above, if one gets the understanding that today it is going to be very hot by looking at today's weather information, that understanding constitutes knowledge. If one says the baby is hungry because of the **Neh** sound it produces that understanding is knowledge.

e. Intelligence

Intelligence therefore refers to the better judgment that one exercises after gaining the knowledge (Ross 2006). It is the acumen and expertise, insight etc one exhibits in business or in daily life. For example, if one decides to wear summer cloths because he/she has come to know that it would be hot, that is intelligence or wisdom. If one decides to feed the bay after realizing that he/she is hungry, the act of feeding the baby will constitute the act of being wise or intelligent.

1.3 Components of an information system

It is clear that any information system (manual or automated) is made up of many components that work together to collect, store, analyze and distribute information. These components include *Computer hardware, software, networks* (internet and intranets), *people* and *procedures.*

- **Hardware**

This is the physical component of any computerized information system. It consists of computers which include among other things input and out devices such as monitor, keyboard and mouse that we can all see with our naked eyes**.** The input hardware of the computer such as the keyboard convert our natural language characters, special symbols and numbers into machine readable language called **binary**. One of the coding systems used is called ASCII (American Standard Code for Information Interchange). Many personal computers use ASCII to represent, process, and store information. In ASCII, eight bits represent one natural language character and is called a **byte.** For example, the letter **A** will have the following eight bits **01100001.** A bit is a smallest unit of data. Bits are simply strings of 0s and 1s which is the language a machine understands. So the input hardware of a computer plays a role of an interpreter between you and the machine or computer. If for example, you are typing the word **hello**, the keyboard will convert it into binary language and it will be as follows: **01101000 01100101 01101100 01101100 01101111 01110111**

The output hardware components converts back the machine readable language into human language for you and me to understand. Examples of output devices include *monitors* and *printers*.

- **Software**

Computerized information systems cannot work without software. According to Stephen, Maeve and Army (2007), software is a set of instructions that the hardware executes to perform an information-processing task. Software therefore is the program used for collection, storage, processing and dissemination of data. Software is two fold namely; *Application Software* and *Operating System Software*.

a) Application software

Application software as the name suggests is software you use to meet your specific information-processing needs, including word processing, payroll, customer relations management, project management, accounting etc. Application software is broken into two types; namely *Personal*

productivity software and Vertical & Horizontal market software. Productivity personal software helps perform personal tasks such as memos, creating graphs and slides. Examples of personal productivity software include *Microsoft Word, Excel, Power point, FireFox, Internet Explorer etc.* On the other hand, Vertical Market Software is an application that is unique to a particular industry. It is software written and used in a particular industry. Examples of such software include *integrated library management systems* that are used in field of Library Science such **Koha, ABCD** and **OpenBiblio**. Others include Patients Scheduling Systems that are used in the health industry. Horizontal market software as the name suggests is software that is applicable to a variety of industries. In short, it is general in terms of its application. Examples of such software include Payroll systems, Billing software and invoice processing.

b) Operating system software

This software controls your application software and manages how your hardware devices work together. If for example you are using Word Processor to create a word document, and you choose to print it, the operating system software takes over by ensuring that the printer is attached to your computer and it has sufficient papers; if does not have papers, it will tell you. It will finally send instructions to the printer to print and how the document should be printed. Examples of Operating System Software include Windows 7, 8 &10, Linux (Ubuntu) both for Personal Computers and Servers.

Note: All operating systems software need utility software for them to work properly. Utility software adds functionality to the operating system software. Examples of utility software include *Anti-virus software* that protects your information system from viruses and drivers such as sound drivers (to produce sound on your PC), network drivers (for allowing your PC connect to the Internet or a Local area network) and printer drivers (allowing your computer and printer to communicate).

Network

According to Oxford Advanced Learner's Dictionary (2010), a network is a closely connected group of people, companies or computers etc that exchange information. In the case of manual information systems, a network could mean offices, units, departments and people in an organization that closely work together to exchange information. For computerized information system, a network could mean two or more computers that are linked in order to share resources (such as printers and CDs), exchange files, or allow electronic communications. The computers

on a network may be linked through cables, telephone lines, radio waves, satellites, infrared or light beams. Examples of networks for computer based information systems are the Internet and Intranets. This shows that a network is a component that is feasible or present in all information systems. A network is needed for the exchange of information. Information moves from one point to another through a network. This could be a computer network or offices/ people.

- **People**

The human resource part is critical to any information system. These are foot soldiers of any information system. In the computerized information system, people operate / manage the computer (both software and hardware). They too administer all computer networks including the Internet and Intranets. They capture data onto the system and distribute to others in and outside the organization through computer networks. For a manual information system, people perform more tasks that include capturing, storing, analyzing data and distribute to others in the organization through a network of offices and units.

- **Procedures**

Procedures are ways of performing or affecting something or a series of steps taken to accomplish an end. Procedures could also mean a set of established forms or methods for conducting the affairs of an organized body such as a business, club, or government. Procedures need to be established and handed down to people working with the information system on how they should capture data, store, analyze and distribute it. There is also need to set parameters or privileges in terms access to data and information stored in systems through the use usernames and passwords for a computerized information system. For the manual information system, rules are set out on who should access what information.

1.4 History of information systems

Information systems are as old as civilization. When people knew how to write and read, they devised ways of collecting, storing and distributing information. For example, people used clay tablets to collect and store information about any transactions. It is however salient to mention that automated or computerized information systems are a new development. They are just over five decades old. The emergency of automated information systems has been influenced by three factors; namely the birth of the global economy, the transformation of industrial economies and

societies into knowledge and information-based service economies in the early 1960s and the change in the business environment and climate (Samppa (1999).

i. Transformation of industrial economies into knowledge economies

After the Second World War, many economies thrived on having sophisticated machines. Competition was based on machine. Economies like the USA, Japan etc that invested heavily in machinery became industrialized. This however was short lived as many economies became industrialized. With the birth of Computers in the 1960s, many economies looked to ICTs to gain competitive advantage. As observed by Nisbitt in Samppa (1999), the USA made this transition from and industrial economy to knowledge or information society where majority of its people were and are still employed in the handling of information. Economies began to use information/knowledge as a source of competitive advantage. ICTs were adopted to collect, store, process and distribute data and information.

ii. Birth of global economies

The emergency of the Internet and World Wide Web (W3) in the early 1990s has created a global village; a world where societies are linked through the Internet and W3. Physical barriers such as distance that inhibited many companies and people from communicating faster have been eliminated by the Internet. For instance, many firms have now diversified geographically by investing in foreign counties. They have branches all over the globe. To ensure coordination and control of branches that are dotted around the globe, corporations have developed computer based information systems to help them manage these firms.

iii. Change in the business environment and climate.

Related to the second reason for the evolution of information systems, is change of business environment. In the past, organizations used to adopt a large bureaucratic organizational structure or centralized organizational structure; many hierarchies which made organizations inefficient. Today's organizations adopt a flat or decentralized organizational structure; offices are spaced geographically and employees are empowered to make decisions. This change has necessitated the use information systems so that both the workers and supervisors could easily access the much needed information as they perform their functions.

1.4 Types of Information Systems

There are basically **six** types of information systems that exist at various levels of an organization. These are:

i. Transactional information system

This system is found at the bottom of an organizational hierarchy. Transactional process system is computerized/non- computerized system that performs and records the daily transactions performed in an organization. It supports operational employees to perform day to day functions of the organization. Employees that make use of the transactional system include cashiers, accountants, sales officers, receptionists, etc. An example of transactional system includes Point of Sales systems, Payroll systems, Hotel reservation systems, Library circulation system, Bank transactional systems and University and College registration systems. Transactional systems collect and process internal data.

ii. Office automation system

This system support workers such as clerks, secretaries, bookkeepers etc. These workers are sometimes referred to as **data workers**. These employees create, use and manipulate data in their work but they do not typically create new information. Example of office automation systems include word processing, desktop publishing software such Adobe Photo shop, Document imaging systems, communication software and spread sheet software programs.

iii. Knowledge work systems

These are highly specialized systems that are used by specialists. These workers include medical doctors, Lawyers, Professors and Engineers. Examples of knowledge systems include anti-plagiarism software used by professors such as *Viper*. *AutoCad* and *Design analysis* software used by engineers are also examples of knowledge work systems. Lawyers also do have their specialized systems such as *Practice* and *Case management systems* which are deemed to be knowledge work systems.

iv. Management information system

As the term suggests, these are systems that provide information to middle level management to generate reports, warn them when problems are about to happen in a particular area of their operations (such as Operations, finance and human resources). These systems are used by heads of departments/units in organizations. These systems process internal information coming mostly from the transactional processing systems. These systems can generate reports about sales and

indicate whether the division is meeting the agreed quota for the month or year. Management information system can allow the managers to intervene early to collect problems. Examples of such systems include *marketing information systems* and *finance management systems*.

v. Decision support system

Unlike the management information system that allows managers only to make decisions on routine activities, decision support systems have capabilities that enable managers to make none-routine decisions; for example, *choosing the amount of product to manufacture*. These systems have analytical tools such as sensitivity analysis and simulation modeling that is combined with both internal and external data. For instance, data on sales and employees salaries are internal data while data about economic indicators or competitors' products constitute external data. These systems for example could analyze the future wage bill of the company based on the current salaries and economic indicators such as inflation in the country. Examples of Decision support systems include **SearchBusinessAnalytics** and **Boundless** (global marketing decision making system).

vi. Executive information systems

These support top managers; the executive to make strategic decisions. According to Marius (2006), executive information systems supply corporate information such as financial conditions, market share, and organizational performance in a graphic form. It provides strategic information to top managers (decision makers). A good executive information system should gather both internal (financial standing of the organization, labor wage bills, and human resources skills etc) and external information sources (economy, politics, and technology, legal as well as competitors' moves). An executive information system taps into other organizational information systems for information that is summarized and presented to top managers for decisions making. Example of an executive information system includes **Visual Mining.**

Executive information systems should incorporate *Michael Porters Five Forces Model* if they have to provide the much needed competitive intelligence to decision makers. However, research conducted by Marius (2006) in South Africa shows that many executive information systems do not collect external information therefore, unable to provide the much needed competitive intelligence.

1.5 Designing information systems

Designing an information system is anchored by many approaches. These include the Traditional System Life cycle, Prototyping, Application Software Packages and End- User development approaches.

a) Traditional system life cycle

This is the oldest method of developing an information system. According to Kenneth and Jane (2006), this methodology assumes that an information system has a life cycle similar to a living organism, with a beginning, middle and an end. The life cycle for an information system has six stages: *project definition, systems study, design, programming, installation and post-implementation***.**

In project definition, an organization seeks to ask and find answer to questions such as: Why do we need an information system? What do we want to accomplish with this information system? Is the current information system fulfilling its purpose? In short, the first stage seeks to establish a problem and whether the problem identified can be resolved by building an information system or modify the current ones. The project definition stage spells out the objectives, specifies the scope of the project and develops a project plan that can be presented to management.

The systems study stage analyzes the problems of existing information system in details and identifies objectives to be achieved by a solution to the identified problems. It also examines the feasibility of all the possible solutions. This stage require extensive information gathering through interviews with staff in the organizations, examining existing systems in terms of how they work and review the documents relating to the systems under study.

The third stage which is the design stage involves producing a logical and physical design specification for the solution. The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system while the physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is authenticated, how it is processed, and how it is displayed. At this stage, design and documentation software are used to draw up the new design such as Data flow diagrams and program structure.

Note: You will learn more on this in due course.

The programming stage translates the design into a program code. The programmers and the system analysts will work together to ensure they prepare specific programs for the system. The program codes will be written in third generation programming languages such as COBOL or FORTRAN. The codes can also be written in fourth generation of programming languages such as Focus, Fusion and Mathslab.

The fifth stage, installation entails putting the new system or modified system into practice. The software is tested and used for technical and business point of view.

The last stage which is post-implementation stage involves evaluating the system installed to determine if it has achieved the set objectives. The users of the system need to be interviewed on how the system is working. Areas that need modification are identified and attended to. The implemented system requires continues maintenance if it has to remain efficient.

The Life cycle approach is being used actively by large organizations for technical and complex information systems such as air traffic control system, space launch and refinery systems. The main disadvantages of this approach are that is costly and time consuming. It is also not flexible and discourages change. Therefore, it is not suited for decision-oriented application that requires experimenting with concrete system to clarify the kind of decision they wish to make.

b) Prototyping

This method involves building an experimental information system rapidly and inexpensively for the end users to evaluate. According to Barbra and Ralpha (2008), using a prototype, a system can be built within hours or days not years. The users interact with the prototype and better define their information needs. The prototype if endorsed by users is used as template to create the final system. The prototyping method has four steps; which are as follows:

- Identifying the user's basic requirement- the system specialist works with the users long enough to capture their needs.
- Develop an initial prototype- the system designer quickly develops a working prototype using fourth generation of programming languages.

- Use of the prototype- users are encouraged to work with the prototype system in order to determine how well it meets their needs.
- Revise and enhance the prototype- The system builder notes all the changes requested by the users and refined the prototype accordingly. After is revised, the cycle returns to stages 3 and 4.

This is useful method to use in situation where there is uncertainty about the requirement of a system. This method also enables end-users to immediately react to the system. It is therefore recommended when you are developing end-user interface of information systems such as online display, data entry screens, reports or web pages.

c) Application software packages

Another alternative method to develop an information system is to buy an application software package or get free and open source software. For example, if you want to build a payroll information system, you can just buy an already written payroll system. And if you want a library management system, you can buy from a range of the available software or download free and open source software such as OpenBiblio, ABCD and Koha.

This method is fast and safe as many of the commercially available software have been tested before marketed. Further, the vendors of such software provide maintenance and support services to your system. Example of the University of Zambia library management system called *Unicorn.*

The only disadvantages are that the software purchased might not meet all your requirements. Therefore, there is need to customize which will in turn result in more expenses. In addition, your organization will continue paying subscription fees to the vendor as long as you continue to use the software. This last disadvantage does not arise if you are using open source software. You may just pay for installation and training as most of the Open sources software require technical skills to be able to install and use them.

d) End- User development

In many organizations today, users are developing information systems with little or no formal assistance from technical experts. This is called end- user development. With the coming of fourth generation programming languages such as Fusion, SQL, HTML and CSS and graphical languages and PC software, end users can access data, create reports and develop an entire information system from scratch.

This method is the quickest way to have a system. Unfortunately, users can only develop small and limited systems. Robust systems need to be developed by specialists and should be written in the third generation languages that are robust enough. Further, this method posses challenges to organizations on how control and coordinate end- user system developers. Further, organizations need to provide some training to its workers if they have to be in position to develop serious systems.

1.6 Managing information systems

Once a system is built and functioning, an organization needs to ensure that it is protected and secured to ensure data integrity. The system should be protected from among other things:

- Unauthorized accesses to the information systems- Hackers can gain access to your system, deny you a service (unable to use the system for your work) or steal your data and information. This happens when hackers discover a fault in the way the program was written. To prevent this, you need to protect your information systems by installing firewalls on the servers or computers where they systems are installed. For physical access to the system by unauthorized users, the organization should ensure that every employee accessing and using a system has a *username* and *password*. **Note** that passwords should have both letters and numbers making them difficult to be cracked.
- Protect your computers/Servers where your information system is installed against Trojan horse and Viruses by installing an anti-viruses software
- Check your information system (software) for bugs and attend to them immediately. Bugs are portions of the program that were not written properly or have errors. These errors in a program may occur during compilation (translating a program from the human language to computer language; the codes.

- Look out for batches if you have built an information system using purchased or open source software. Batches are small programs created to resolve the weakness identified in the main software.

Summary

From this unit, we have learnt that information systems either manual or automated do exist in organizations. Their role is to collect, store, process and disseminate information/data for decision making and other uses in organizations. It has also emerged that any information system has components; these are Computers or Machines (hardware and software), people, network, and procedures/rules. Further, you have learnt that the evolution of information systems was influenced by the transformation of industrial economies into knowledge economies (competition among economies and companies was based on human knowledge not machine), the birth of global trade facilitated by the coming of the Internet in the early 1990s and organizations' adoption of flat organizational structures that decentralize decision making. We also learnt that there are basically **six** types of information systems found in organizations. These are *operational information systems* (that provide information to operational staff), *office automation systems* (provide information to data workers or help them process data), *knowledge or expert information systems* (provide information to knowledge workers), *management information systems* (provide information to managers/heads of units), *decision support systems* (assist managers/executive officers to make decisions) and *executive information systems* (provide information exclusively to executive officers). We have further learnt that there are **four** approaches to information systems design; namely *traditional system life cycle* (with six stages: project definition, systems study, design, programming, installation and post-implementation), *prototyping (*involves building an experimental information system rapidly and inexpensively for the end users to evaluate), *use application software packages(* buy an application software package or get free ware and use it to develop an information system) and e*nd- user development (*users develop an information system with little or no formal assistance from technical experts). Last but not the least; we have learnt that information systems need to be managed if they have to achieve their intended purposes. This implies that there is

need to regulate access to the information systems through the use of usernames and passwords. There is need also to protect the network and computers where the information system is installed from hackers , viruses and worms.

Activity

1. Get your computer connected to the Internet and search for any binary converter application; use it to convert your names from human language to binary language.
2. Visit any organization and study the information system it uses.

UNIT 2: Introduction to database systems

Introduction

Unit two ushers you into the world of database designing. It begins by defining the term database, explaining the historical background of database processing, characteristics and types of databases. Further, this unit explains the benefits of using databases in information management.

Aims

This unit endeavors to explain the concept of a database, historical background, characteristics and types of databases. It also aims at explaining the benefits of using databases in information management.

Objectives

At the end of this unit, you should be able to do the following:

- Describe a database
- Explain the history of database systems
- Explain the components of a database
- Explain types of databases
- Explain the benefits of database approach to information management

Equipment needed

For this unit, you may need a computer to be able to draw data models and transform data models into real database designs. Plain papers and pencils could could be used in case you do not know how to use drawing software to create data models.

Other Resources

You need books that discuss database systems. These books include the listed below:

Beighley. L and Morrison. M (2009). *Head First PHP &MySQL*. Cambridge: O'Reilly.

Chao.L (2006). *Database Development and Management*. Boca Raton: Taylor &Francis Group.

Hoffer, J.A, Ramesh. A and Topi. H (2011). *Modern Database Management*. New Delhi; Chennai: Pearson; Dorling Kindersley

Kroenke, D.M (2005). *Database Processing: Fundamentals, Design, and Implementation*. 10th Ed. Beijing: Publishing House of Electronics Industry

Kifer. M, Bernstein.A and Lews, M.P (2005). *Database Systems: An Application-Orientation Approach*. 2rd Ed. New York: Pearson Education

Narang, R (2009). *Database Management Systems.* New Delhi: PHI Learning

Stephens. R, Plew. R and Jones, Arie. D (2012). *Sams Teach Yourself SQL*.4th Ed. New Delhi: Pearson; Dorling Kindersley

Time Frame

You need **One** week to finish studying this unit.

2.1 ***Introduction***

Databases are created by all of us in life. We often write down (either on paper, tablet or on a computer) a list of people who for example have borrowed books from us. That simple act of listing names of borrowers is synonymous with database creation. Organizations do create databases to keep records of among other things employees. Details that are recorded about employees include their *full names, year they started work, qualifications* and *contacts*. This information is kept to ensure easy access to employees' records. A database of any type (computer or hard-copy) is a cardinal unit of an information system. Many scholars of information systems treat a database as one of the major components of an information system. However, in this module, we deliberately fused it with the software component of a computer when discussing components of an information system. Those who treat a database as a component on its own might be correct because it is an engine of all information systems. Any information system needs a database to be able to capture, process, store and disseminate data /information.

2.2 ***Characteristics of databases***

All databases regardless of size and purpose have the following below features:

i. **Databases store data and information in tables**

Data and information about things (instances) is stored in tables which are made of columns and rows. The columns describe the data/information about an object (thing) and such data is stored in rows. This implies that data in columns is metadata because it is information about rows. If you have created a simple database in Microsoft Access before, you will see that indeed data is stored in a table as shown below.

Table 1: ***Microsoft Access table***

Comp #	Test1	Test2	Test3	Total	CA (%)	GRADE
12225667	11	15.5	10	37	**73**	**B+**
10099182	12	14.5	10	37	**73**	**B+**
29106338	11	14.5	10	36	**71**	**B+**
11067845	10	16	10	36	**71**	**B+**
10012150	9	16	10	35	**69**	**B+**
11119101	12	12	10	34	**68**	**B+**
11078294	12	12	9.5	34	**67**	**B+**
11068604	12	12	9.5	34	**67**	**B+**
11068361	8	15.4	10	33	**67**	**B+**

Note: Tables are a building block of any database. A database can have more than one table. But it is also possible to create a database with one table.

ii. **A database has data and relationships**

Another feature of all databases is that data in one table is related to other tables in a database. The relationships could be on the basis of *primary key*. For the two tables below, table 1 is related to table 2 on the basis of a **computer number**; the computer number is the link between the two tables.

Table 2: ***Microsoft Access table***

Comp #	Test1	Test2	Test3	Total	CA (%)	GRADE
12225667	11	15.5	10	37	73	B+
10099182	12	14.5	10	37	73	B+
29106338	11	14.5	10	36	71	B+
11067845	10	16	10	36	71	B+
10012150	9	16	10	35	69	B+
11119101	12	12	10	34	68	B+
11078294	12	12	9.5	34	67	B+
11068604	12	12	9.5	34	67	B+
11068361	8	15.4	10	33	67	B+

Table 3: ***Microsoft Access table***

Comp #	Yearof Study	Course
12225667	2	LIS 2010
10099182	3	LIS 3210
29106338	4	LIS 4135
11067845	3	LIS 3213
10012150	1	EDU 1020
11119101	2	LIS 2231
11078294	3	LIS 3010
11068604	3	LIS 3010
11068361	4	LIS 4031

As you can see above, the relationship among the two tables is based on student computer number which acts as a primary key. Rows of data are related and therefore their relationship makes it possible to extract information from them easily. For instance, you can get information that: *student (12225667), a LIS 2010 student doing her/his second year of study got 73% of CA.*

iii. Databases create information

From the data that all databases keep, information can be extracted and presented in a meaningful context. From the two tables above, one can extract data from the tables using Structural Query Language (SQL) and contextualize it as demonstrated above by saying that *student (12225667), a LIS 2010 student doing her/his second year of study got 73% of CA.*

2.3 ***Definition of concepts***

The term database has many meanings. David (2005, p.11) defines a database as a self-describing collection of integrated tables. Integrated tables store both data and relationships among the data. For example, tables that store data about employees' names, qualifications and contacts will equally store data about relationships among the rows of data and tables. A database is self-describing because it contains a description of itself. This is because database

contains not only tables of user data but also tables of data that describe the user data. This descriptive data is called *metadata* because it is data about data. It is data that contextualize data in table rows.

2.3 ***History of database systems***

The evolution of database systems can be broken into **five** periods. These are as follows:

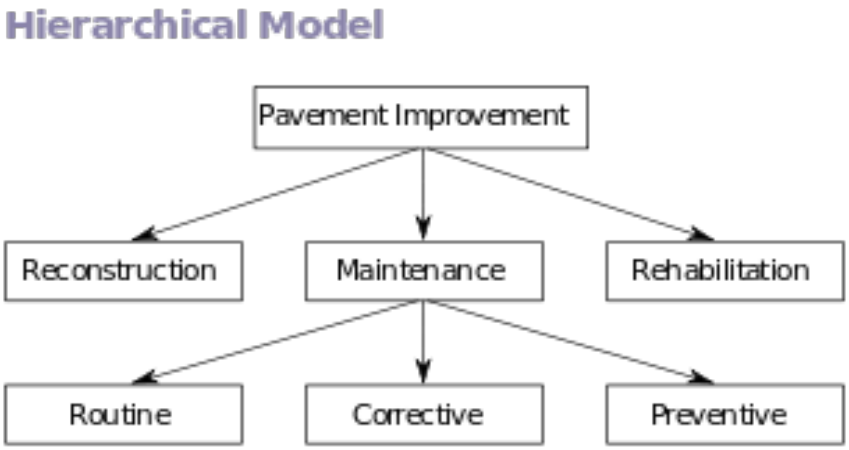
(a) 1960s

During the above period, data was stored in separate files, most of which were kept on reels of magnetic tape. These had limitations in terms of capacity and cost. Further, the storage of data on separate files created a problem of data integration for organizations that wished to use different data files simultaneously as this meant that they have to retrieve information separately from the files. For example, if the University wants to relate student's results to his or her contact details, two separate files; one on student's results and the other on contacts need to be opened. Jeffrey, Ramesh and Heikki (2011), argue that towards the 1960s, an experimental "proof-of-concept" of database management system was developed by Apollo moon-landing project in the United States of America (USA). The database created helped to store massive of data needed for the launch of the rocket into space by NASA. Further, during the first committee called Data Base Task Group in the USA was formed to try and create some standards regarding database management systems.

(b) 1970s

By this period, the need for integrated databases heightened. Therefore, many organizations were compelled to develop database management systems. According to David (2005, p.19), by 1973, several commercial database management systems (DBMS) had emerged. These database systems include ADABAS, IDMS, System2000, Total and IMS. Today only two (ADABAS and IMS) of these database systems are still in use but with a small market share. The early DBMS products varied in the way they structured data relationships. Most of these early DBMS used the method called Data Language/I or DL/I. This model represented data relationships in a hierarchical or family tree structure. IMS, a database management system developed by IBM was based on this model. This structure represents information using parent/child relationships; each parent can have many children, but each child has only one parent (also known as a *1-to-many relationship).*

Figure 2: *Hierarchical model*

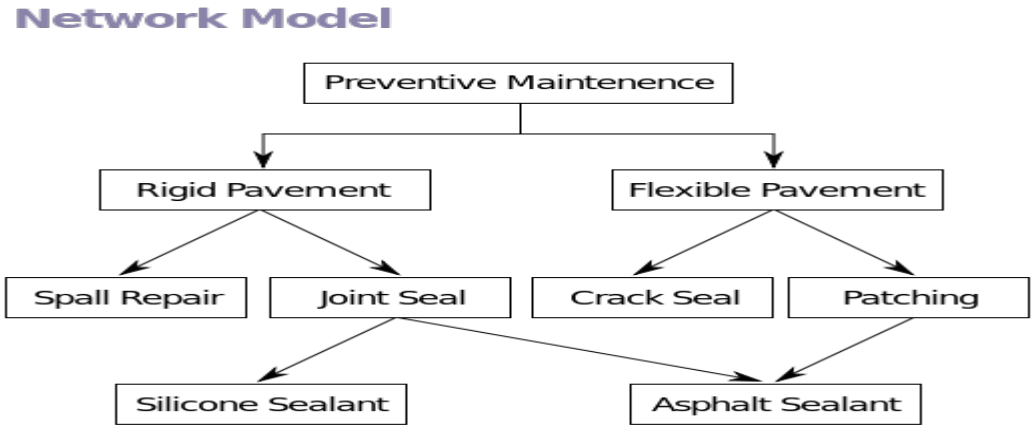


Source: Kroenke, D.M (2005)

Note that most companies developed databases based on hierarchical model because by then data was stored on serial devices such as magnetic tapes; for which data was only accessible in sequence (one after the other).

Other database management systems during this period were based on a network model. The network model allows each record to have multiple parent and child records, forming a generalized graph structure.

Figure 3: *Network model*



Source: Kroenke, D.M (2005).

These two models are considered to be the first generation of database management systems and still being used by organizations. Database management systems based on hierarchical or network model like the file processing system have disadvantages. These include limited data dependence (*data descriptions are included with the application programs that use the data*) and lengthy of development times for application development.

(c) 1980s

As a solution to flawed database management systems that existed in 1970s, E. F. Codd, a Research Fellow at IBM Research Laboratory produced a paper on relational data model. The paper addressed the weaknesses of the earlier existed database management systems. Many experiments were performed and Codd's idea of data model was perfected and widely implemented in 1980s. In the relational model, data is represented in the form of tables. Typically, Structural Query Language (SQL) is used to retrieve it. This model provides easy access to data by none-programmers as it has among other features graphical user interface (*a visual way of interacting with a computer using items such as windows, icons, and menus, used by most modern database management systems*). This stage is considered to be the second generation of database management systems.

Note: We shall discuss SQL in details later.

(d) 1990s

You will agree with me that the Internet was born in the 1990s. It is this period that client/server computing, data warehousing and Internet applications emerged**.** The most remarkable development in database management system is the emergency of database systems that could keep not only structured data but also unstructured data such as sounds, images, video and graphs. Database management systems that existed prior the 1990s kept only structured data such as accounting data. During this stage, object-oriented databases were introduced. These are considered to the third generation of database management systems. Object oriented database management systems (OODBMSs) combine database capabilities with object-oriented programming languages (Java, C++, Perl etc) capabilities. OODBMSs allow object-oriented programmers to develop the product, store them as objects, and replicate or modify existing objects to make new objects within the OODBMS. Some Server based database applications such e-commerce databases use Object-oriented data model.

(e) 2000 and beyond

Currently, the most used database type is the relational databases. However, object-oriented and object-relational are increasingly being used especially with the emergency of web 2.0 applications such as Facebook, Twitter, Linkedin, Blogs, and wikis.

The main development in the field of database technology is the birth of object-relational database management system (ORDBMS). It is a hybrid model born from the relational and object oriented models. This model has borrowed features from the above mentioned database models. The idea is to keep multimedia data in tables. Object relational database management systems provide a middle ground between relational and object-oriented databases. In an ORDBMS, data is manipulated using queries in a query language. Examples of Database systems based on this model include Oracle and Microsoft SQL Server.

2.5 ***Advantages of database approach to information management***

There are a lot of benefits that accrue to an organization or individual that manages information through the use of a database. These include the follow:

a. Control of data redundancy

As we learnt earlier, in the traditional file-based systems, information about an object (object/instance) was stored in separate files. For example, information about property for rent and clients could be stored separate in the sales and contract departments (Thomas and Carolyn, 2013, p.26). This results in duplications as the same information is kept in many places making it difficult to use it. However, the Database approach eliminates this flaw by keeping all related information together in tables and integrate it through the use of primary key so that it becomes easy to access and use related information. This reduces redundancy of data as multiple copies of information are not stored.

b. Data consistency

Since database seeks to ensure that data/information about an object is stored once, it therefore reduces the possibility of retrieving incorrect information. This is because, if data in a table is updated, the new information will be immediately available to users. But if so many files of the same data are kept, you may forget to update some files, therefore; resulting in having different information about the same thing. Imagine you have been typing an assignment and decides to save it in two folders; when you want to continue typing the same assignment, you will open one

file and continue typing. When saving the latest version of your assignment, chances are high that you may forget to update the other file on which you were not working on. If care is not taken, you may carry the unfinished assignment for printing. This however is eliminated with database as information about the same object is kept in related tables and any changes will reflect immediately in other related tables.

c. Easy access to information

Since data is stored in a database (server based database) in an organization, an employee with access rights can access it at any time. They can do that at their offices by just login to the database system. In addition, many users can access the information at the same time (simultaneously access to information). Unlike in the traditional file system where only one user can have access to the file at a given time.

d. Improved data integrity

Database enforces data integrity in the sense that restrictions and rules are applied when dealing with stored information. For example, the database system will not allow any one apart from the database administrator to change information stored in the tables. This protects data and makes it trust worthy.

e. Improved security

Related to data integrity, data/ information stored in the database is secure as only authorized users can access the information. This does not mean the system cannot be hacked by hackers. *(Hackers are computer sage who can access your data by cracking the logging details)*

f. Economy of scale

Storing related information in one place (database) reduces the cost in terms of space and human effort to look after scattered information. You need more people and other resources to manage scattered files in various departments of your organizations. Unlike in the database environment where you just need a skeleton of staff and take care of your information. This does not mean keeping information in database is cheap. The initial capital is high as you need to spend money on things such as computers hardware (including servers), software and training of people in database skills if they are lacking skills.

g. Increased productivity

Since database helps to organization and keep information in an efficient manner, it can therefore be said that people in the organization can fast and easily access the information they need in the

execution of the mandate/ mission of your organization. Therefore, they can work fast and produce more results.

2.6 ***Components of a database***

Like an information system, a database (a sub-component of an information system) has components; parts that make it up. There are five components; namely *hardware, software, data, people* and *procedures*.

a) Hardware

As agreed, computer hardware is the physical components of the computer that you and I can see. All database Management systems (DBMS) require computer hardware for them to work. They run on hardware. This hardware could be a personal computer or server computer. Note that some DBMS require a minimum amount of memory from your hardware if you have to install them.

b) Computer Software

According to Thomas and Carolyn (2013), the software component comprises the DBMS software itself and the application programs, together with the operating systems, including network software if the DBMS is being used over a network. Databases applications are normally written in third –generation languages such as C, C++, Java, Visual Basic etc. They are also written in fourth-generation languages such as Structural Query Language (SQL), embedded in most of the third-generation programming languages. Some of the network software that is used in databases includes Server software such as **Apache.** We shall use this server in our module as we design server based databases.

c) Data

This is probably one of the most important components of a database from the user's point of view. Data in any database acts as a bridge between machines and people components. This data is directly accessed or through some application programs. There are basically three types of data found in databases. These are:

i) User Data - It consists of a table(s) of data called **relation**(s) where Column(s) are called fields of attributes and rows are called records for tables.

ii) Metadata **-** A description of the structure of the database is known as Metadata. It basically means "data about data". System tables store the Metadata which includes; number of tables and table names, number of fields and field names and primary key fields.

iii) Application metadata - It stores the structure and format of Queries, reports and other applications components.

d) Users

There are various types of users that can access the database. These are database administrators (DBA), database designers and end users. Database administrators are in charge of the day to day managing of the database. In addition, he/she is in charge of setting up of access privileges to the data by users and updating the system. The database designer is the person who produces the data model. He/she plans how the tables in the database will relate to each. In short, he/she is the architecture of a database. In many organizations, the database administrator performs even the role of the database designer. End users are those persons who need the information from the database in order to carry out the functions of an organization. These could include; personnel, staff, clerical, managers, executives etc. On the basis of the job and requirements made by them they are provided access to the database totally or partially.

e) Procedures

As we learnt earlier, users of any system requires documented rules and procedures on how to work or interact with the system. Similarly, in the database environment, procedures are needed to spell how to run and use the database. For example, users rules on how to login and logout should be given out to the users. Further, rules how to backup data from the database need to be explained.

2.7 ***Types of databases***

Databases take many forms. They are really classified by their usage and number of people or organizations that use them. Below are the categories of databases:

a) Single-user database applications

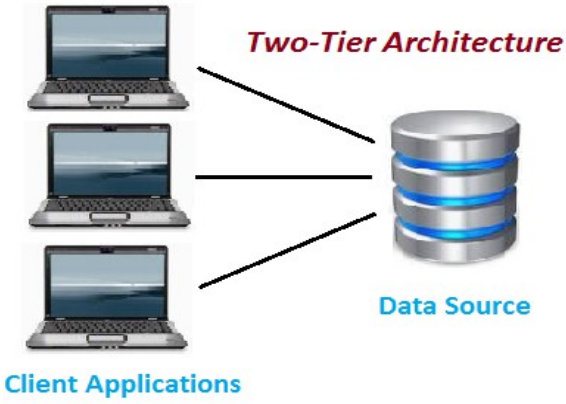
As the name suggests, these are databases that are used by a single user. Only one person can use this kind of a database at a time. These types of databases are created on personal computers such as Laptops and mainly are for personal use. They are created using simple database management programs such as Filemaker (old version), Easytable, Personal library manager and Excel. For example if you create a record in Excel on computer of all people who have borrowed your books, that database qualifies to be called a single –user database application. A database in a shop that list all the products and prices is also an example of a single-user or

personal database. These types of databases are common in small businesses and help in improving productivity as the needed information could be quickly retrieved and used in decision making. The main drawback of this type of databases is that data or information cannot be shared because the database application is installed on a personal computer not on a server.

b) Two-tier/Server databases

They are a direct opposite of a single-user database. These databases allow many users to access and share information. Two-tier or groupware databases are installed and created on servers therefore by allowing many users to access data.

Figure 4: *Two-tier architecture*



Source: Narang, R (2009)

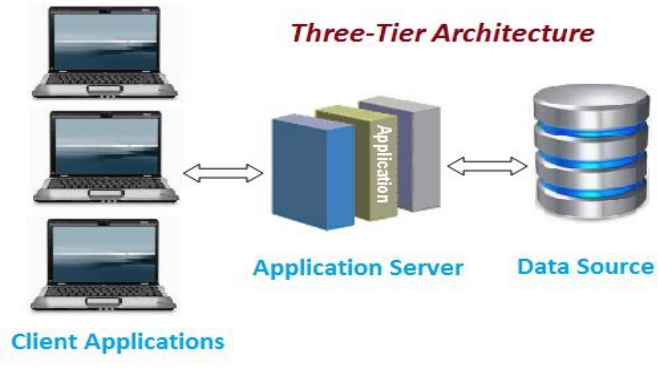
As you can see in the diagram above, in the two tier database, the client computers retrieve data/information directly from the server. There is direct communication between the server and clients computers. The database and server are incorporated (integrated). **Note** that client computers should have an application to communicate with the server and if the servers' software is updated, clients' applications too need to be updated. These databases are common in the transport sector (especially the reservation units). The major drawback of these types of databases are that the client computer has to be powerful enough to handle programmed application and the clients computers always have to update the application software that enable them to access the server. Further, the server cannot respond to multiple requests from the clients at the same time. The University of Zambia Students Examination Results Database System is an example of the two tier principle as computers of staff involved in exam results entry have to install the application called **ewan** on their personal computers (client) for them to access the

server and database. They cannot access the server and database using browsers such as Mozilla Fire Fox or Internet Explorer.

c) Multi-Clients/Server databases

These are modern databases. They have emerged to resolve the weaknesses of the two-tier databases.

Figure 5: *Three-tier architecture*



Source: Narang, R (2009)

As you can see in the diagram above, the three-tier database comprises a *presentation tier*, *a data access tier,* and *a data tier*.

a) *Client layer*- This layer is called the *presentation layer* which contains the user interface (UI) part of the application. The user interface is accessed from the client computers using a browser or any written application. This layer presents data to the user or takes queries from the users to the server application.

b) *Business layer*-This is a server application which talks to the database on behalf of the client. The database understands this application not the client application. It gets queries or requests from the client and translates them and forwards the translation to the database so that the requested information can be retrieved.

c) *Data layer*- This is the server database. The Data Access Layer contains methods to connect with database and to perform insert, update, delete, get data from database based on our input data.

This type of database has many advantages which include the following:

- High performance, lightweight persistent objects
- Scalability – Each tier can be scaled horizontally
- Performance – Because the presentation tier can cache requests, network utilization is minimized, and the load is reduced on the application and data tiers.
- High degree of flexibility in deployment platform and configuration
- Improved data integrity
- Improved security – Client is not direct access to database.
- Easy to maintain and modify; won't affect other modules
- In three tier architecture application performance is good. **Note**: In this course, we shall use the Multi-tier Client/Server database

Summary

In this chapter, we have learnt that databases are used to keep information on an entity or object. Entity could be an employee, student, book and house. Data in the databases are kept in tables and these tables do not only keep data but also data about relationships among data. This is the reason why some scholars define a database as *self-describing collection of integrated tables*. Data about data in the tables and their relationship to others tables is called **metadata.** We have also learnt that database systems have evolved since the 1960s. In the 1960s, data was kept on separate magnetic tapes and this made it impossible to integrate data. In the 1970s, however, there was propensity by many organizations to develop integrated database systems. During this period, several commercial databases systems were developed which include IDMs and IMS. In the 1980s, IBM developed a database management system which was based on Structured *Query Language* (SQL). In the 1990s, database technology entered a new phase with the introduction of features that enable databases not only to keep structured data but also none-structured data such as images and documents. In the 2000s and beyond, database management systems developers have continued to perfect the existing systems so that all manner of multimedia could be stored in tables alongside structured data.

The idea of using integrated database management systems has brought about accuracy, efficiency and effectiveness in data management as issues such as data redundancy and inconsistencies have been solved.

In this unit, we have also learnt that a database could *either be single- user, two-tier or multi-clients database.* Single user means that data processing is done on a single machine and one user can use a database at a time; two-tier database implies that two machines are involved in the processing data; one is the *client* and the other *machine (server)* hosts a database where the client gets data from. The client connects direct to the server to retrieve data. The multi-user or tier systems have three layers namely; *client application, server application and data source layers*. The client does not connect direct to the database (data source) to get data but connect to the server application which in turn connects to the data source to get data and pass it over to the client. This architecture has added more versatility to database systems as many users can query the system at the same time.

Activity

1. Study the University of Zambia Student Registration System to determine what type of a database it is

UNIT Three: Database design approaches

Introduction

Unit three ushers you into the world of database designing. It begins by explaining the meaning of database design and other terms used in database design. Unit three further explains database design steps and data modeling with the Entity-Relationship Model.

Aims

This unit endeavors to explain the concept of a database design. It also aims at explaining steps to database design and the data modeling method called the Entity-Relationship Model.

Objectives

At the end of this unit, you should be able to do the following:

- Explain the meaning of database design
- Use the database deigning steps to design a database
- Create data models through Entity-Relationship Model
- Identify Primary and Foreign keys
- Transform data models into real database designs

Equipment needed

You need a computer to be able to draw data models and transform data models into real database. Alternatively, you may use plain papers and pencils to draw data models.

Other Resources

You need books that discuss database management systems. These books include the listed below:

Beighley. L and Morrison. M (2009). *Head First PHP &MySQL*. Cambridge: O'Reilly.

Chao.L (2006). *Database Development and Management*. Boca Raton: Taylor &Francis Group.

Hoffer, J.A, Ramesh. A and Topi. H (2011). *Modern Database Management*. New Delhi; Chennai: Pearson; Dorling Kindersley

Kroenke, D.M (2005). *Database Processing: Fundamentals, Design, and Implementation*. 10th Ed. Beijing: Publishing House of Electronics Industry

Kifer. M, Bernstein.A and Lews, M.P (2005). *Database Systems: An Application-Orientation Approach*. 2rd Ed. New York: Pearson Education

Narang, R (2009). *Database Management Systems.* New Delhi: PHI Learning

Stephens. R, Plew. R and Jones, Arie. D (2012). *Sams Teach Yourself SQL*.4th Ed. New Delhi: Pearson; Dorling Kindersley

Time Frame

You need **Two** weeks to finish studying this unit.

3.1 Introduction

Those who have built a house before will agree with me that before one begins the actual construction of the house, they first get the house drawing done. The drawing is done by an architect. It is this drawing that the builder (Bricklayer) follows when he begins the construction of the house. You may also agree with me that the house drawing should be clear and logical so that it becomes easy for the Bricklayer to interpret and build the house according to the drawing. The house drawing in this example acts as a model for the real house to be constructed.

Similarly, in database design, a model of a database is developed before the actual creation of the database. This is refers to as a data model of a database. A model can therefore be said to be a thing used as an example to follow or imitate. A model imitates an original object. In database semantics/language, a model refers to a prototype. The data models are logical because they clearly and accurately depict data relationship based a certain agreed standards and practices in database designing such as the Entity Relationship Model.

3.2 Database design steps

As per practice, in the construction of a house, owners of the house especially women are interviewed or asked what the new house should have. Women take advantage of this to explain the features their new home should have. The information which is obtained is carefully examined and put together by an architect to produce a drawing that meets the needs of the users. The same practice is used in the world of database designing. Information is collected from the end users of the database in what is called **database requirements analysis**.

3.2.1 Database requirements analysis

For a database to be able to meet the organization's data processing needs, the designers ought to collect and analyse information about the organization's needs. According to Thomas and Carolyn (2013, p.288) requirements collection and analysis is the process of collecting and analysing information about the part of the organization that is to be supported by the database system, and using this information to identify the requirements for the new system. The system requirements defines what the system must do and what the team must build; thus information collection process sets expectations and helps define success. Database requirements analysis is vital as data and information collected form the basis for generating or creating data models, vital for the successful implementation of a database.

3.2.2 Conceptual design

Once all the requirements have been collected and analyzed, the next step is to create a conceptual schema/ model for the database using a high level conceptual data model.

This phase is called **conceptual design**. The result of this phase is an **Entity-Relationship** (ER) diagram or Unified Modeling Language (UML) class diagram. In this module, we will use the Entity- Relationship Model. This model describes how different entities (objects, items) are related to each other. It also describes what attributes (features) each entity has. During or after the conceptual schema design, the basic data model operations can be used to specify the high-level user operations identified during the functional analysis. This also serves to confirm that the conceptual schema meets all the identified functional requirements. In short, during conceptual data modeling, the Entity –Relationship diagrams are used to show the relationship among entities and how data relates to each other.

3.2.2.1 Entity Relationship Model

Before we discuss in details the entity relationship model, it is important for us to know the meaning of the term **entity**. According to David (2005), an entity is an object or a thing that has physical or conceptual existence. It is something one wants to track by keeping information on it. Entities with physical existence are those that we can see with our eyes which include employee, student, petty, book, library user and car. On the other hand, entities with conceptual existence are things that we cannot feel, touch and see with our eyes. These include among other things; employee contract, library membership, product sales, circulation of library material, and year of study. Organizations keep information on these various entities by creating databases. For example, a library will keep information on library users, library materials and circulation of library materials. This will entail that the library is keeping information on three entities; namely library users, library materials and circulation of library materials. Note that entities of a given type are grouped into **entity class.** Therefore, the *library user entity* class will be a collection of *library user entities*

These entities have number of instances (meaning the number of times the entity appears in the database). This is what is referred to as **entity instance.** If for example you enter 100 users of the library in the database, it implies that the entity class *library user* has 100 instances.

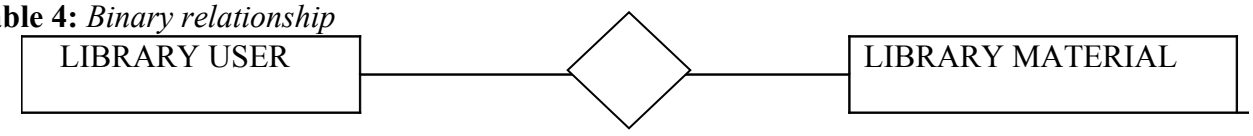
It is important to mention that all entities whether with physical or conceptual existence have **attributes**. Attributes are features that help to describe them. For example, the entity library user has many attributes which include; *First Name, Last Name, User Id number, Address, E-mail, and Phone #.*

The entity library material could have attributes which include; *Material Title, Author, Accession number, Year of publication and Classification numbe*r. On the other hand, the entity circulation of library material may have attributes such as *Date borrowed, Date due, Title borrowed and User names.*

It is also worth mentioning that all entities do have **identifiers.** Identifiers are attributes that uniquely identify each and every entity instance. It is an attribute which is not common to all the entity instances. For library user entity instances, the identifier could be the **User Id number** because each instance will have a different Id number. For the Library material instances, the **accession number/barcode number** will be the identifier as each material holds a different accession number or barcode number. These identifiers are turned into *Primary Keys* (Fields) during the logical and physical designs of the database.

On the other hand, the term relationship refers to association that exist among entities. The E-R model contains both **relationship classes** and **relationship instances** (David; 2005). A relationship instance exists among entity instances. Note that all relationships are given names that describe the nature of the relationship. For example, a relationship called *Loan relationship* shows the relationship that exists between the library user and library material. The relationship can involve two or more entity classes. The number of entity classes in a relationship is referred to as the **degree** of relationship. The *Loan relationship* above could be said to be a **degree of two.** The relationships of degree two are called *binary relationships* while those involving degree three are called *ternary relationships*.

Table 4: *Binary relationship*

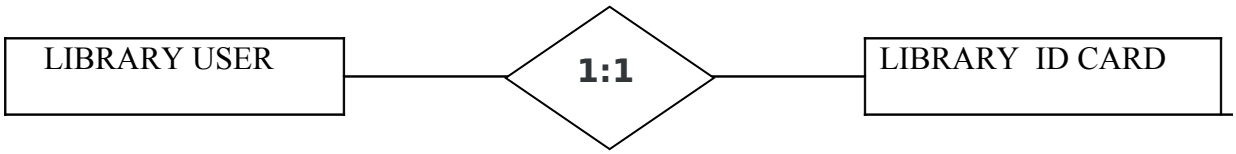


Loan

In any Entity –Relationship Model there is mapping cardinalities. This entails indicating the number of entities with which another entity can be associated with via a relationship. According to Rajesh (2005), the degree of relationship is called cardinality. There are

maximum and minimum cardinalities. Maximum cardinality entails the maximum number of entity instances that can participate in a relationship instance. Minimum cardinality implies the minimum number of entity instances to take part in relationship instance. There are basically three maximum cardinalities; *namely one-to- one, one- to-many* and *many-to-many relationship.*

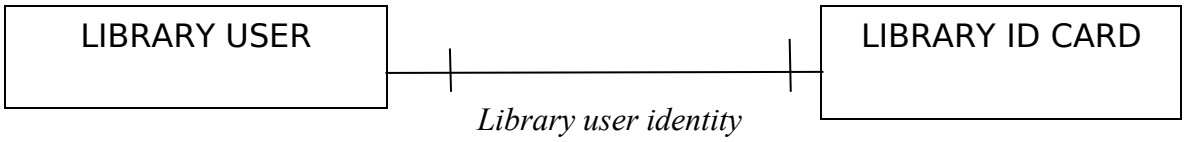
Figure 5: *One-to-One Relationship*



Library user identity

In other literature, the one to one relationship will be represented as shown below:

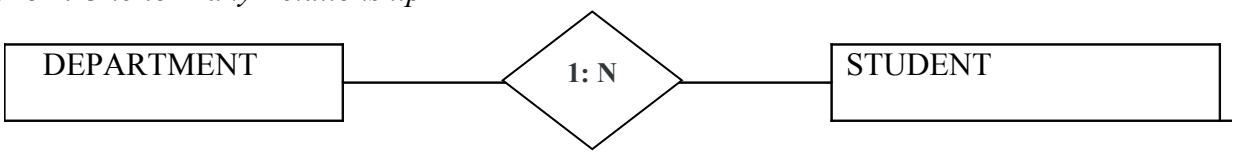
Figure 6: *One-to-One Relationship*



Library user identity

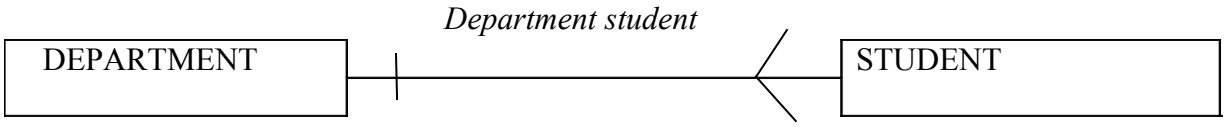
In this one-to-One Relationship, one instance of entity class LIBRARY USER is associated with one instance of the entity class LIBRARY ID CARD. This implies that in this relationship, one user of the library can only have one library ID card. Vice versa, one Library ID card can only belong to one user of the library. You will never see one library user having many different library IDs and the opposite is true. You will never find one library ID card given to many library users.

Figure 7: *One-to-Many Relationship*



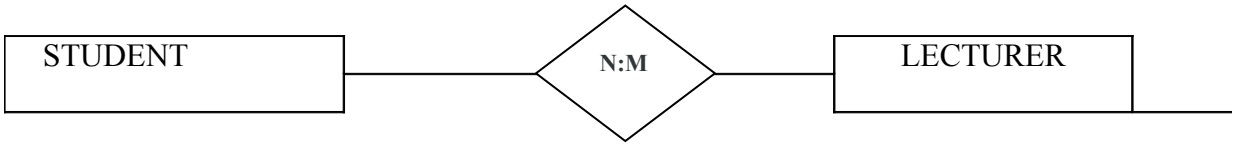
Department student

Figure 8: *One-to-Many Relationship*



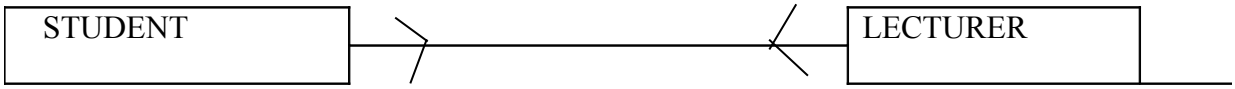
In the One-to-Many Relationship above, one instance of the entity class DEPARTMENT is associated with many instances of the entity class STUDENT. This implies that one department has several students. You will agree with me that one department can have many students or workers. The relationship which exists in this case is One- to- Many. Note that the letter **N** is used to denote the word many in Entity- Relationship drawings.

Figure 9: *Many-to-Many Relationship*



Student-Lecturer

Figure10. *Many-to-Many Relationship*



Student-Lecturer

In the Many-to-Many Relationship, one instance of the entity class STUDENT can be associated with many instances of LECTURER entity class. On the other, one instance of the entity class LECTURER can be associated with many instances of the entity class STUDENT. This entails that one student could have many lecturers and one lecturer can have many students

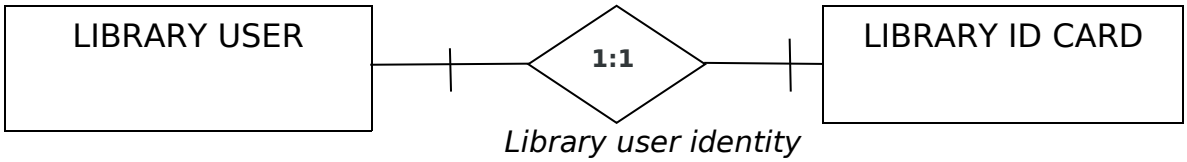
You may be wondering why we write the Many-to-Many Relationship as **N: M**, instead of **M: M** .The reason is that cardinality in one direction may be different than the cardinality in the other direction. In other words, in an **N: M** relationship, **N** need not equal **M**. If we look at

the Many-to-Many relationship above, one student can have maybe four lecturers and it is possible to for one lecturer to have 100 students. Therefore the cardinalities in both directions are not equal; hence we cannot use **M: M** but **N: M**.

3.2.2.2 Optional and none-optional relationships

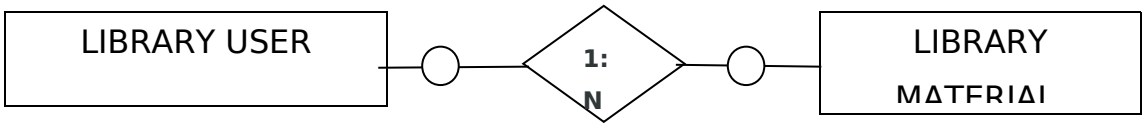
In any relationship there must be a minimum number of entity instances taking part. The minimum are either **zero** or **one**. If zero, then the participation in the relationship is optional, if one, then at least one entity instance must participate in the relationship. The optional relationship is represented by a **small circle** while the mandatory are represented by a **pipe** symbol (|)**.**

Figure 11: *Mandatory-to-mandatory Relationship*



The above example entails that each library uses requires an ID card and that a ID card must be allocated to at least one student. This relationship is compulsory.

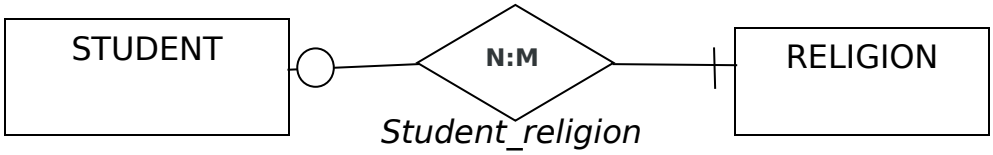
Figure 12: *Optional-to-Optional Relationship*



Library user_library material

The above relationship implies that a library user needs not to have a library material and a library material need not to be assigned to a student. This relationship is sometimes referred to as 0-0 relationship

Figure 13: *Optional-to-Mandatory Relationship*



The above relationship entails that a student must at least belong to one religion and that a religion may not be associated with a student. It is entails that it is not compulsory for a student to belong to a religion

3.2.3 Logical design

In logical design, you are required to show entity relationships as we did in the conceptual design and identify attributes for each and every entity that is taking part in the relationship. You need further to identify attributes that will be designated as both **primary** and **foreign keys.** A primary key is an attribute that uniquely identifies each instance or record in a database. Primary key could include *Student ID Number*, *Course Code*, and *Employee Number*. These attributes of an entity cannot be shared among instances of an entity. Put simply, you cannot find many instances having the same above attributes. For instance, each student is given a unique student number different from the rest. Similarly, different courses have different course code**s.** On the other hand, a foreign key is an attribute (field) of an entity instance that matches the primary key of another entity instance. For example, if the entity LIBRARY USER has an attribute **user ID number** and another entity called CIRCULATION has an attribute **user ID number**, this entails that user ID number which is a primary key in LIBRARY USER entity is a foreign key to the entity CIRCULATION. Similarly, the **barcode number for the material** in the LIBRARY MATERIAL entity can be said to be the foreign key for the entity CIRCULATION and vice versa.

These entity relationships and attributes need to be diagrammatically presented. You may agree with me that attributes are features or properties of an entity. Let say for the entity LIBRARY USER, the attributes could include the following:

User ID number, First name, Last name, Year of study, Physical address, Phone number, E-mail

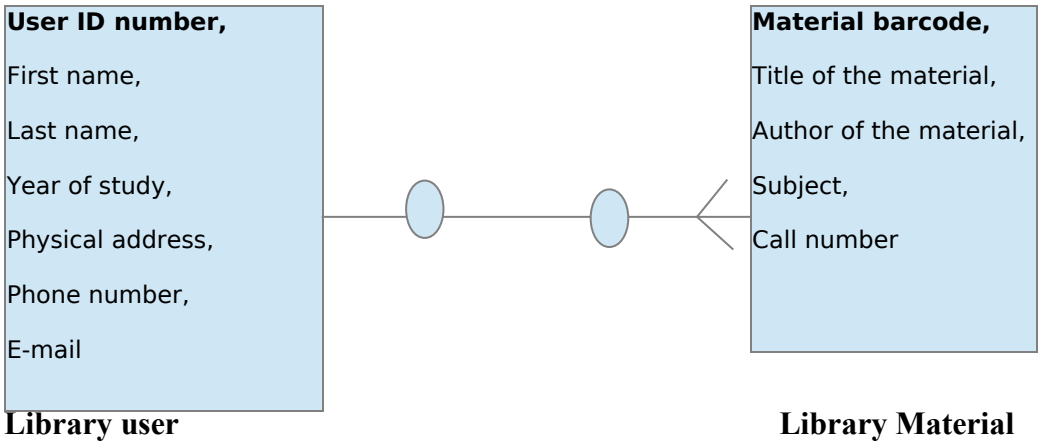
For the entity CIRCULATION, attributes could include:

Date borrowed, Date due, User ID number, Library material barcode number

For the entity LIBRARY MATERIAL, attributes could include:

Title of the material, Author of the material, Barcode number for the material, subject, Call number

Figure 14: *Entity Relationship*



In the above example, the less than symbol (<) shows that many library materials can be associated with one library user. This is **One-to-Many relationship** that exists between Library user entity and Library material entity as one user of the library can be associated with many books at a time. However, one book cannot be associated with many users at a time. The two circles indicate that the relationship between two entities is an optional one.

3.2.3.1 Normalization

After diagrammatically representing the entities attributes and their relationships, there is need to conduct the normalization process. The goal of normalization is to eliminate redundancy and potential update anomalies. Redundancy means that the same data is saved more than once in a database. Update anomaly is a consequence of redundancy. If a piece of data is saved in more than one place, the same data must be updated in more than one place. If for example when you are were identifying and listing attributes for the CIRCULATION, you include attributes such as user first name, User last name, title of the material etc, you need to remove these entities from the CIRCULATION entity as it will be a repetition of attributes already defined in LIBRARY USERS and LIBRARY MATERIAL entities. Failure to remove the twice appearing attributes will result in wastage of storage space and data redundancy. Normalization is a technique by which one can modify the relation schema to reduce the redundancy. There are **five** normalization forms that are used. These are first (1NF), second (2NF), third (3NF), fourth (4NF) and fifth (5NF) normal forms. **Note** that database designers do not use all the five normal forms but in most cases they just use the first three normal forms.

First Normal Form (1NF),

In the first normal form, there is need to ensure that no two or more rows contain repeating information. Each column should therefore be unique thus creating unique rows. Further, 1NF advocates that each row should have **a primary key**. And that no column should keep more than one value. Let see how we can apply first normal form to the below table.

Table 4: *Student table*

Name	Sex	phone#
Danny	Male	+26097655443,+26099544463
Maureen	Female	26096776778

In the above example, phone# column has defiled the 1NF in that they contain more than one value. Secondly, there is no primary key. To solve the first problem, we need to duplicate the entry of Danny because of two phone numbers (two values).

Table 5: *Student table under 1NF*

StudentID	Name	Sex	phone#
2000200	Danny	Male	26097655443
2000200	Danny	Male	26099544463
2133355	Maureen	Female	9677677888

1NF has solved one problem but created the problem of redundancy which the subsequent NF should be able to solve.

i. Second Normal Form (2NF)

2NF emphasizes that there should be no partial dependency of any column on a primary key. This means that all the columns (fields) should depend entirely on the primary key. In the table above, one may say that the column **phone#** does not strongly depend on the primary field hence the need to create a two related tables as shown below.

Table 6: *Student table under 2NF*

StudentID	Name	Sex
2000200	Danny	Male
2000200	Danny	Male
2133355	Maureen	Female

Table 7: *Phone# table under 2NF*

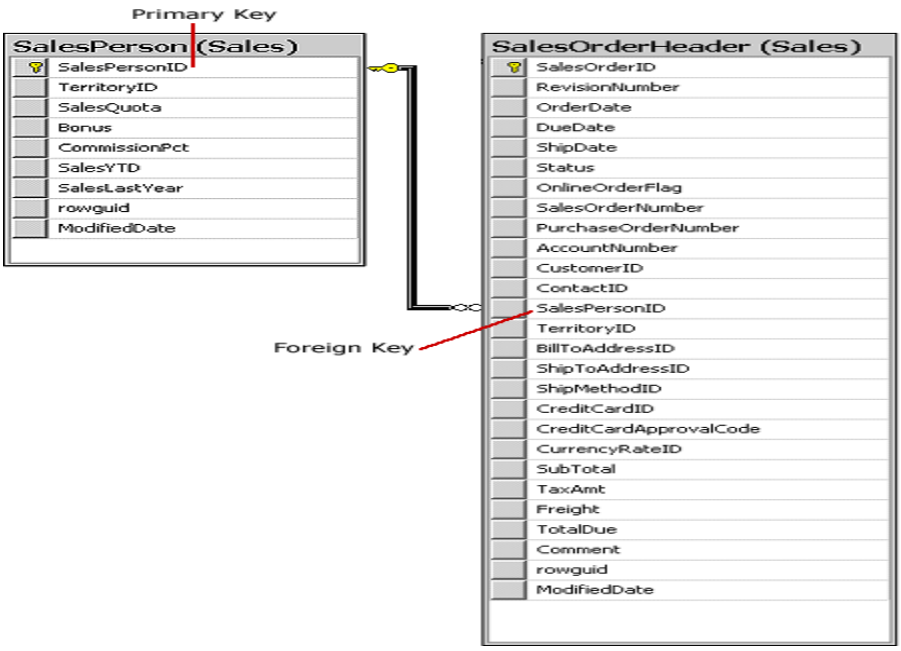
StudentID	phone#
2000200	26097655443
2000200	26099544463
2133355	9677677888

ii. Third Normal Form (3NF)

The table is said to be in 3NF if it meets the requirements of 2NF and that none- key attribute in the row does not depend on the entry in another column. There should be no transitively dependency on a primary key and none- primary columns. In the 2NF first table, **sex** can be said to transitively depend on studentID. This will entail split that table into two so in order to meet the requirements of 3NF.

Note that foreign keys are important in the creation of relationships among tables (entities). They also help to enforce data integrity as data in related tables cannot be deleted without deleting data in the table containing the primary key which is a foreign key in another table.

Table 8: *Relationships among tables*



Source: Kroenke, D.M (2005)

In this diagram above, for one to delete data in SALESORDER, he/she needs to begin deleting data in the SALESPERSON table. Otherwise, you will be unable to delete data in the SALESORDER

3.2.4 Physical design

The goal of the last phase of database design, physical design, is to implement the database. You need to decide on the number of tables to create for your database. Normally the number of tables is determined by the number of entities. If you have two entities, you will create two tables. If they are three entities, you will create three tables. Further, you need to decide on the names of your tables. It is advisable to use the entity's name to represent the table name. For instance, if I have the entity LIBRARY USER, the table name should be **library user**. Note that the name of the table should be devoid of spaces. In addition, you need to decide on the number

of column to have for your tables. Columns are referred to as **fields**. Columns are a translation of the attributes of an entity. For the LIBRARY USER entity, columns could include *User ID number, First name, Last name, Year of study, Physical address, Phone number* and *E-mail.*
There is need also to decide on the data type each field will have. Different DBMS's have different names for data types and have different data types. You need also to decide on which field to be indexed to create an index and define integrity constraints (rules) and the users' access rights. Below are the activities that are performed in **three stages** of data modeling.

Table 9: *Data models and their features*

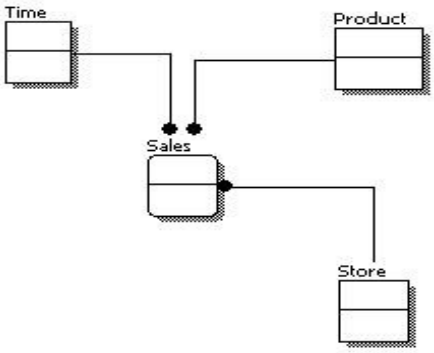
Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

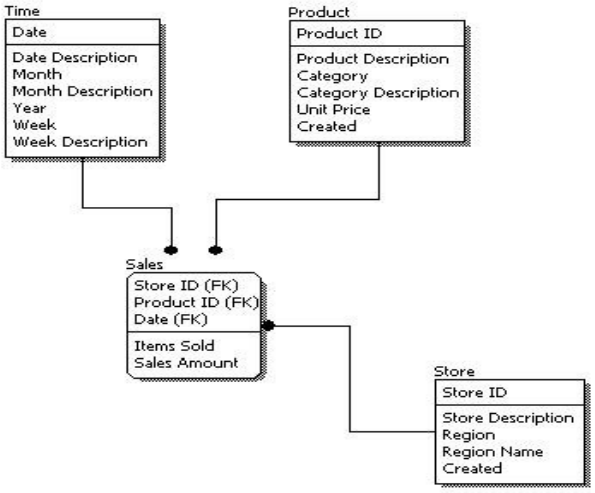
Source: Kroenke, D.M (2005)

Figure 15: *Conceptual and Logical design models*

Logical Design Model

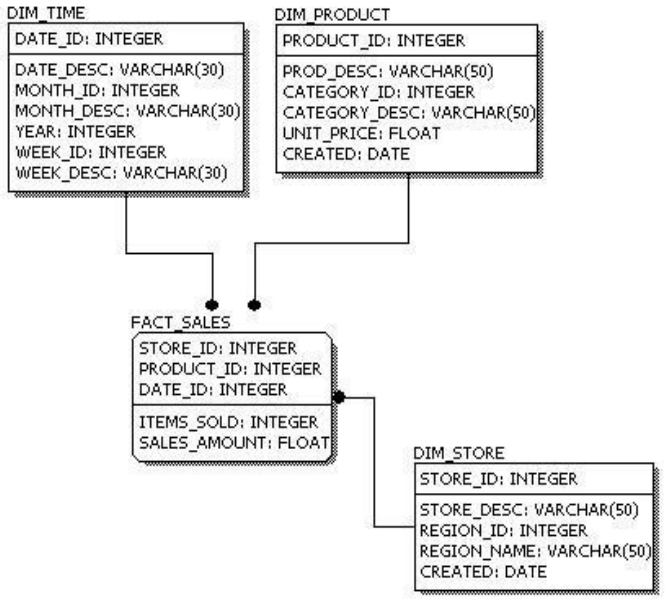
Conceptual Model Design





Source: Kroenke, D.M (2005)

Figure 16:*Physical Design Model*



Source: Kroenke, D.M (2005)

Summary

In this chapter, we have learnt that before one creates databases, there is need to have a database model in place. A model is something that represents a real object. For you to come up with a database model, you need first to collect information from those who will be the users of the database and analyzed how information is used by users in the organization. This process is called *information gathering* and *analysis*. After collecting information, you need to begin creating a data model by following **three** steps or phases. These phases are *conceptual, logical* and *physical design*. In the conceptual design, you begin by identifying entities; objects or things that you want information kept on them. Entities could have physical or conceptual existence.

For instance, *library user* could be an entity with physical existence because you can see the user with while *circulation* could be an entity without physical as it cannot be seen. Entities that are related are grouped together as entity classes. The number of times an entity appears in the database is called entity instance. Further, we learnt that entities have attributes-features which describe them. For example, the entity **library_ user** may have the following attributes: *User_ID, Name,* and *Address*. In the conceptual design, you need also to show the relationships among these entities in what is called the *Entity-Relationship Model.* We have learnt that the relationship in which they are two entities is called *binary relationship* while a relationship with three and above entities is called *Ternary relationship.* We further learnt that the degree of entities participating in a relationship is called the degree of relationship. The relationship among entities could be *One-to-One, One-to-Many or Many-to-Many*. In One-to-One, one instance of an entity class is associated with one instance of another entity class and in the One-to- Many, one instance of an entity class is associated with many instances of another entity class while in the Many-to-Many relationship, many instances of an entity class is associated with many instances of another entity class.

In logical design, you go beyond identifying entities and their relationships by listing all the attributes in each and every entity class. Therefore after, identify *primary* and *foreign keys* from the attributes identified in each entity class. Primary key is field or attribute that uniquely identifies an entity instance. For example, in the entity class **library_user** , the attribute *user_id* could be taken as a primary field. A foreign key is an attribute that matches the primary key of another entity instance in another entity class. For example, if *user_id* is used in the **Circulation** entity class, it can be said to be a foreign key as it matches the primary key in the Library_user entity class.

We learnt that the last stage in the database design is called physical design. At this stage, you need to you need to determine the number o tables to create. Normally the number of entities determines the number of table. You need also to decide the name of your tables. It is advisable to use entities for table names and avoid using capital letters and spaces for table names. Further, you need to know the number of columns each table will have. In this case, the identified attributes will be taken as your columns or fields. Among these fields, you need to identify some that can be indexed to create an index.

Activity

Get a piece of paper and pencil, then list down the attributes of the entities *library user, library_material, and circulation***.** Show how they relate to each other by drawing the Entity-Relationship Model.

UNIT4: Introduction to SQL

Introduction

Unit four introduces you to the database language commonly known as SQL. This is the language that we will use to design a database in MySQL database management system. It begins by giving you a history of the language and then the SQL syntax and explains the types of SQL commands.

Aims

This unit aims at explaining the history, syntax and commands of SQL. It also aims at explaining the categories or types of SQL statements or commands.

Objectives

At the end of this unit, you should be able to do the following:

- Explain the history of SQL
- Explain the features of syntax and commands of SQL
- Explain SQL command categories

Equipment needed

This unit is also theoretical; you will therefore not need a computer to cover this part of the module.

Other Resources

You need books and information resources that discuss SQL and MySQL database management systems. These books include the listed below:

Beighley. L and Morrison. M (2009). *Head First PHP &MySQL*. Cambridge: O'Reilly.

Chao.L (2006). *Database Development and Management*. Boca Raton: Taylor &Francis Group.

Hoffer, J.A, Ramesh. A and Topi. H (2011). *Modern Database Management*. New Delhi; Chennai: Pearson; Dorling Kindersley

Kroenke, D.M (2005). *Database Processing: Fundamentals, Design, and Implementation*. 10th Ed. Beijing: Publishing House of Electronics Industry

Kifer. M, Bernstein.A and Lews, M.P (2005). *Database Systems: An Application-Orientation Approach*. 2rd Ed. New York: Pearson Education

Narang, R (2009). *Database Management Systems.* New Delhi: PHI Learning

Stephens. R, Plew. R and Jones, Arie. D (2012). *Sams Teach Yourself SQL*.4th Ed. New Delhi: Pearson; Dorling Kindersley

Time Frame

You need **One** week to finish studying this unit.

4.1 Introduction to SQL

You may agree with me that we have once in life fail to communicate with people who speak a different language. If we fail to communicate, we call for a translator; an intermediary who understand both languages. In today's server databases, client computers need a language; a translator to enable access to data in the database. SQL is the perfect translator or intermediary. SQL is more than a translator as it is a language used not only to manipulate data in database but also to create,update and delete it. It is widely used in robust or big database systems.

4.2 What is SQL?

The Acronyms SQL stand for **Structured Query Language**. Others argue that SQL does not stand for anything. According to Lynn and Michael (2013), SQL is the query language used to communicate with a database. Whatever the argument, SQL is a language used by all server databases .It is a set of statements that define and manipulate data. As earlier alluded to in the introduction, SQL is powerful that it can be used to create, modify and delete the databases or tables of a database. It is used by relational database management systems such as Oracle, Sybase, PostgresSQL, SAP, MySQL and Microsoft Office Access. In this module, we will learn how to use MySQL as it is free and widely used.

1.3 History SQL

SQL was originally developed by Dr. E.F.Codd who the 1970s was a Research Fellow at IBM Company in The United States of America. He wrote a scientific paper of how to develop a relational database management system based on *Structured English Query Language (SEQUEL)*. In 1979; the first DBMS based on Codd's SEQUEL called Oracle was developed and released. It is the SEQUEL which was later named SQL. Oracle is one of the best proprietary DBMS on the market today. Later, many organizations developed their DBMS based on SQL. SQL has been approved and recognized by many international organizations as a standard language in relational database communication. In 1986, the American Notational Standards Institute (ANSI) endorsed SQL as a standard language. Further, in 1987, the ANSI

SQL standard was accepted as the international standard by the International Standards Organizations (ISO).ISO SQL standard of 1987 has been revised on several occasions. The latest ISO SQL is the SQL-2008.

1.4 SQL Syntax

In any language, there are rules that govern the way sentences are constructed. These rules have to be followed for one to effectively communicate. In English for example, one need to use punctuation marks such as commas, full stops etc. Similarly, SQL has rules to follow when constructing SQL statements. These SQL rules have to be followed for one to construct SQL statements that are correct and can be accepted by database management system. **Note** that many database management systems such as MySQL, Microsoft server and Oracle syntax are based on the human language, they have borrowed heavily from human language punctuations such as semi-colons. SQL statements also use symbols which the computer understands and acts upon. Note futher that there could be slit variations among database management systems in terms of syntax. In this module, MySQL database management system syntax will be used as examples. Below is an example of SQL commands/sentences.

SELECT* from students;

SELECT student_Id from students;

In the first example, the SQL statement or command tells MySQL database management system to select and display data on all (*) the fields from tables students. In the second example, MySQL database management system is being asked to select and display data only from a field called student_id from the table called students. The above two SQL statements are correct as the syntax has been followed. Your question could be: what if I did not put a **full colon** at the end of the SQL statement? The answer is simple; MySQL database management system will fail to execute your command because it is wrongly phrased.

1.5 Types of SQL commands

There are several SQL commands/statements used in the manipulation of data in a database management system .These commands are however grouped into six categories. These are;

i) Data Definition Language (DDL)

This SQL commands enable a database user to create and restructure database objects. With these commands, one can create, modify and delete the database and tables. These commands can also be used for creating the index to the tables. Examples of DDL commands include the following examples:

CREATE DATABASE students;

In this example, MySQL database system is being told to create the database called **students.**

DROP DATABASE students;

In this last example, the database systems is being told to delete the database called **students**.

ii) Data Manipulation Language (DML)

These are SQL commands that are used to do among other things; inserting data into tables, updating data and delete data from the tables. Examples of these SQL statements include the following:

UPDATE results SET surname =“Bwalya” WHERE id=2002033;

This above SQL statements instructs MySQL database systems to update the table called **results** by changing the value in the field surname to Bwlaya and the Where statement is a condition which will ensure that the surname value is not changed in other records (instances) but only where the value for student_id field will be equal to 2002033.

INSERT into results VALUES (2002033, “Bwalya Tuesday”, “40 %”);

This last SQL statement tells MySQL database systems to inserts data into the table called **results** in all the three fields which are *student_Id, Name and Percentage.*

iii) Data Query Language

This is basically one command or statement called SELECT. It is used as a query into the database. It is used to retrieve data from the database. Examples of SELECT statements include the following:

SELECT* from results;

This query display data on the fields from the table called **results.**

SELECT surname, firstname FROM results;

This query display data only on first name and surname fields from a table called **results.**

iv) *Data Control Language*

These commands in SQL enable you to control access to data in the database

They define access privileges. These include commands to ALTER PASSWORD, GRANT, REVOKE access privileges. Examples include the following:

grant all privileges on students.* to tuesday@localhost identified by 'lis2014';

This SQL statement tells MySQL database system to allow a user called **tuesday** all rights to the database found at local machine (server) by using a password called **lis2014.** All rights,means , the user can access data from the database and modify, delete it and the database**.**

revoke all privileges, grant option from 'tuesday'@'localhost';

This SQL for sure will make the user called **tuesday** fail to access the database.

V. Transaction Control Commands (TCC)

These are commands that enable the user to manage transactions in the database. Transactions include updating and deleting database, tables or data from the table rows. The most used TCC commands include the COMMIT, ROLLBACK and SAVEPOINT commands.

- COMMIT-Saves database transactions
- ROLLBACK-Undoes database transactions
- SAVE POINT-Creates points within groups of transactions in which to rollback

Note that all changes or transactions that are effected to many server database systems are initially stored in the Temporary Buffer. By executing the three TC commands you can validate or nullify the transaction. By executing the COMMIT command, you will empty the buffer and your data will be completely deleted from the table and database. To use TC commands, you may delete some data from the table using DML commands as follows:

DELETE FROM results WHERE student_id=20020333;

You can validate or completely delete data even from the temporal buffer by the use of the COMMIT command follows:

COMMIT;

From my experience, these COMMIT and ROLLBACK commands have no impact on MyQSL. Once you delete, it may be difficult to roll back data. Each MySQL session has the COMMIT command set as default. You can unset that at the beginning of any MySQL session (interaction with MySQL) by the below commands:

SET autocommit=0;

START transaction;

Then, you can continue with any transactional commands. If you delete something accidentally, you can try to roll back.

❖ *Data Administration Commands*

These commands enable system administrators to perform among other things:

a) Audit of the system-to know what transactions took place in the system etc

b) Performance analysis of the system- analyzes whether the system is performing according to its design purpose and objectives.

Commands used are START AUDIT and STOP AUDIT.

Summary

You have learnt in this unit that SQL stands for Structured Query Language; a language that is understood by all server based database management systems including MySQL**.** It is the language which was conceived by Dr. Codd in the 1970s when he worked as a Research Fellow at IBM corporations in the United States of America. IBM was the first company to create a database management system based on SQL. Thereafter, other organization s adopted SQL in their database management systems. Currently, major database management systems such as MySQL, Oracle and Windows database systems use SQL. SQL is an international standard language for database design. SQL is a language based on English statements which are based on accepted syntax. The syntax should be followed strictly. In SQL statements, there are punctuations and symbols that should be observed if one has to write a valid SQL command or statement. We have learnt also that there are six categories of SQL commands. These are data definition commands (create and delete databases and tables), data manipulation commands (update, delete etc data in the tables of a database), data query commands (retrieve data from the tables of the database), data control commands (create or revoke access privileges), transactional control commands (perform rollback of data once deleted etc) and data administration commands (used to conduct system audit etc).

Activity

Without referring to your module, explain the **six** categories of SQL.

Unit 5: MySQL database management system installation

Introduction

This unit will show you how to install MySQL database management system. It however begins by giving you a brief history of MySQL and its features. Thereafter, the unit shows you step by step how to install and test run MySQL and its supporting software.

Aims

This unit aims at showing the installation procedure of MySQL database management systems. It also aims at explaining the history and features of MySQL database management system.

Objectives

At the end of this unit, you should be able to do the following:

- Explain the history of MySQL
- Explain the features of MySQL
- Know how to install MySQL and its accompanying software
- Know how to test run MySQL and its accompanying software

Equipment needed

This unit is practical; you therefore need a computer. You need also to have access to the Internet to download the MySQL database management system and supporting software.

Other Resources

You need books and information resources that discuss SQL and MySQL database management systems. These books include the listed below:

Beighley. L and Morrison. M (2009). *Head First PHP &MySQL*. Cambridge: O'Reilly.

Chao.L (2006). *Database Development and Management*. Boca Raton: Taylor &Francis Group.

Hoffer, J.A, Ramesh. A and Topi. H (2011). *Modern Database Management*. New Delhi; Chennai: Pearson; Dorling Kindersley

Kroenke, D.M (2005). *Database Processing: Fundamentals, Design, and Implementation*. 10th Ed. Beijing: Publishing House of Electronics Industry

Kifer. M, Bernstein.A and Lews, M.P (2005). *Database Systems: An Application-Orientation Approach*. 2rd Ed. New York: Pearson Education

Narang, R (2009). *Database Management Systems.* New Delhi: PHI Learning

Stephens. R, Plew. R and Jones, Arie. D (2012). *Sams Teach Yourself SQL*.4th Ed. New Delhi: Pearson; Dorling Kindersley

Time Frame

You need **One** week to finish studying this unit.

5.1 Introduction

MySQL is the widely used database management system in the world. According to Wikipedia (2014), as of March, 2014, MySQL was the world's second most widely used open-source relational database management system (RDBMS). It used by many organizations for managing their databases or database applications. MySQL is also the database system that powers many Contents Management Systems such as Joomla, Drupal and Word press (these are software used to create modern websites.) It is also the engine behind all the Integrated Library Management Systems (Koha and Openbiblio) that you will learn in Module 2.

5.2 History of MySQL

MySQL is pronounced as “My Sequel”. It is named after co-founder Micheal Widenius’s daughter, **My**. In the previous unit, we learnt that SQL phrase stands for Structured Query Language. It was initially (first) released in 1995. MySQL was developed by the Swedish company called MySQL AB. But it is now owned by Oracle Corporation. MySQL is available in two licenses namely: GNU General Public License and a variety of proprietary agreements. Under General Public License, MySQL is released as a Free and Open Source Software (FOSS) while under proprietary license; it is released as commercial software such as Oracle and Microsoft SQL database system. For this course we will use FOSS version of MySQL. It works well and widely used to created databases.

5.3 Meaning of Free and Open Source Software (FOSS)

Free and Open Source software is software which is freely available to the public to use at the same time its source code is publicly available for users to modify it so that it suits the users need. The source code is the actual instructions of the computer written in programming language which is behind the application software we use. Since the source code is available for all FOSS, if one has programming language skills, he/ she can modify the software by playing with the source code. Note that all FOSS are released under General Public License (GNU-GPL). This implies that this software can be used and distributed to other people and organizations freely.

5.4 MySQL supporting software

As you will see very soon, MySQL is a server based database management system. Therefore it should be installed on a server. This implies that you will need to install server software first. Don't worry about where you will get the server software. We will use **Apache** which is also a Free Open Source Software. The latest version of MySQL is 5.6 as at 2014 but we shall be using MySQL client version: 5.1.41 and Apache server version 1.7.3. The current version of Apache by the day of writing this module was Apache 1.8.3.

5.5 Source of MySQL software

Being a Free and Open Source Software, MySQL can be downloaded from the Internet for free. For this purpose, we will download the software called XAMPP. This XAMPP contains not only MySQL but also Apache and other MySQL supporting programs such as **PHP** (Pre text Processor), the software which you may need to send Queries to MySQL database when you will be creating library systems and **FileZilla** (used to transfer web pages from your local machine to the external machine) etc. In old days, one needed to download MySQL, Apache and other accompanying software and install them separately. This has changed. You will just download XAMPP; a program comprising four software which include Apache and MySQL , PHP, Filezilla and Mercury. You can download XAMPP from the following link http://xampp.en.softonic.com. **Note** that you can also download **Wamp** for windows; it also works and has all the main software.

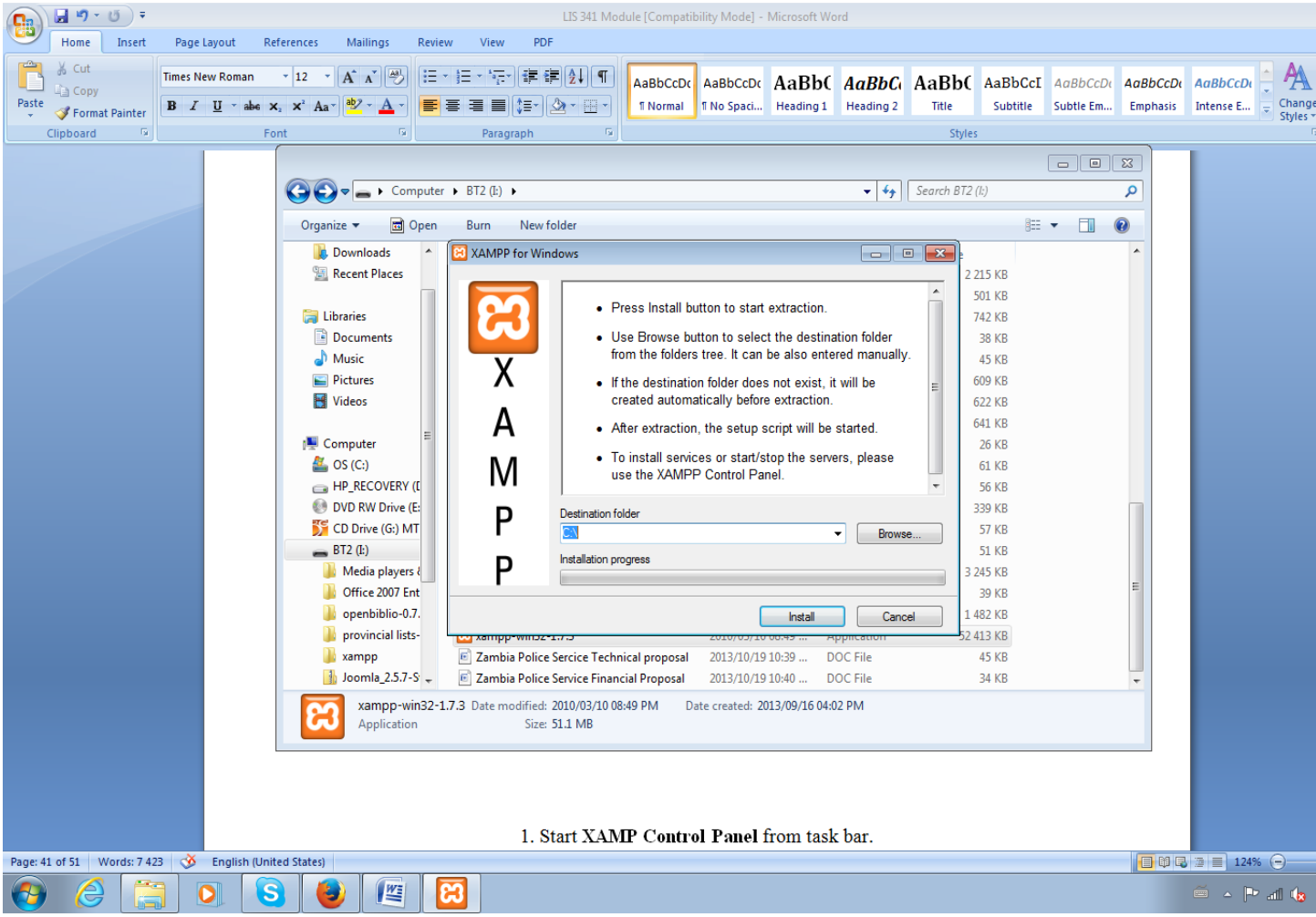
5.6 Installation requirements

It runs on both windows and Linux platforms. We will download and use XAMMP for Windows. Xampp can be installed on a machine with 500MB of RAM and at least 10 Gbs of Hard disc. These requirements have been met as many computers whether Laptops or Desktops have more than 1Gbs of RAM and more than 80Gbs of hard disc. This implies that any modern computer today can take on Xampp. Further, your computer should have Windows XP, Vista, Windows 7 or a better version as your operating system software. You need also to install a browser such as Internet Explorer or Mozilla Fire Fox. The later is preferred.

5.7 Installation procedures

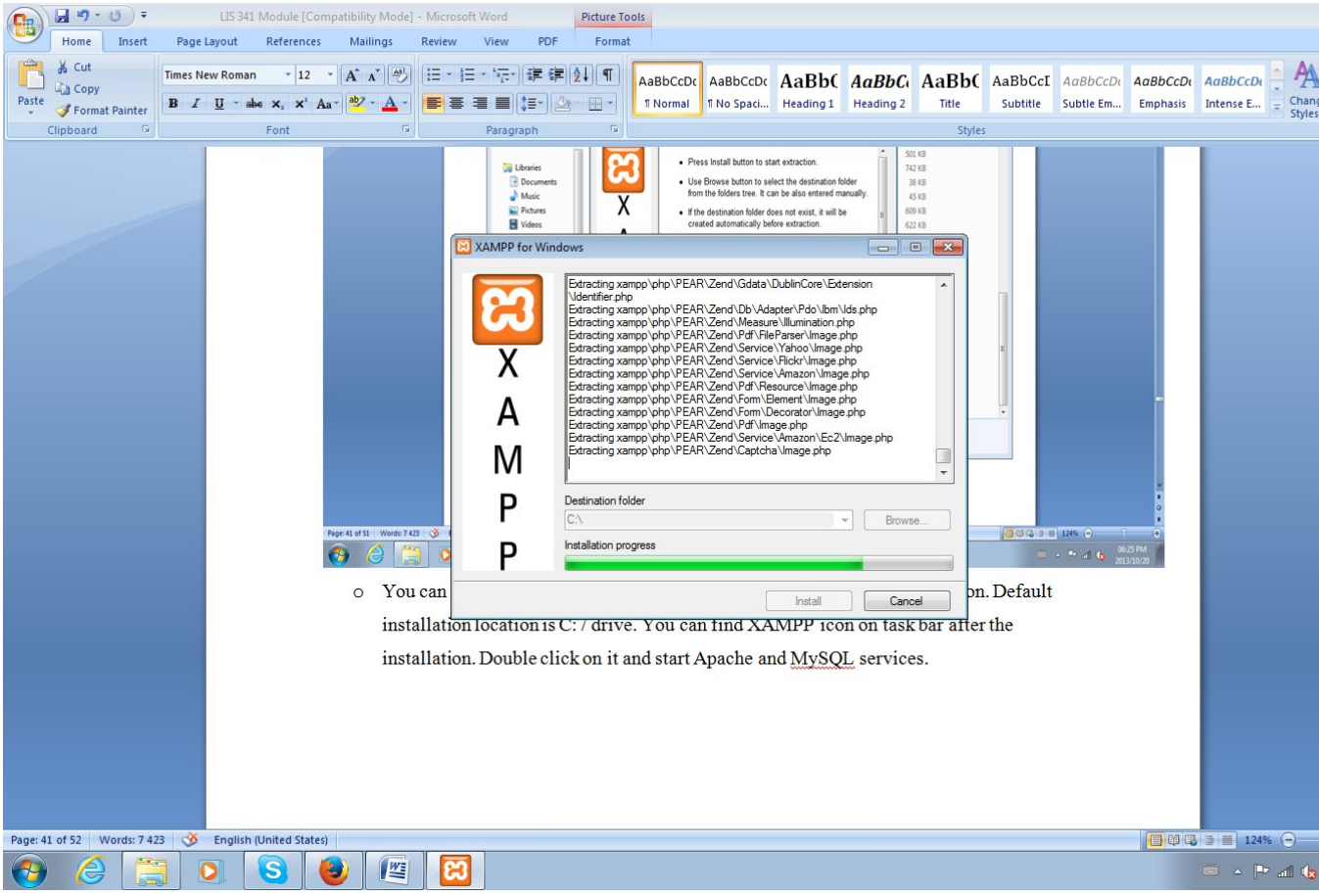
Once you have successfully downloaded Xampp, double click on the xampp.exe file and start installation. Xampp will be loaded on the machine and it will display this.

Figure 17: *Installation of Xampp*



- As you can see, Xampp wants to be installed in **C** drive by default. Don't change the installation drive. Just leave the default settings and click on the **install** icon.

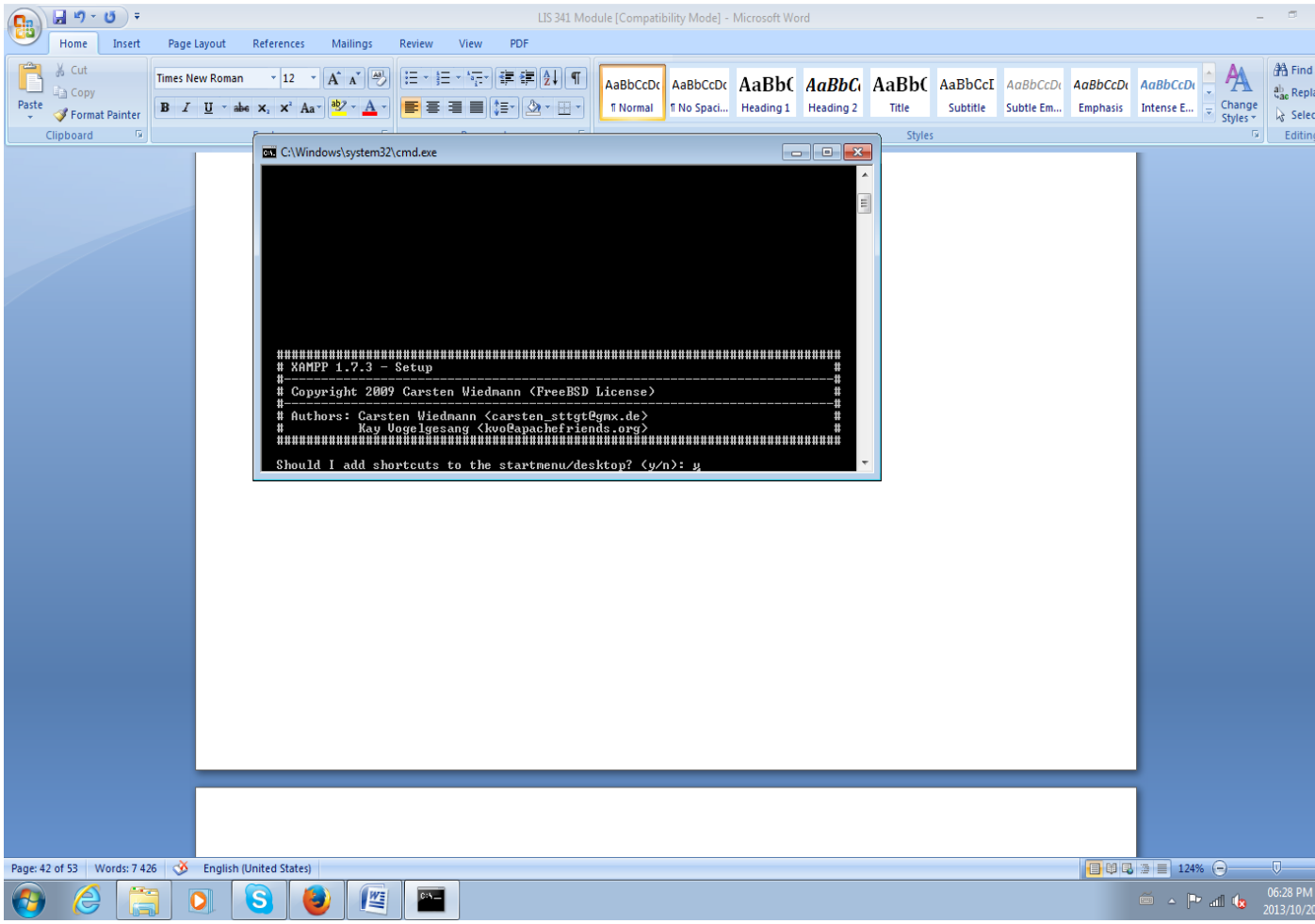
Figure 18: *Installation of Xampp*



- You can … on. Default installation location is C: / drive. You can find XAMPP icon on task bar after the installation. Double click on it and start Apache and MySQL services.

Figure 19: *Finishing the Installation of Xampp*

-

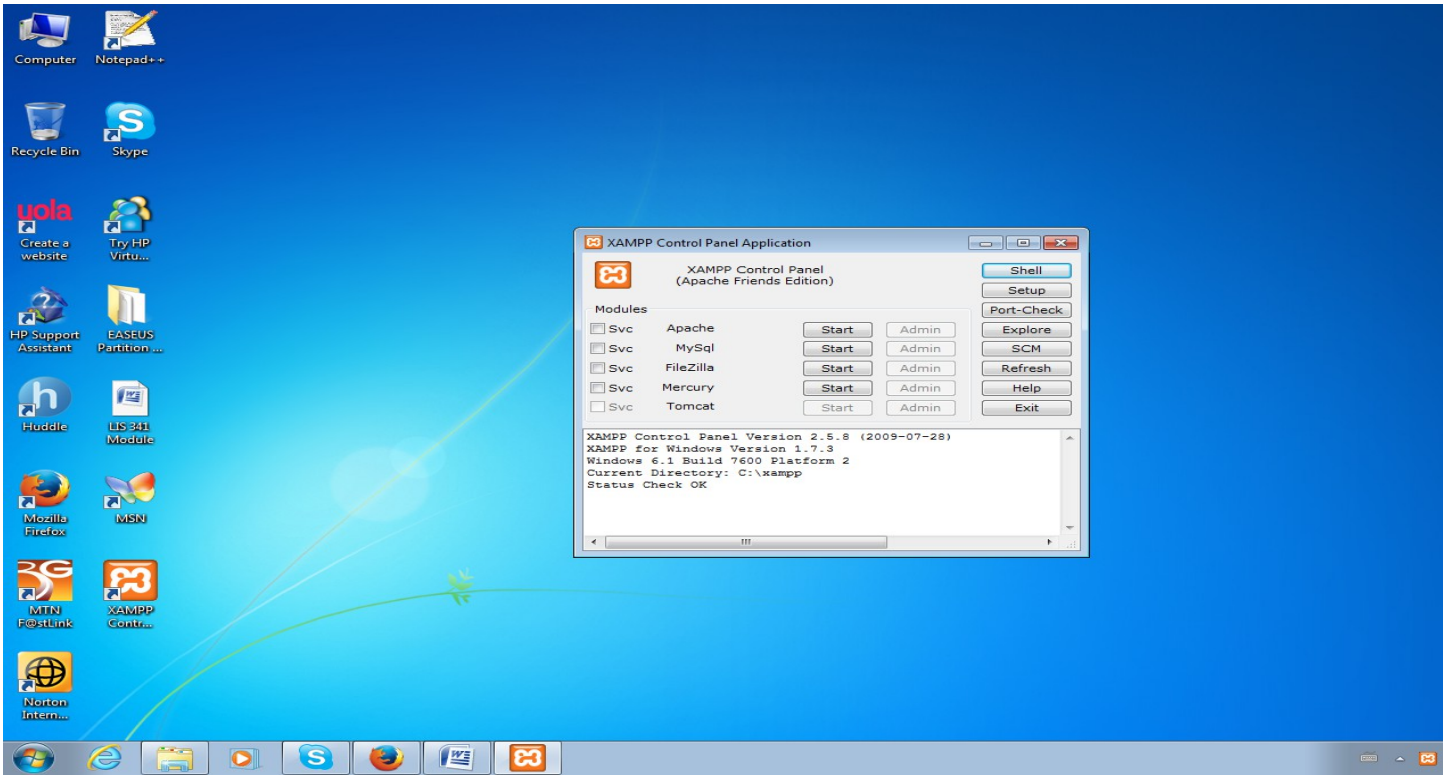


- When this window appears, just press enter button and do it for another three times before you enter **x** in the box to tell the machine to exit and that we will mark the end of your installation. The installation is successfully finished.

5.7 Starting Xampp

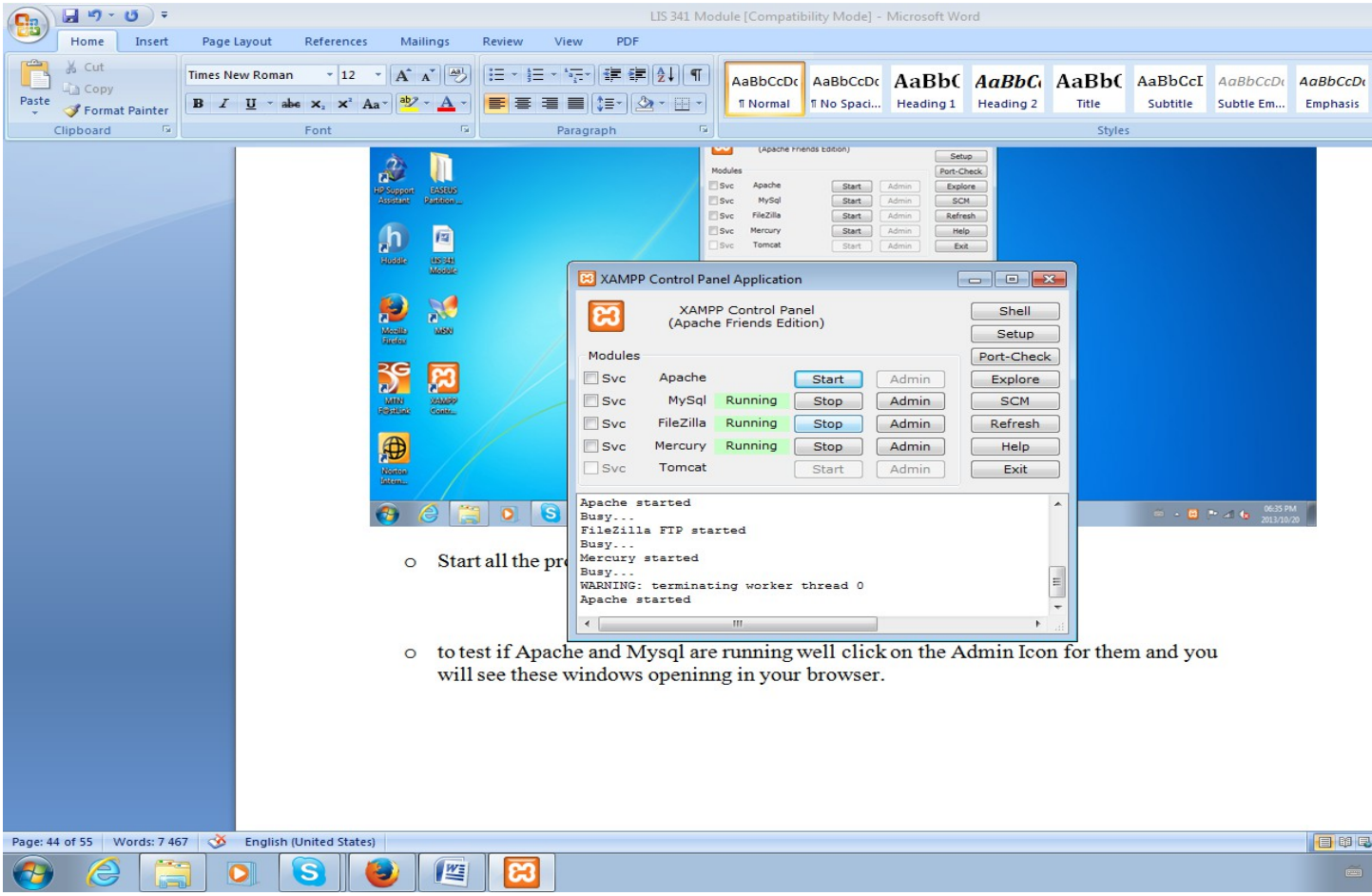
For you to start Xampp, go to the Desktop and find a Xampp shortcut icon. Double click it and you will be greeted by the below screen.

Figure 20: *Starting Xampp*



- Start all the programs installed by Xampp, by clicking on Start.

Figure 21: *Starting all Xampp programs*

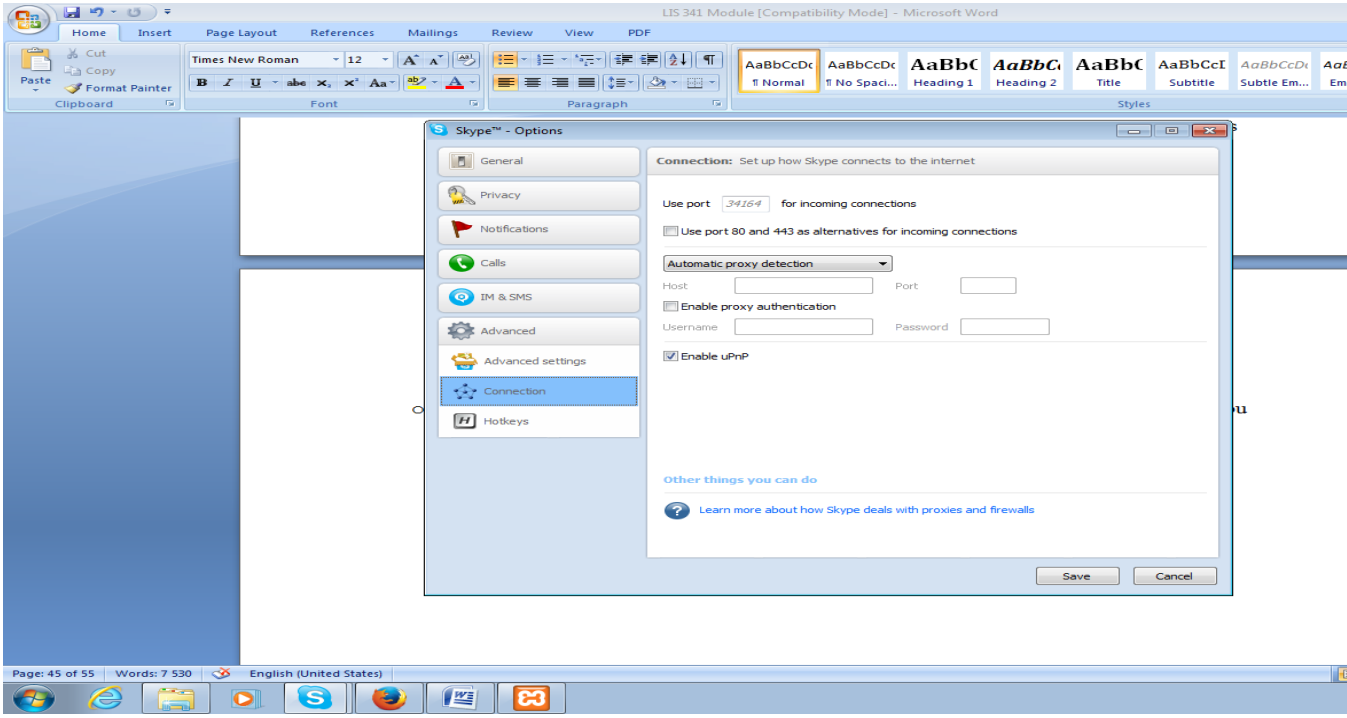


5.7.1 Trouble shooting Apache's failure to start

If Apache is failing to start, it means that there is software on your computer which is using port 80 and 443. By the way, what is a port number? A port number is simply a part of the addressing

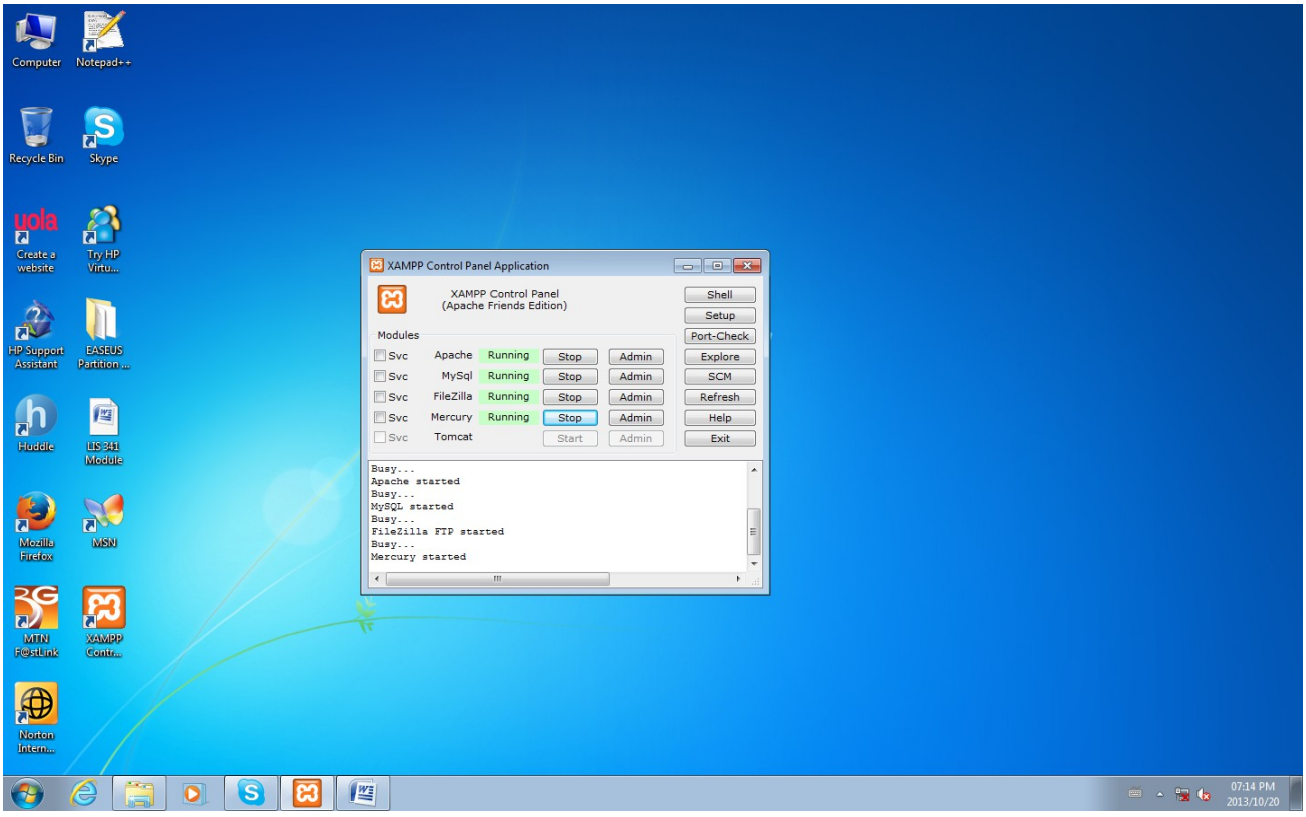
information used to identify the sender and receiver of a message on a network. Port numbers are like our telephone extension numbers that each and every office has an organization. For instance, the office the Head of Library and Information Studies Department at UNZA has a telephone extension number 2211 which he/she uses to receive and make local calls. Similarly, all network based software (include Apache) do have port numbers that they use to receive and send information on a network. Apache in this case uses port 80 and 443 on any computer network. In the above example, Apache has failed because I have installed Skype which also uses port 80 and 443. I need therefore to disable Skype. I have to open Skype and click on tools, then options, then click on advanced and lastly on connection and un-tick port 80 and 443.

Figure 22: *Trouble shooting Xampp*



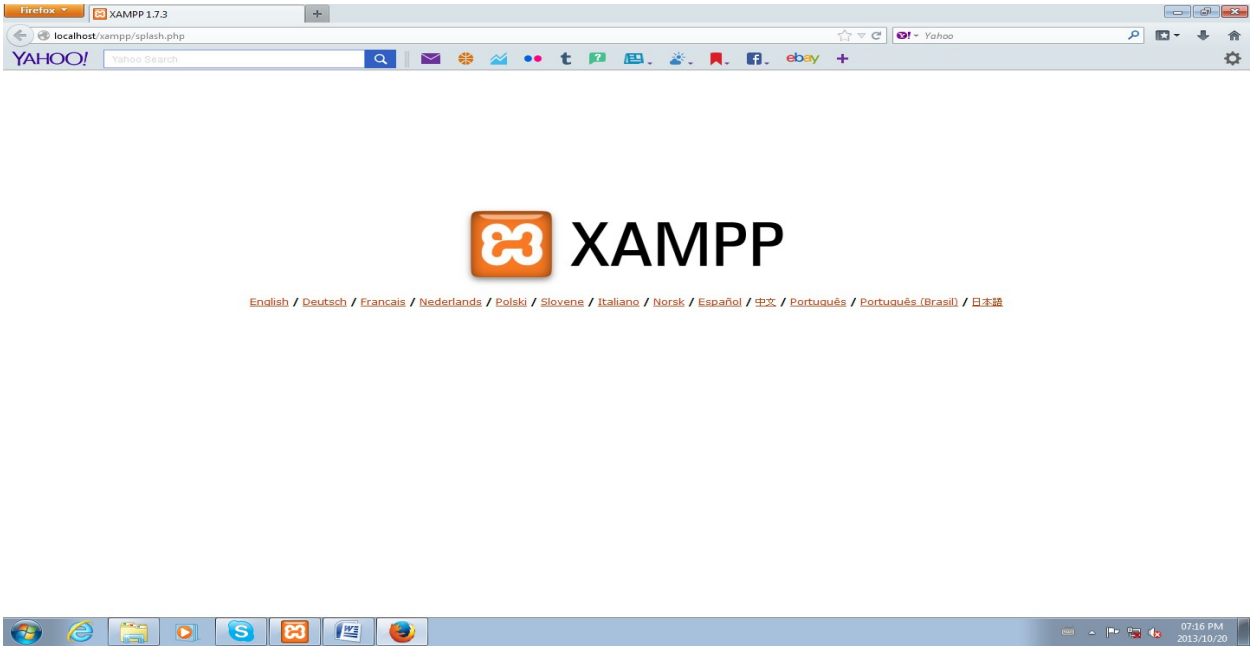
- Thereafter, I have to restart my computer so that the changes to Skype take effect.

Figure 23: *Restarting Xampp*



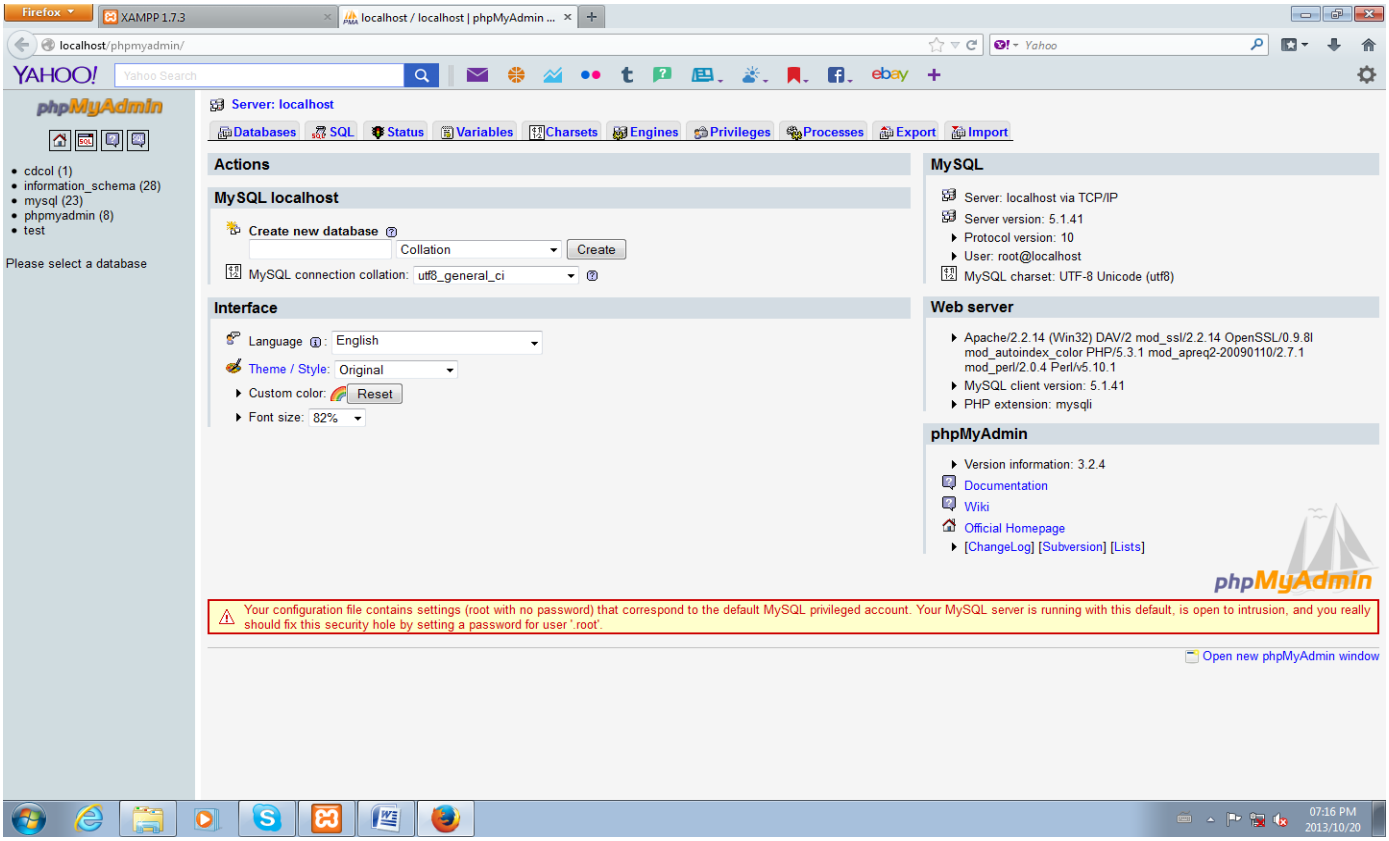
- You can see that it has worked. Apache is working.
- To test if Apache and Mysql are running well, click on the **Admin Icon** for Apache first; you will see this window opening in your browser.

Figure 24: *Test run Xampp*



- The above shows that Apache, your server software is working. A server is a computer program that provides services to other computer programs (and their users) in the same or other computers. It is on the server that MySQL is installed on and other computers can access it if they are networked.
- The next thing, is to click on Admin Icon for MySQL, you will see the below screen. If Apache is off, MySQL cannot run. Apache is the base software upon which you install MySQL database management system.

Figure 25: *Test run MySQL*



The above show sthat our MySql is also working well. MySql is a database management software. It is where we will create our database. You can test run other software but if thes others are not running don't worry. Our interest is to ensure that Apche and MySQL are working.

Summary

You have learnt in unit five that My SQL database management system is one of the widely used database management system in the world. It was started in 1995 by Swedish company and later bought Oracle; one of the IBM product lines. Many organizations have however adopted MySQL technology. We also learnt that MySQL released through two licenses; namely GPL as a Free Open Source Software and as a proprietary software. We have discovered that MySQL database management system works on a basic machine with at least 512 Mbs of RAM and works on all operating systems such Windows and Linux. It was also learnt that MySQL is a server based database management system; therefore, you need to install it on a server and Apache is one of the best free server software that works well with MySQL. For the quick and easy installation of MySQL, we learnt that we need to download software called Xampp which has MySQL, Apache and other software. The installation of MySQL on Windows Systems is easy. One need just to double click the download Xampp and it will begin the installation. We further learnt that it is advisable to install Xampp in the default drive C. Once the installation is finished, you need to start Xampp by double clicking the short cut of the desktop then clik ADMIn icon of Apache to start it and test if it is running. If it fails to run, it means that there is another program on your computer which is using port number 80 because Apache server uses this port. Disable that software using that port number. You need also to click on ADMIN icon on MySQL database and see if it is running. If all the major two programs are running, it means your installation was successful and you are ready to begin practical database design.

Activities

a) Download and install Xampp
b) Test run Apache and MySQL.

UNIT 6: Database design with MYSQL database management system

Introduction

In unit six, you do practical database designing; which will involve transforming the Entity – Relationship Models into a database. The three entities that you defined in unit three will be turned into database tables. It is in this field you will learn about types of data for your entity attributes (fields). You will also use in this unit the major SQL command categories discussed in unit four. This module will also explain to you how to back up and restore your database.

Aims

This unit aims at transforming the data models (Entity-Relationship Models) into a real data base. It also seeks to explain how backing up and restoration of MySQl database is done

Objectives

At the end of this unit, you should be able to do the following:

- Create database using SQL in command line
- Create and populate the database tables using SQL statements
- Know how to Query the database tables using SQL statements
- Update fields and data using SQL statements
- Know how to backup and restore MySQL database

Equipment needed

This unit too is practical; you therefore need a computer. You may not need internet access.

Other Resources

You need books and information resources that discuss SQL and MySQL database management systems. These books include the listed below:

Beighley. L and Morrison. M (2009). *Head First PHP &MySQL*. Cambridge: O'Reilly.

Chao.L (2006). *Database Development and Management*. Boca Raton: Taylor &Francis Group.

Hoffer, J.A, Ramesh. A and Topi. H (2011). *Modern Database Management*. New Delhi; Chennai: Pearson; Dorling Kindersley

Kroenke, D.M (2005). *Database Processing: Fundamentals, Design, and Implementation*. 10th Ed. Beijing: Publishing House of Electronics Industry

Kifer. M, Bernstein.A and Lews, M.P (2005). *Database Systems: An Application-Orientation Approach*. 2rd Ed. New York: Pearson Education

Narang, R (2009). *Database Management Systems*. New Delhi: PHI Learning

Stephens. R, Plew. R and Jones, Arie. D (2012). *Sams Teach Yourself SQL*.4th Ed. New Delhi: Pearson; Dorling Kindersley

Time Frame

You need **Three** weeks to finish studying this unit.

6.1 Recap on the Unit three assignment

In unit three assignment, you drew the Entity- Relationship Model for three entities; namely **library_user, library_material** and **circulation**. You also needed to show even their attributes fields and identify some of the attributes as primary key. The following could be the attributes or fields for the above entities:

LIBRARY_USER

*user_ID(**Primary Key**)*
first_name
last_name
phone_number
e-mail_address
residential_address

LIBRARY_MATERIAL

*barcode_number (**Primary Key**)*
author
title
call_number
edition
place_of _publication
publisher
year_of _publication
series
pagination

ISBN
Terms_of-availability

CIRCULATION
*Transaction_number (**Primary Keys**)*
user_id (foreign key)
barcode_number(foreign key)
loan period
date_borrowed
date_ due

The above listed fields are the possible attributes. You can add other attributes which we have omitted.

6.2 Data types

Data that is stored in the row of each field or column differs. Others are numbers while others are characters or text. There are a number of data types; these include: **character, numerical, date, time, date time, enumeration** and **objects.**

a) *Character*

This is data which is in the form of strings (a sequence of characters). This is text data in most cases. This type of data is broken into **fixed (char)** and **variable character (varchar)** data. Fixed means that characters are fixed and the row will only allow you to keep the amount of characters you specified when you were defining the tables. If you say; *first_name char (20),* you are implying that for the first_name only 20 characters of the name should be stored. If the name is longer than 20 characters, other characters of the name will be truncated. You can use **char** to keep data on fields such as *user_id* because the length of the field is already predetermined. Variable character is used to keep data on the fields that differs in length. Fields such as names, addresses and title are best defined as variable characters as they will differ in length. The system will just use the need space to keep the field values without leaving space. You can say: *first_name* varchar (100). Suppose your name has only 50 characters, what happens to the other 50 character space? MySQL will not waste that space instead it will just create a field row of 50 characters. That is the reason varchar is preferred to char because it does not waste storage space. Many data in tables are kept as characters (either char or varchar). Data which is kept as character include attributes such as first_name, title, and author. Phone numbers are also better stored as character because they are not used in numerical operations.

b) Numerical

This is data which consist of **whole numbers** and **integers.** Whole numbers include :0,1,2,3,4---- while integers could include the following numbers-4,-3,-2,-1,0,1,2,3,4---.

It includes also decimal numbers (points). Numerical data also include numbers with decimal points. Examples of numbers with decimal points include the following: 23.6, 3.6, and 98.3. You will keep data such as id, scores etc as numerals. For instance, *user_id* could be kept as an integer while *student _percentage* as decimal.

Table 10: *Comprehensive list of data type in MySQL*

Numerical data	
TINYINT( )	-128 to 127 normal 0 to 255 UNSIGNED.
SMALLINT( )	-32768 to 32767 normal 0 to 65535 UNSIGNED.
MEDIUMINT( )	-8388608 to 8388607 normal 0 to 16777215 UNSIGNED.
INT( )	-2147483648 to 2147483647 normal 0 to 4294967295 UNSIGNED.
BIGINT( )	-9223372036854775808 to 9223372036854775807 normal 0 to 18446744073709551615 UNSIGNED.
FLOAT	A small approximate number with a floating decimal point.
DOUBLE( )	A large number with a floating decimal point.
DECIMAL( )	A DOUBLE stored as a string, allowing for a fixed decimal point. Choice for storing currency values.
String/Character data	
VARCHAR( )	A variable section from 0 to 255 characters long.

TINYTEXT	A string with a maximum length of 255 characters.
TEXT	A string with a maximum length of 65535 characters.
BLOB	A string with a maximum length of 65535 characters.
MEDIUMTEXT	A string with a maximum length of 16777215 characters.
MEDIUMBLOB	A string with a maximum length of 16777215 characters.
LONGTEXT	A string with a maximum length of 4294967295 characters.
LONGBLOB	A string with a maximum length of 4294967295 characters.

Date / Time	
DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYYMMDDHHMMSS
TIME	HH:MM:SS

Other data types in MySQL	
ENUM	To store text value chosen from a list of predefined text values
SET	This is also used for storing text values chosen from a list of predefined text values. It can have multiple values.
BOOL	Synonym for TINYINT(1), used to store Boolean values
BINARY	Similar to CHAR, difference is texts are stored in binary format.
VARBINARY	Similar to VARCHAR, difference is texts are stored in binary format.

6.3. Naming of a database and tables in MySQL

In 6.1 paragraphs, we have identified three entities and their attributes. The three entities will be taken as the name of the tables. In this case, we will the following tables; *library_user, library_material* and *circulation*. What do you think could be the mane of the database in which all the three tables will be housed? I guess you have said *library***.** That is not bad. It can be called library or anything but ensure that the database name embraces all the entities which will constitute the tables. **Note** that when naming the database and tables, ensures that names are in small letters and do not contain spaces. This is the reason why I have used under scores in the middle of certain names. Do not name a table as **library user** but **library_user**. MySQL and other database systems may not accept names of tables with spaces. The same applies to fields names. Avoid spaces in the fields' names.

Having named the three tables, unit three, we will attach data types to all the identified fields as follow:

library_users

(

user_id INT (8) NOT NULL,

first_name VARCHAR (50)NOT NULL,
last_name VARCHAR (50) NOT NULL,
phone_number CHAR (13),NOT NULL
e-mail_address VARCHAR (60),
residential_address VARCHAR (100),
PRIMARY KEY (user_id)
) type=InnoDB ;

library_materials

(

barcode_number INT (8) NOT NULL,

author VARCHAR (100) NOT NULL,
title VARCHAR (200) NOT NULL,
call_number VARCHAR (10),
edition CHAR (4),
place_of _publication VARCHAR (40)
publisher VARCHAR (40),
year_of _publication CHAR (4),
series VARCHAR (50),
pagination VARCHAR (10),
ISBN VARCHAR (20),
Terms of-availability VARCHAR (40),
PRIMARY KEY (barcode_number)
) type=innoDB;

Circulation
(
Transaction_number INT (8) NOT NULL AUTO-INCREMENT,
user_id INT (8) NOT NULL,
barcode_number INT (8) NOT NULL,

```
loan period VARCHAR (6),
date_borrowed  DATE (10),
date_ due DATE (10),
PRIMARY KEY (transaction_number)
) type=InnoDB;
```

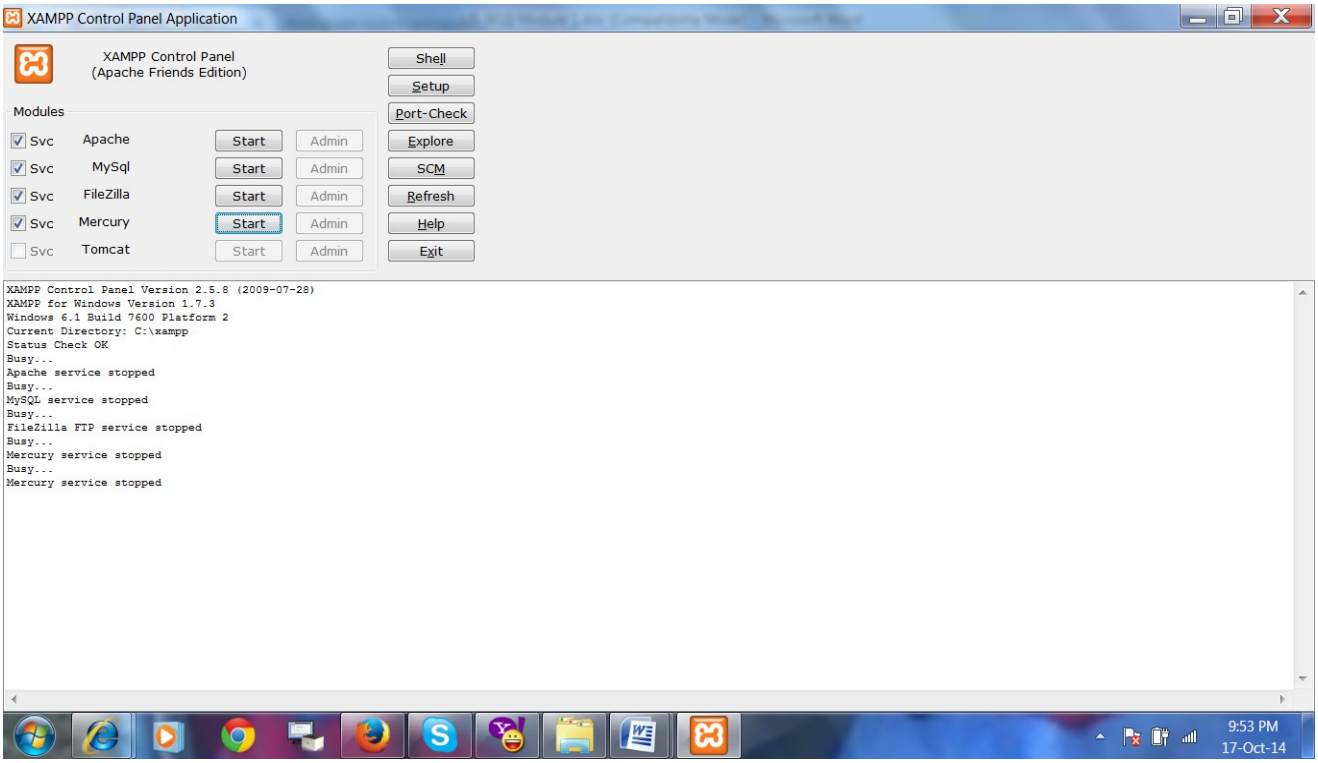
6.3.1 Explanation

In each and every table, you have seen that we are beginning with the name of the table, and then we open brackets and define the fields and their data type. When we say NOT NULL means that the field will never be without data. Some data have to be stored in it. It is also a requirement in MySQL that at the end of each field defined we put a comma. In the two tables we declared the primary keys which will act as the foreign keys in the other table. You have also seen where we have said **type=InnoDB**. This is the way we specify the database engines of MySQL to use to create and store our database. MySQL database management systems have several storage engines which include MyISAM and InnoDB .When creating a table; you need to declare the type of storage engine your table will be using. MYISAM supports Table-level Locking. It is designed for need of speed. However, it does not support foreign keys. Further, MYISAM stores its tables, data and indexes in disk space using separate three different files (tablename.FRM, tablename.MYD, tablename.MYI). It also does not support transaction. You cannot commit and rollback with MYISAM. Once you issue a command it's done. On the other hand, InnoDB supports Row-level Locking. It is designed for maximum performance when processing high volume of data. Further, it supports foreign keys. InnoDB stores its tables and indexes in a table space and it supports transaction. You can commit and rollback with InnoDB. We finally closed the fields definition with closed brackets and immediately that we define the type of MySQL engine to use and end with a semi-colon.

6.4 Connect to MySQL using command line

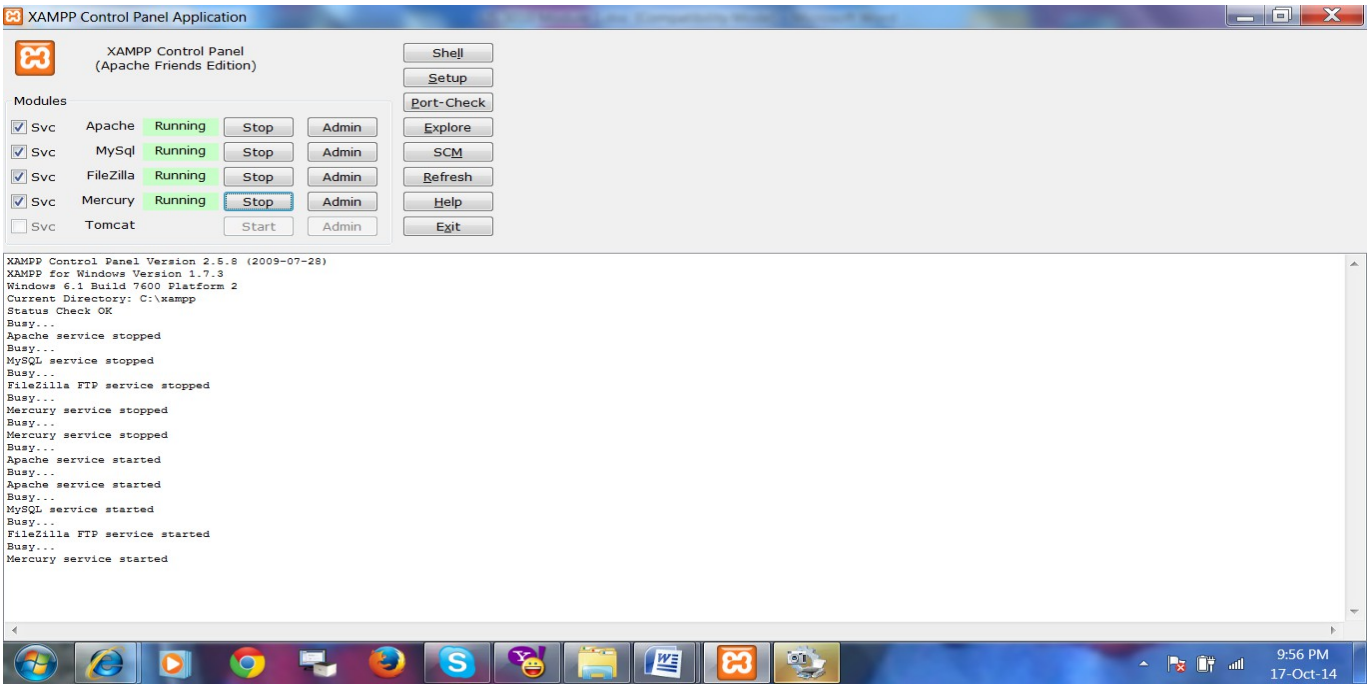
Before you connect to MySQL, you should make sure that both Apache and MySQL are turned on and running. To turn them, double click the Xampp short on the desktop, you will be greeted by the following below screen.

Figure 26: *Starting Apache and MySQL*



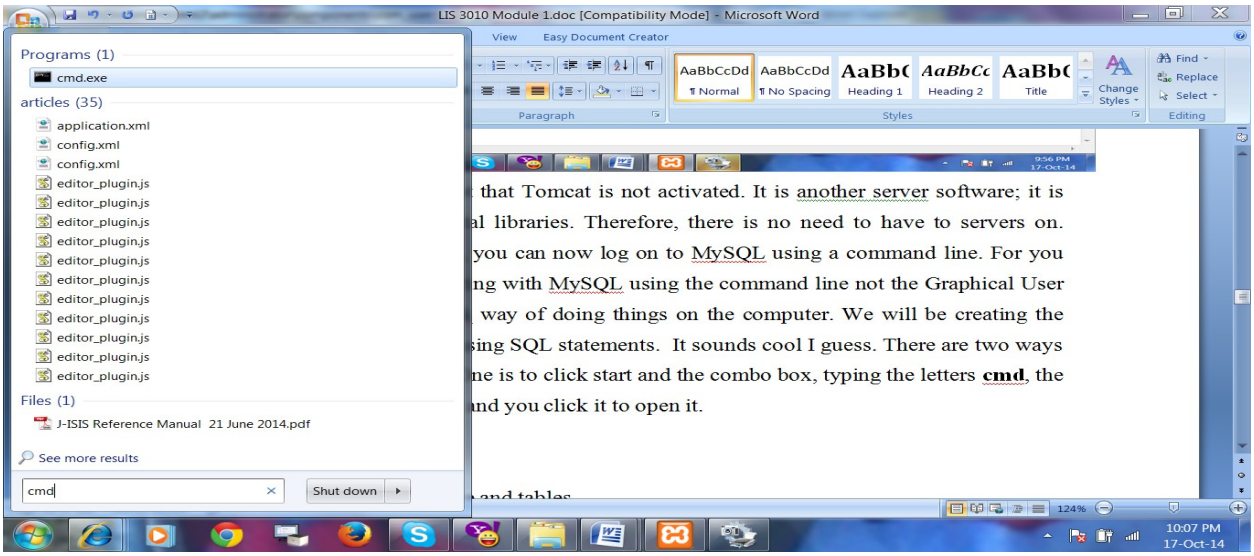
Then you should click on **start,** to start all the programs.

Figure 27: *Starting Apache and MySQL*



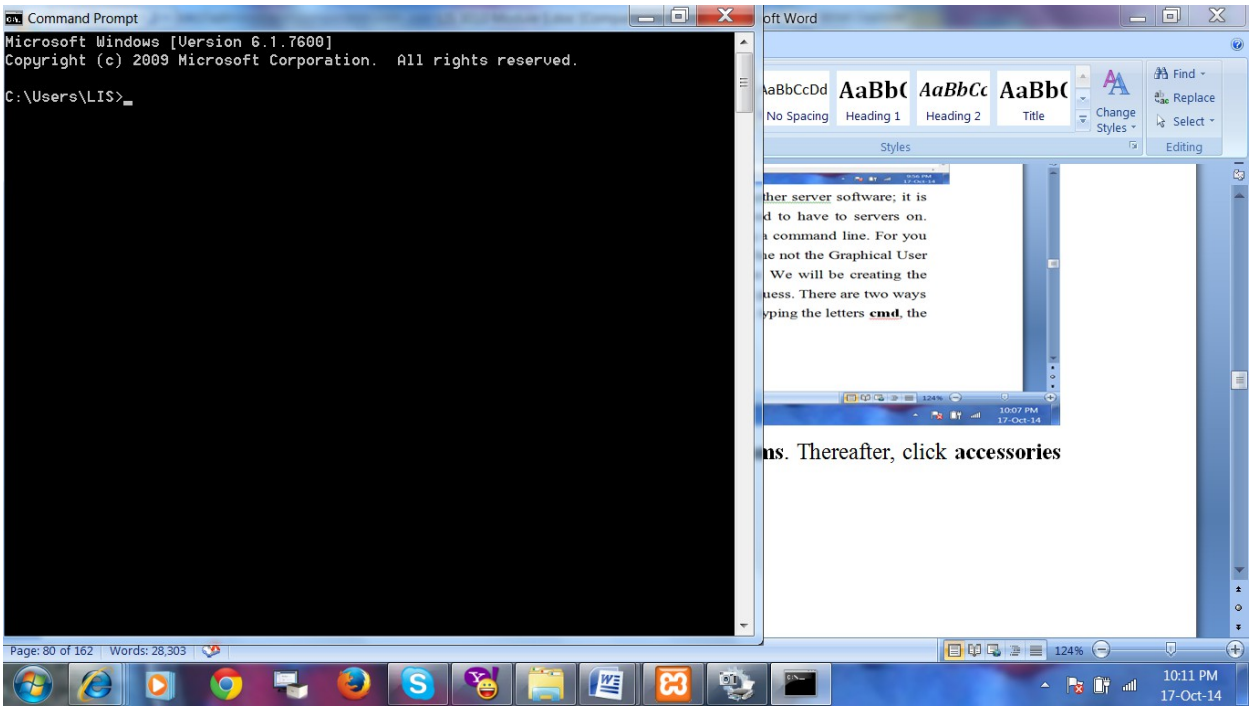
You have seen in this screen shot that Tomcat is not activated. Tomcat is server software like Apache; it is useful for the creation of digital libraries. Therefore, there is no need to have two servers on. Having turned the software on, you can now log on to MySQL using a command line. For you information, we will be interacting with MySQL using the command line not the Graphical User Interface (GUI); the click, click way of doing things on the computer. We will be creating the database in the command line using SQL statements. It sounds cool I guess. There are two ways of opening the command line. One is to click start and the combo box, typing the letters **cmd**, the command line will be retrieved and you click it to open it.

Figure 28: *Opening the terminal to access MySQL*



The other method is to click **start** icon, then click **all programs**. Thereafter, click **accessories** and then **command prompt**, the command line will open. It is a black screen. Do not get afraid of it. You will be working on that screen.

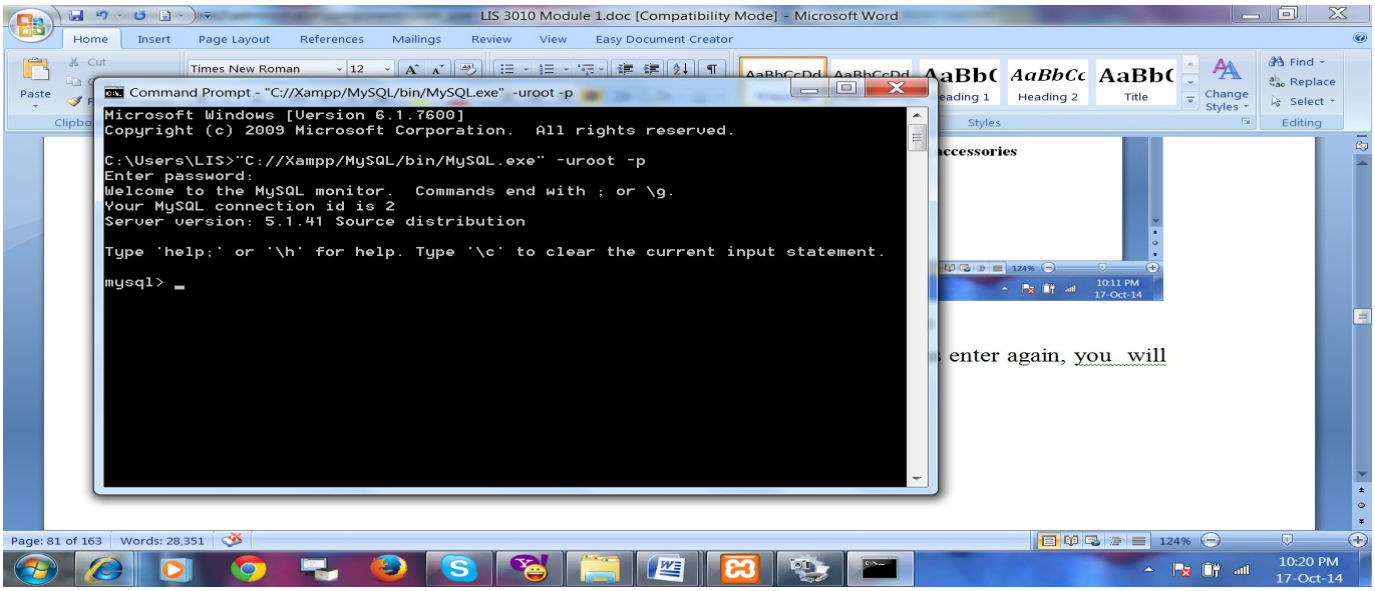
Figure 29: *Terminal*



In the black screen type "C://Xampp/MySQL/bin/MySQL.exe" –uroot –p

Then press the enter button. MySQL will ask for the password, just press enter again, you will successfully logon to MySQL.

Figure 30: *How to access MySQL in the terminal*



Make sure that between **MySQL.exe** and **–uroot** there is a space. The same applies to **–p** and **–uroot**. In addition, ensure that open and closed inverted commas (**“”**) are used as the screen shot and text.

Now, what do we mean by –uroot and –p? –u means users and root is the name of the user. This default username you will be using on MySQL until you change the username to something more secure. –p entails password. At the moment, you still do not have password. That is the reason we did not type anything after –p. And when the system asked for the password, we just pressed the enter button.

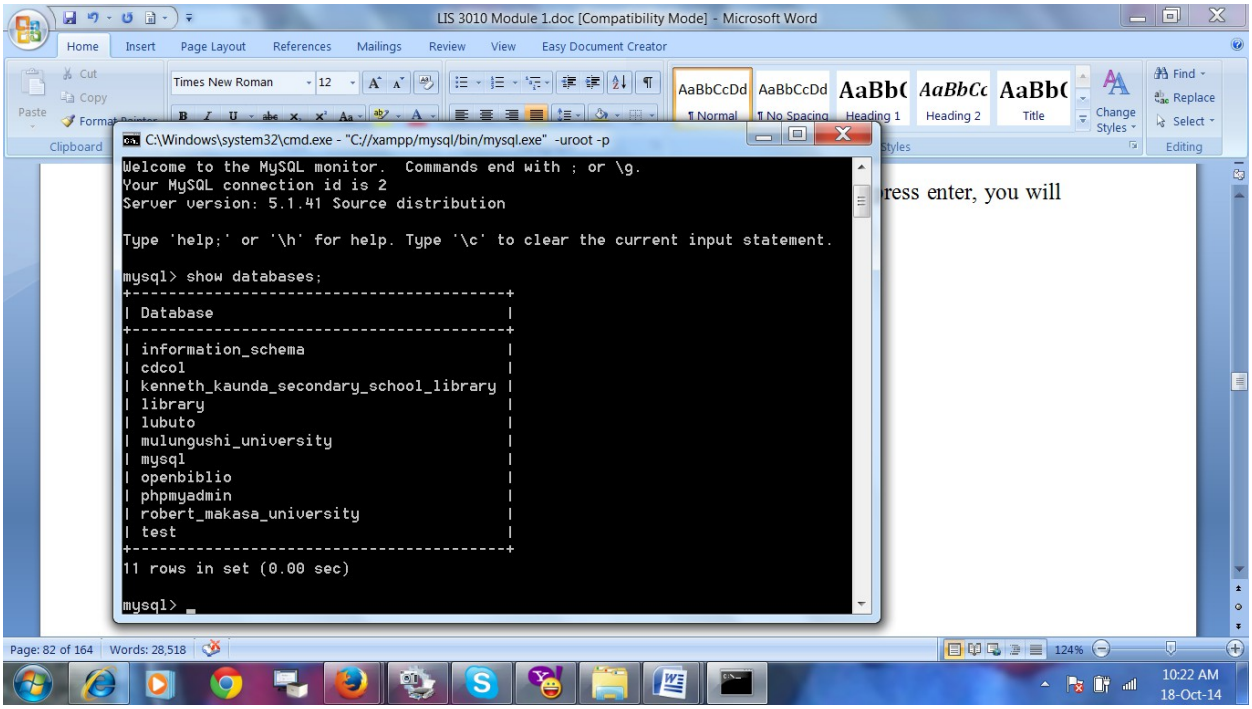
6.5 Creation of the new database and tables

Before you start creating a database, it is advisable to check the existing databases in MySQL by typing this command:

show databases;

Remember always to end SQL statements with a semi-colon (;). Once you press enter, you will be greeted by the below screen showing all the databases.

Figure 31: *Show databases*



My MySQL has many databases because I have created many of them. In your case, you may just see three or so databases. Those databases come with MySQL. Now you are ready to create your database.

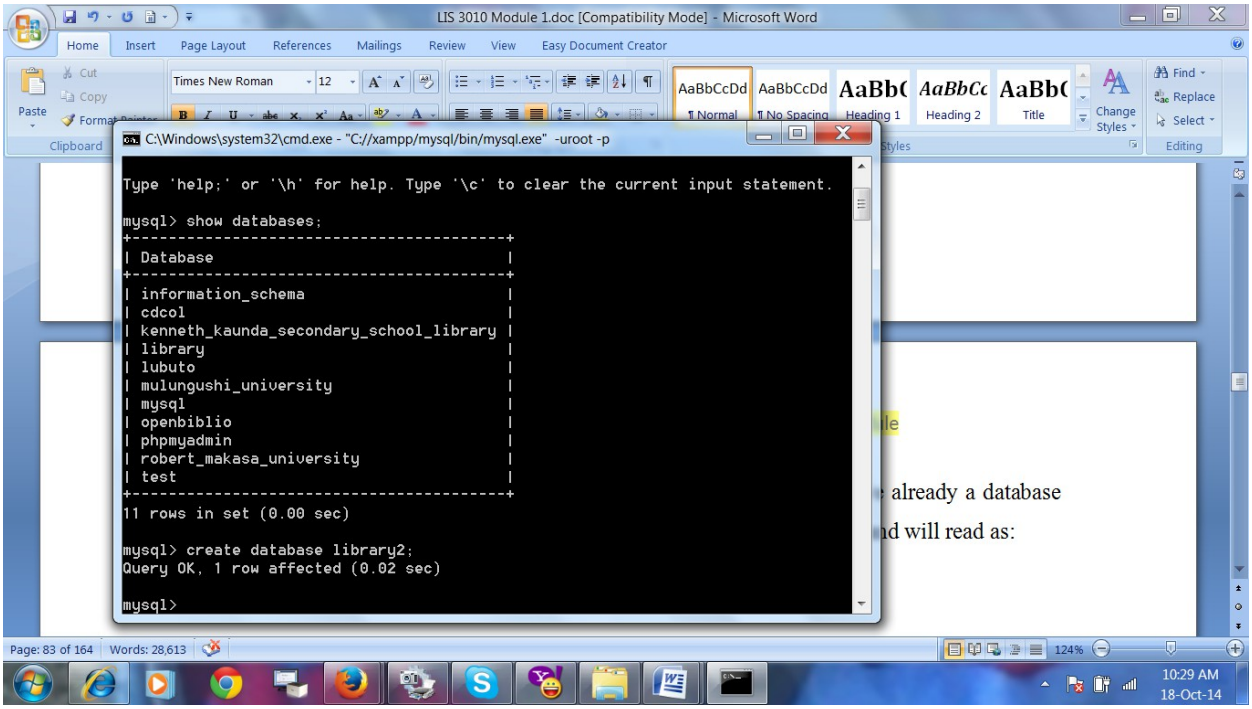
To create database, we use a **create** command and mention the name of the database to be created as follows:

create database library;

The above SQL statement will create a database called library. Since I have already a database named library, I will name the new database as **library2.** So the SQL command will read as:

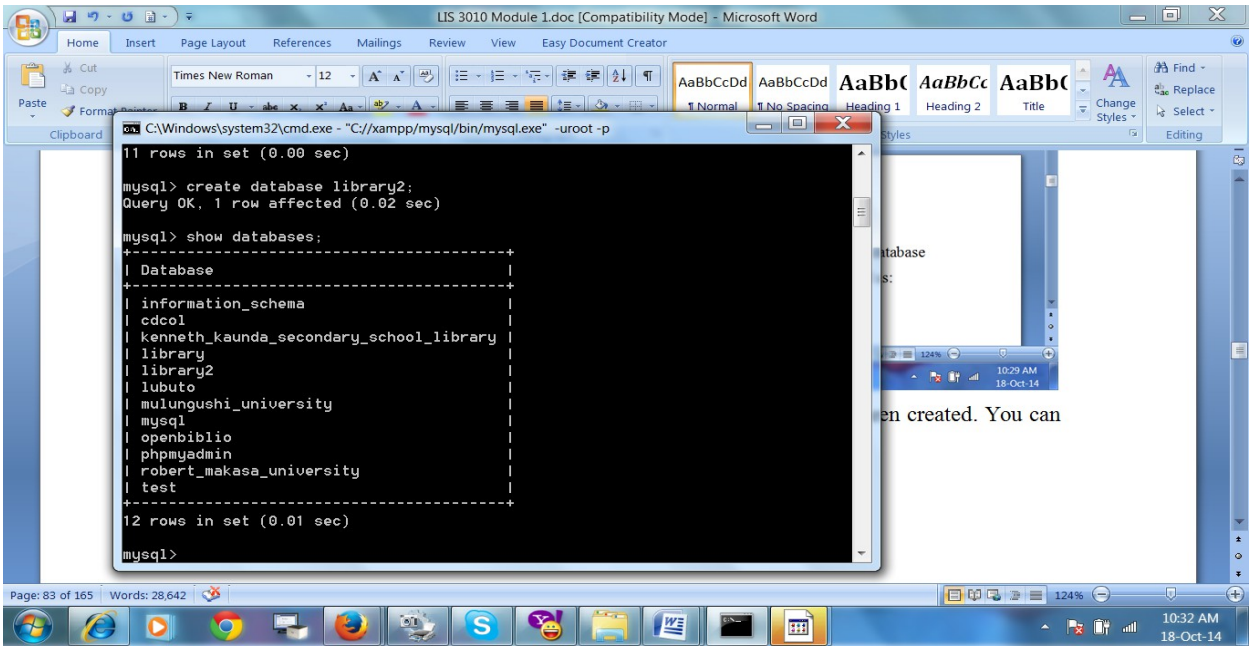
create database library2;

Figure 32*: Show databases*



As you can see above, MySQL says query OK. It means my database has been created. You can verify by sending a SQL statement to MySQL and say **show databases;**

Figure 33*: Show databases*

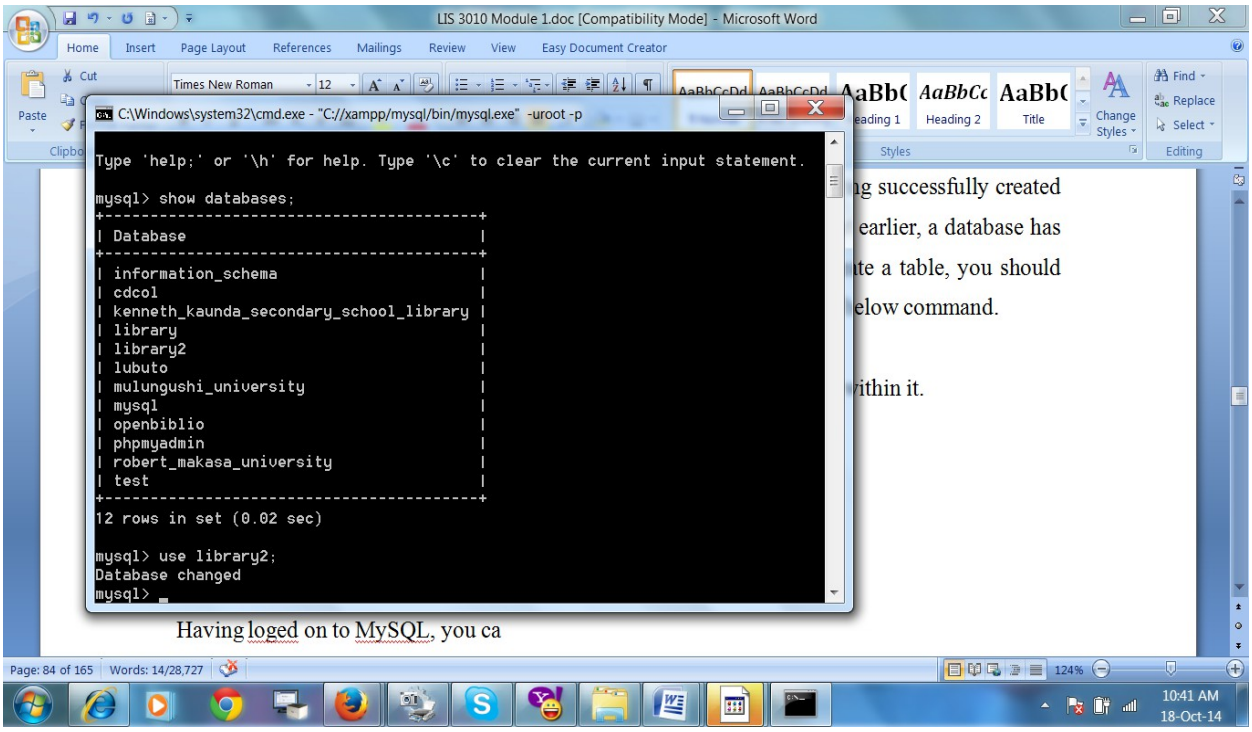


You now see **library2** being shown as one of the databases created. Having successfully created a database, you need now to create tables in your databases. As we learnt earlier, a database has tables or a table. So you need to create tables within a database. To create a table, you should first choose the database in which you want to create a table by using the below command.

use library2;

This command will select the database library2 and you can create tables within it.

Figure 34*: Show databases*



Thereafter, create the first table called **library users** as follows:

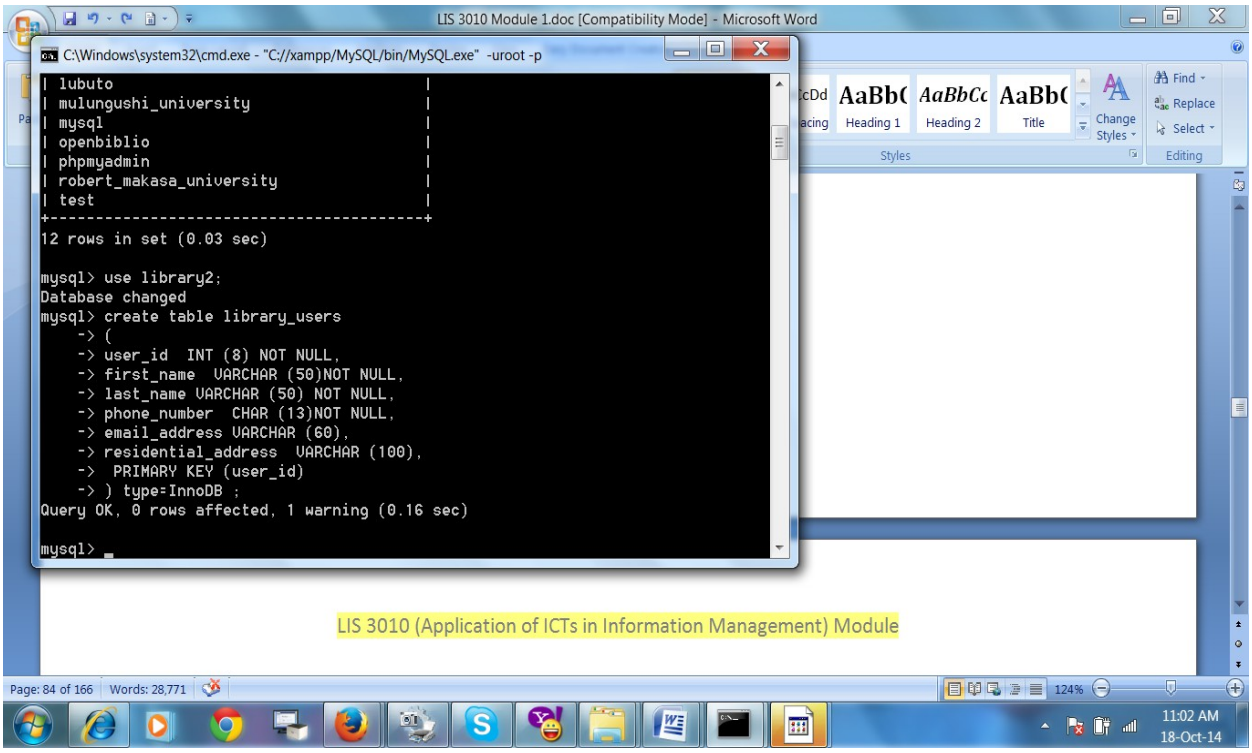
create table library_users

(

user_id INT (8) NOT NULL,

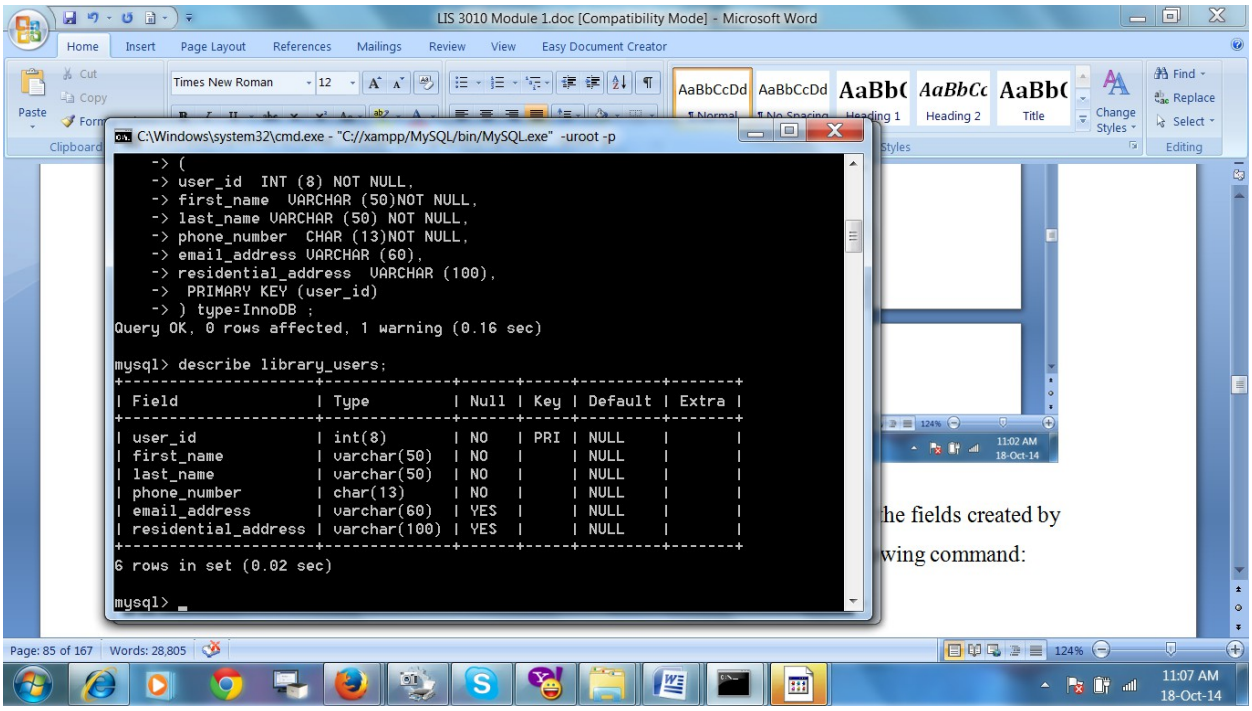
first_name VARCHAR (50)NOT NULL,
last_name VARCHAR (50) NOT NULL,
phone number CHAR (13)NOT NULL,
email_address VARCHAR (60),
residential_address VARCHAR (100),
PRIMARY KEY (user_id)
) type=InnoDB ;

Figure 35*: Creation of database*



As you can see, a table called library_users has been created. You can check the fields created by issuing a SQL statement called **describe.** Type in the command line the following command: **describe library_users;**

Figure 36*: Creation of table of a database*

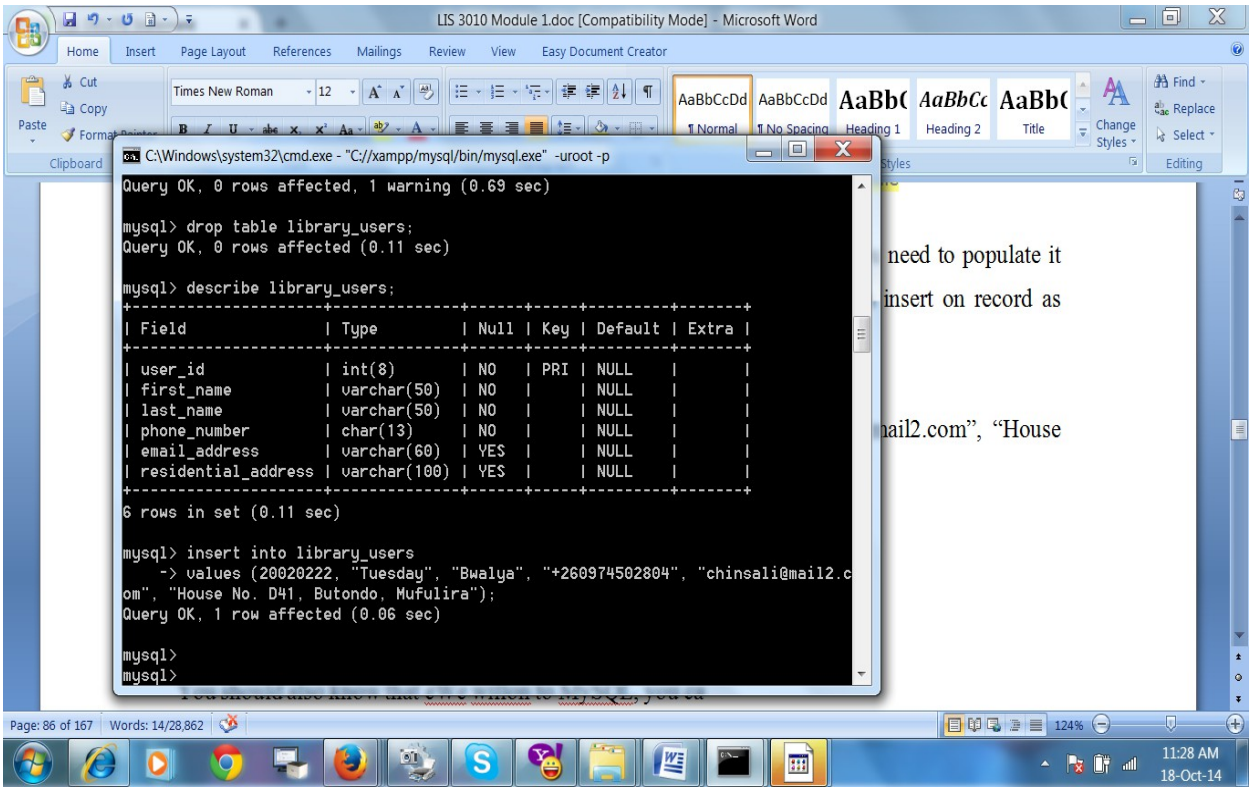


You may agree with me that the created table is empty. It has no data. You need to populate it (put data inside the table). The command we use is called **insert**. Let us insert on record as follows:

insert into library_users

values (20020222, “Tuesday”, “Bwalya”, “+260974502804”, “chinsali@mail2.com”, “House No. D41, Butondo, Mufulira”);

Figure 37*: Inserting data in the table*

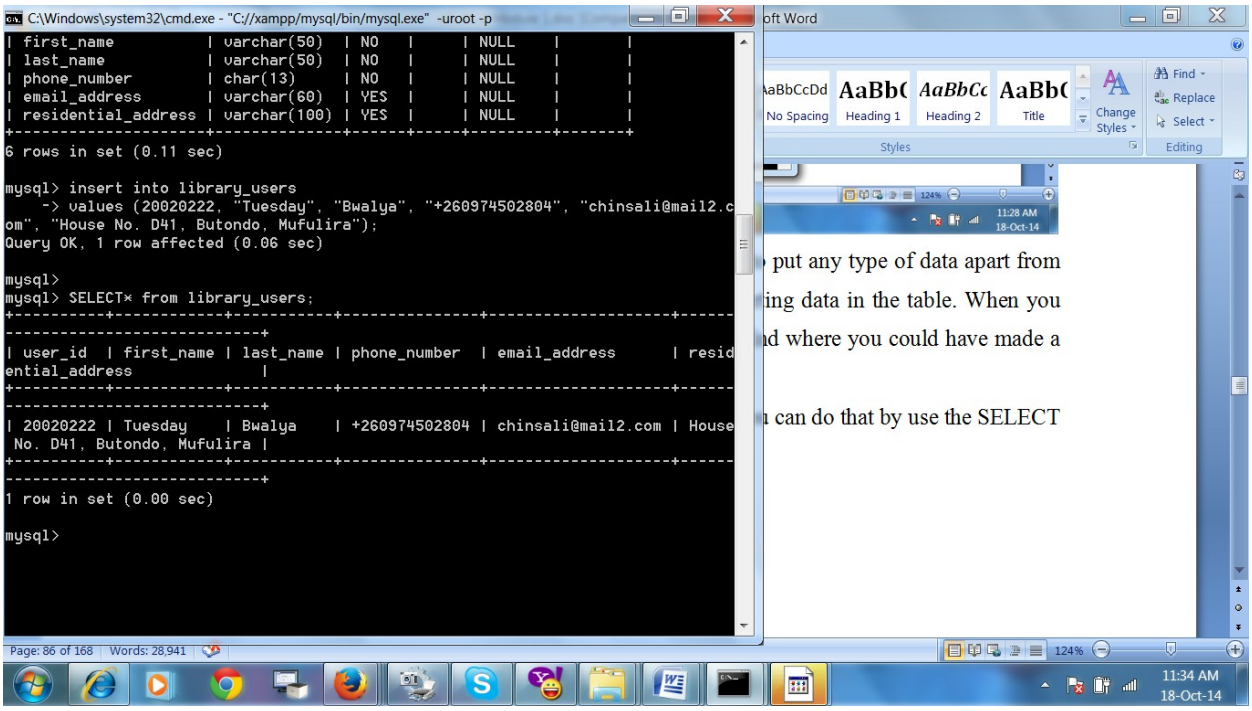


You have successfully inserted data in your table. Remember to put any type of data apart from **integers** in Open and closed inverted commas (“”) when inserting data in the table. When you see the errors, don’t faint just go through you statements and find where you could have made a mistake.

You may wish to retrieve the data that you have just entered; you can do that by use the SELECT command as follows:

SELECT * from library_users;

Figure 38: *Retrieving all data in the table*



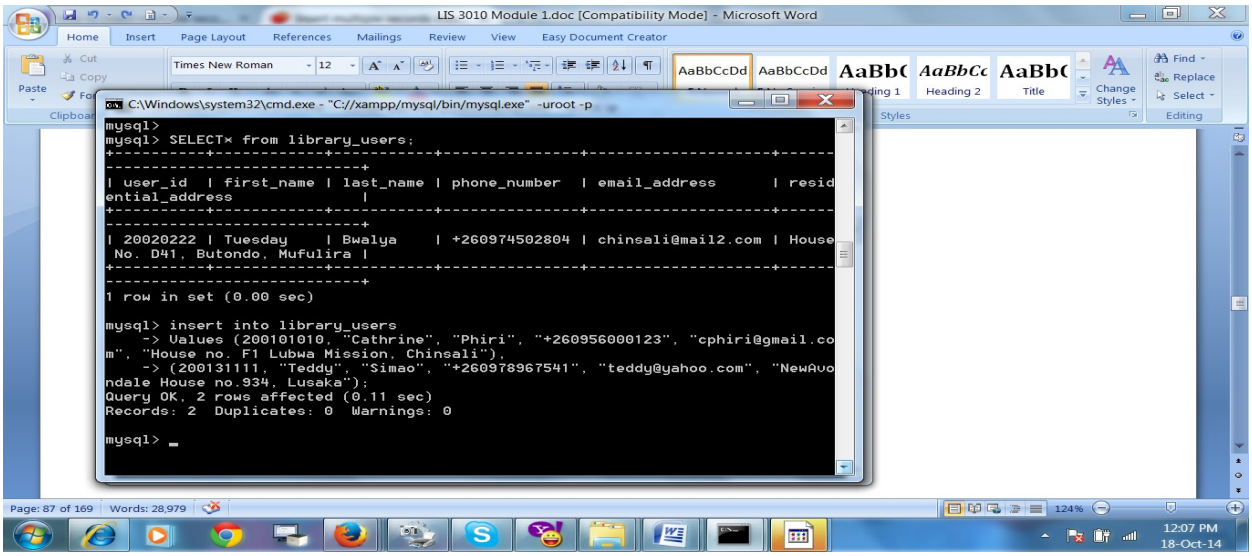
You can also enter multiple records into a table using one SQL statement as follows:

insert into library_users

Values (200101010, "Cathrine", "Phiri", "+260956000123", "cphiri@gmail.com", "House no. F1 Lubwa Mission, Chinsali"),

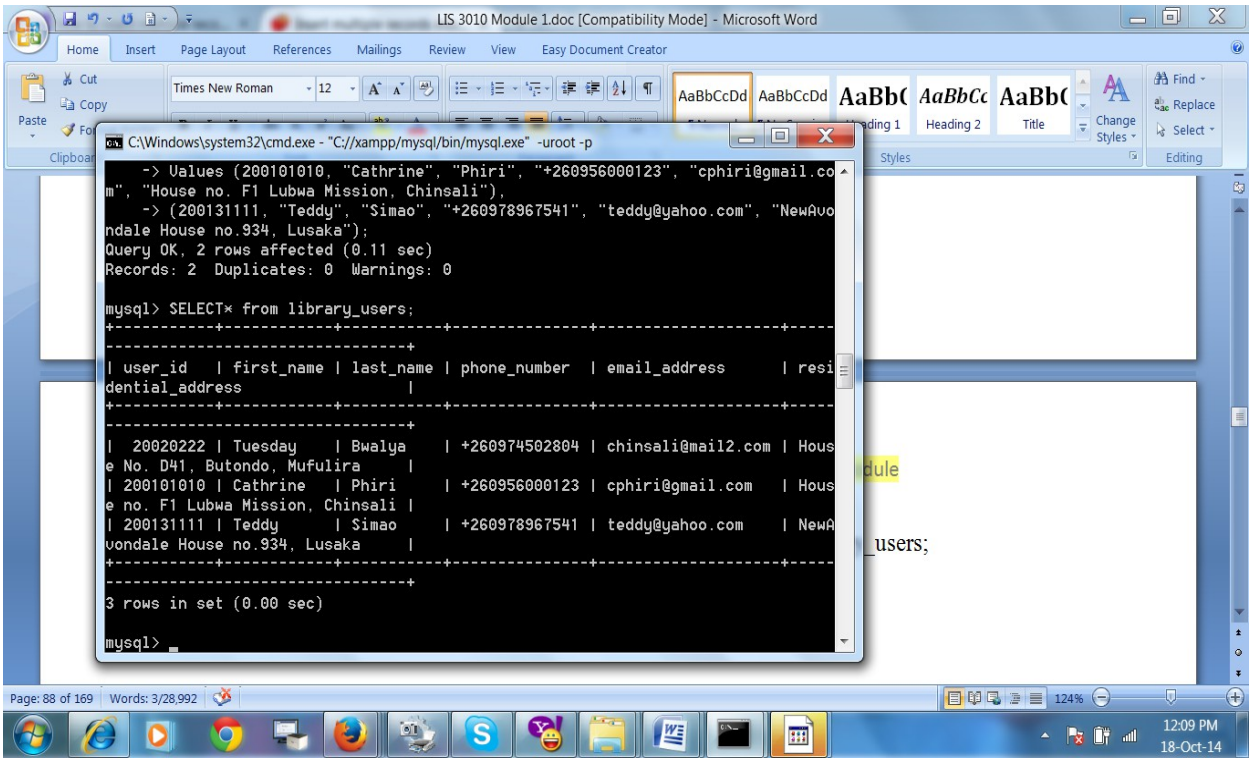
(200131111, "Teddy", "Simao", "+260978967541", "teddy@yahoo.com", "NewAvondale House no.934, Lusaka");

Figure 39: *Inserting multiple data in the table*



You can check the records entered by typing again: SELECT* from library_users;

Figure 40: *Retrieving all data from the table*



My experience with MYSQL has shown that in same versions of MySQL, the double quotes (“”) will not work instead of the single quotes (‘’). You should always also put values of each record in brackets () and separate it from other records with a comma (), if you want to enter many records with a single SQL statement as I have shown above.

You can proceed now to create the other remaining two tables using the above example.

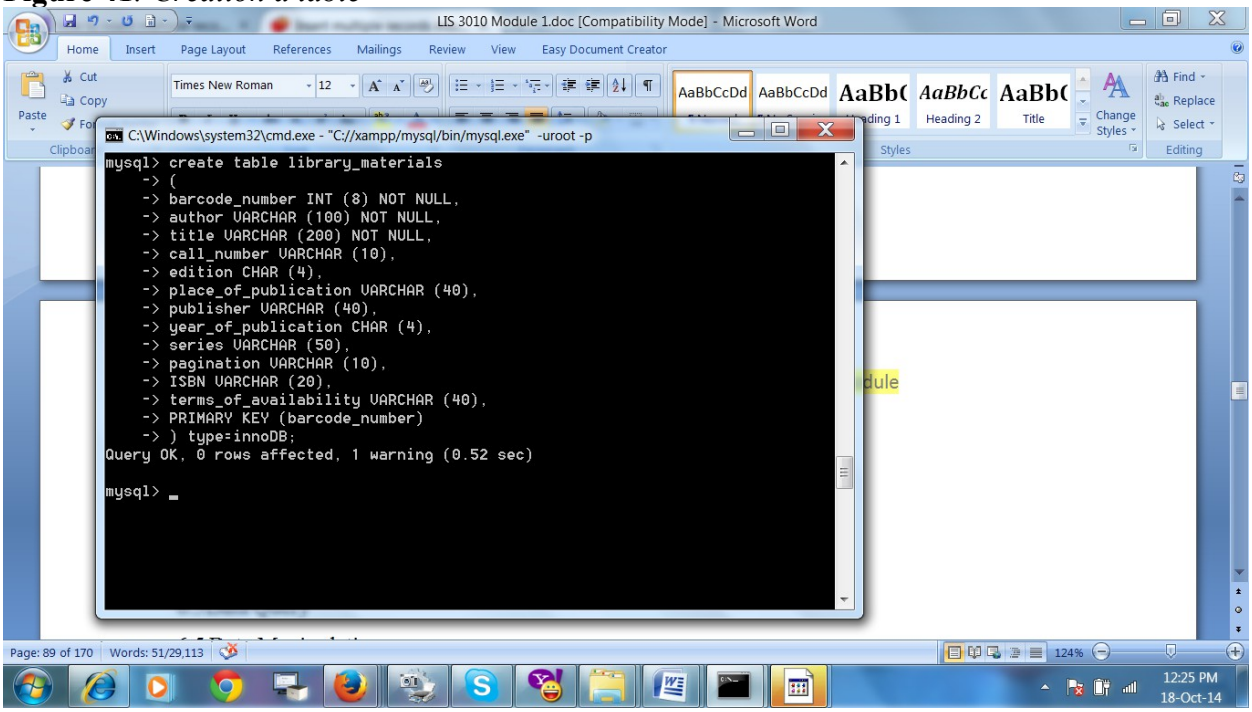
create table library_materials

(

barcode_number INT (8) NOT NULL,

author VARCHAR (100) NOT NULL,
title VARCHAR (200) NOT NULL,
call_number VARCHAR (10),
edition CHAR (4),
place_of_publication VARCHAR (40),
publisher VARCHAR (40),
year_of_publication CHAR (4),
series VARCHAR (50),
pagination VARCHAR (10),
ISBN VARCHAR (20),
terms_of_availability VARCHAR (40),
PRIMARY KEY (barcode_number)
) type=innoDB;

Figure 41*: Creation a table*

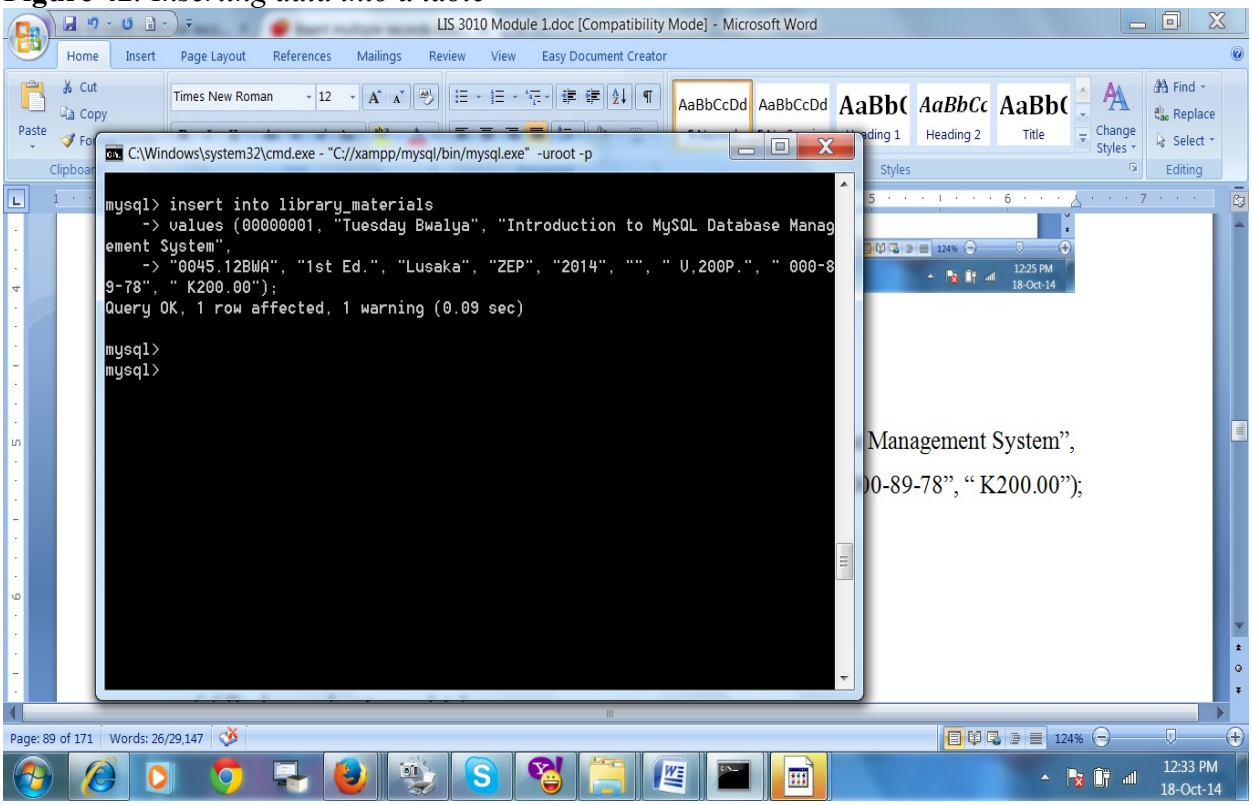


We can now enter one book into it as follows:

insert into library_materials

values (00000001, “Tuesday Bwalya”, “Introduction to MySQL Database Management System”, “0045.12BWA”, “1st Ed.”, “Lusaka”, “ZEP”, “2014”, “”, “ V,200P.”, “ 000-89-78”, “ K200.00”);

Figure 42: *Inserting data into a table*

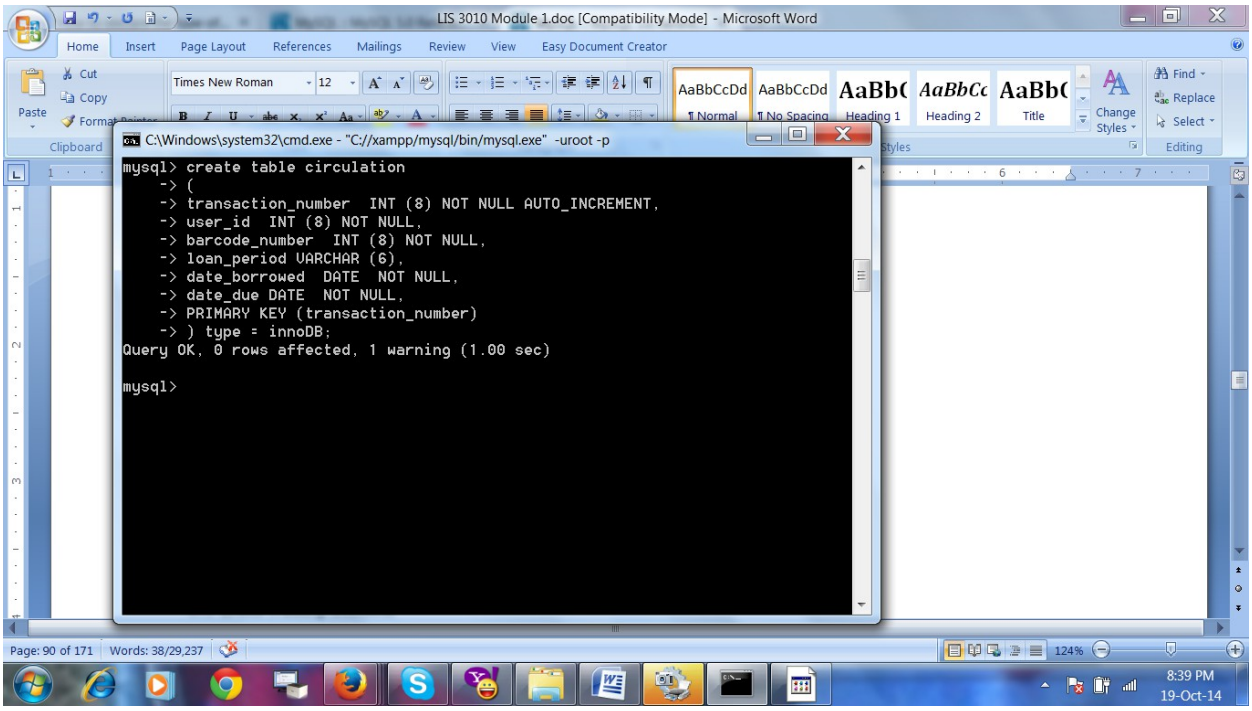


You can see that the field for series was left blank as this book has no series. To leave a field blank, just type the double or single quotes like I have done in the above example.

Create now the last table which is called circulation. The SQL commands are shown below.

```
create table circulation
(
transaction_number  INT (8) NOT NULL AUTO_INCREMENT,
user_id  INT (8) NOT NULL,
barcode_number  INT (8) NOT NULL,
loan_period VARCHAR (6),
date_borrowed  DATE  NOT NULL,
date_due DATE  NOT NULL,
PRIMARY KEY (transaction_number)
) type = innoDB;
```

Figure 43: *Creation of a table*



You can now populate the table as follows:

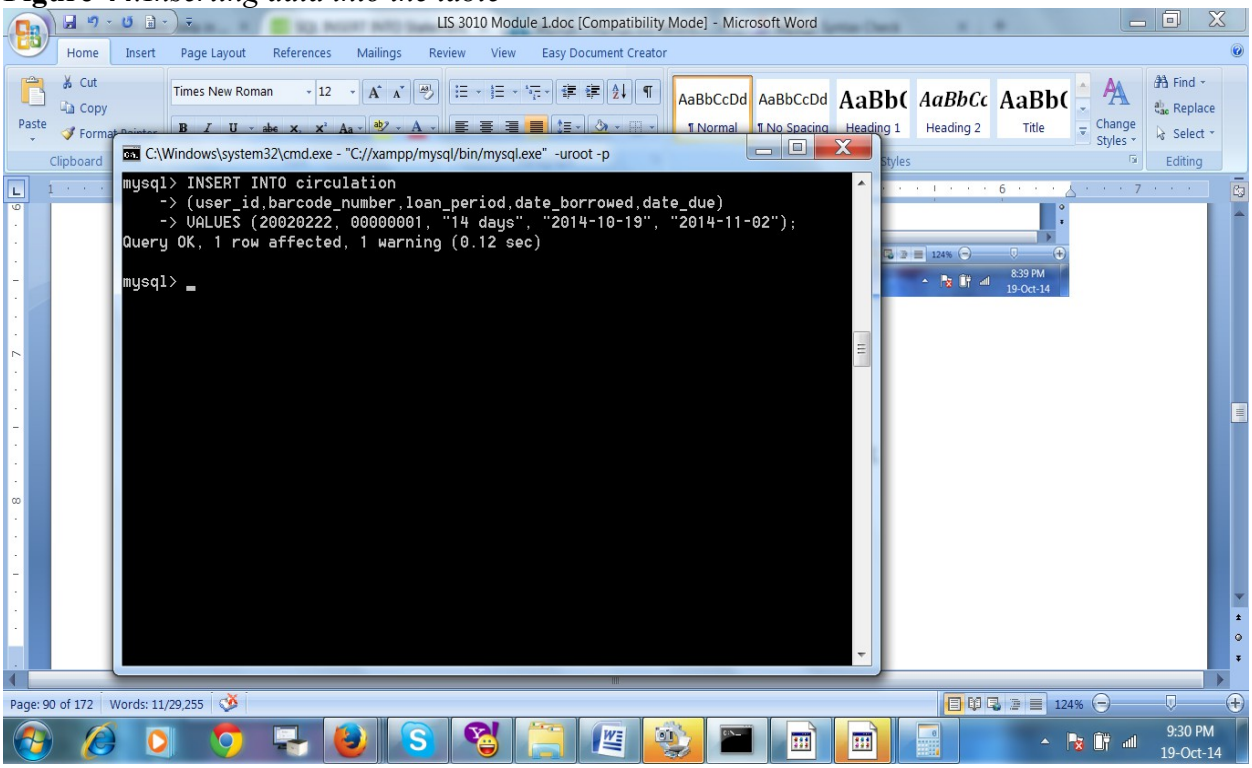
INSERT INTO circulation

(user_id,barcode_number,loan_period,date_borrowed,date_due)

VALUES (20020222, 00000001, “14 days”, “2014-10-19”, “2014-11-02”);

You are wondering why we have changed the way we have been inserting data into the tables this time round. The reason is that transaction_number field is an auto field; this means that MySQL will be generating an auto number for each transaction entered. This implies that you need to specify the fields you want to enter data. Look at the way I did it. I have put the columns or fields in which I want to enter data in brackets and separate the fields with commas. Ensure you do not leave spaces between the columns or fields names or else you will see errors.

Figure 44:*Inserting data into the table*



6.5. Data query

Since you have created three tables in the database called library2, we will now learn how to retrieve data using the SELECT command. As usual, the starting point if your computer and your server were off; you need to start your xampp. Therefore after, start Apache and MySQL. Then logon to MySQL and start transacting. Begin by issuing a SHOW databases command as follows:

SHOW databases;

This command will display all the databases in MySQL and thereafter, issue to MySQL a USE (name of database) command to select the database to transact from as follows:

USE library2;

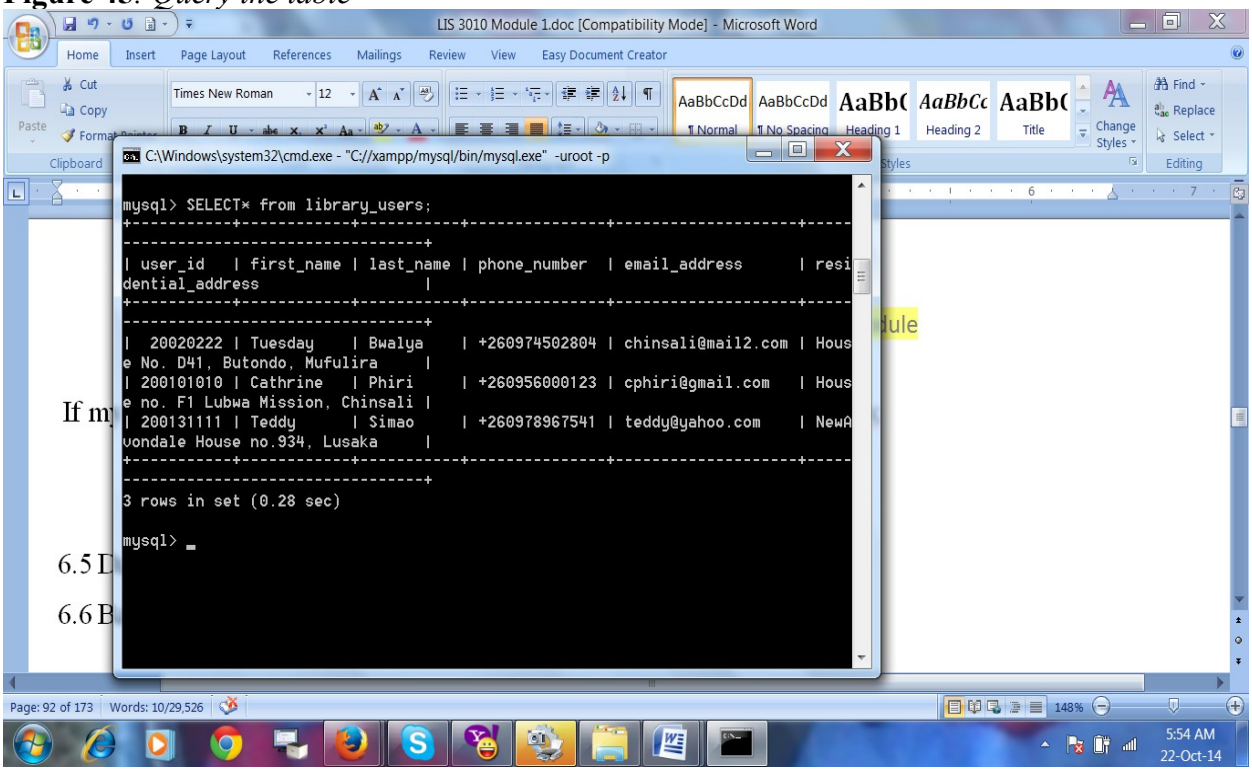
You will remember that **library2** is the name of my database and don't know yours.

This time, we will be retrieving data from our three tables. As we have already seen, if you want to retrieve data on all the fields in a table, we use the below SQL command:

SELECT* from (name of the table);

If my table is **library_users** , I will then say: SELECT* from library_users;

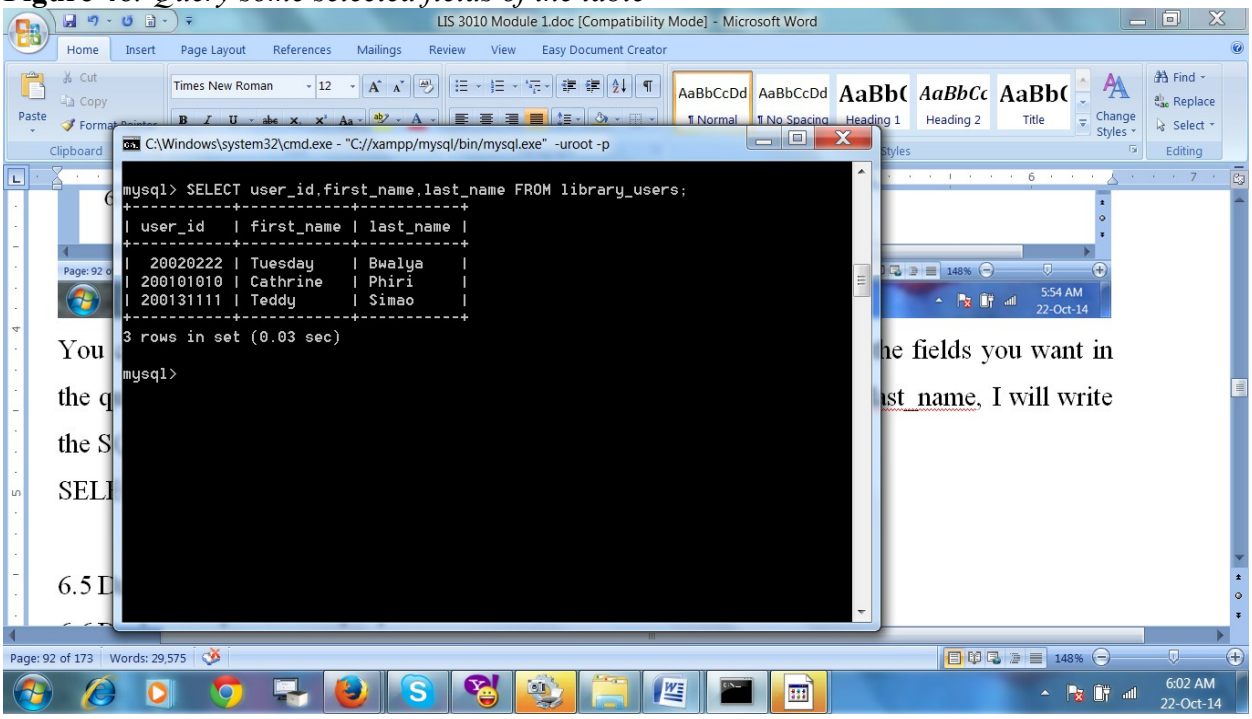
Figure 45: *Query the table*



You can also retrieve data on selected fields; not all the fields by specifying the fields you want to query. Suppose we just want to retrieve data on user_id , first_name and last_name, I will write the SQL query as shown below:

SELECT user_id, first_name,last_name FROM library_users;

Figure 46: *Query some selected fields of the table*

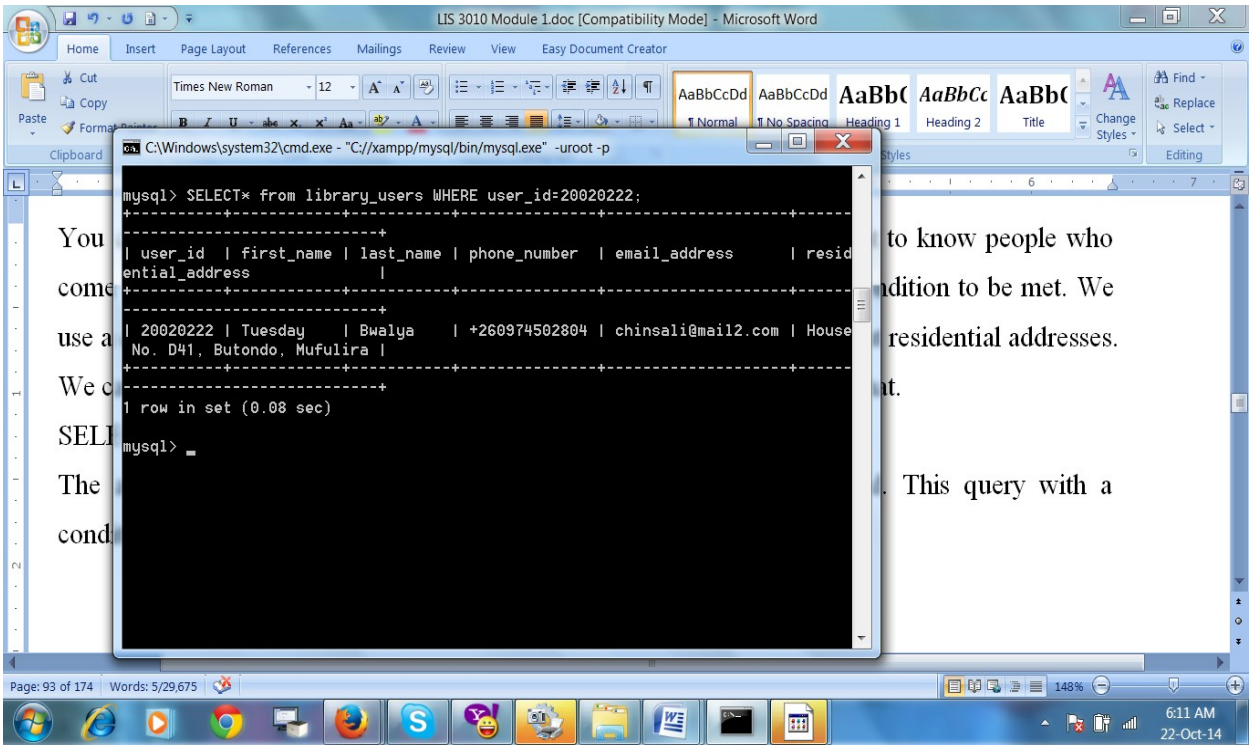


You can also retrieve data depending on the condition. Let say, you want to know people who come from a certain area let say Mufulira. You will use the town as the condition to be met. We use a condition called WHERE. In my tables, there is no field for towns but residential addresses. We cannot use town to retrieve data but I will use user_id to demonstrate that.

SELECT* from library_users WHERE user_id=20020222;

The above query will simply retrieve data on a user with the above id. This query with a condition is important when you want to filter data.

Figure 47: *Query one record from the table*



What happens when you send a query with a condition, MySQL will match the **id** with all the id numbers in the table and only retrieves the records which has that id prescribed in the query. If there is no user with such an id, you will not see results. Read more about WHERE queries. They are many and their usage depends on the circumstances.

We can also retrieve data from many tables at the same time. Let say, a user has borrowed a book, you can get data from the user table, library materials table and circulation using the JOIN QUERY.

SELECT* from library_users

JOIN circulation

ON library_users.user_id=circulation.user_id

JOIN library_materials

On circulation.barcode_number=library_materials.barcode_number;

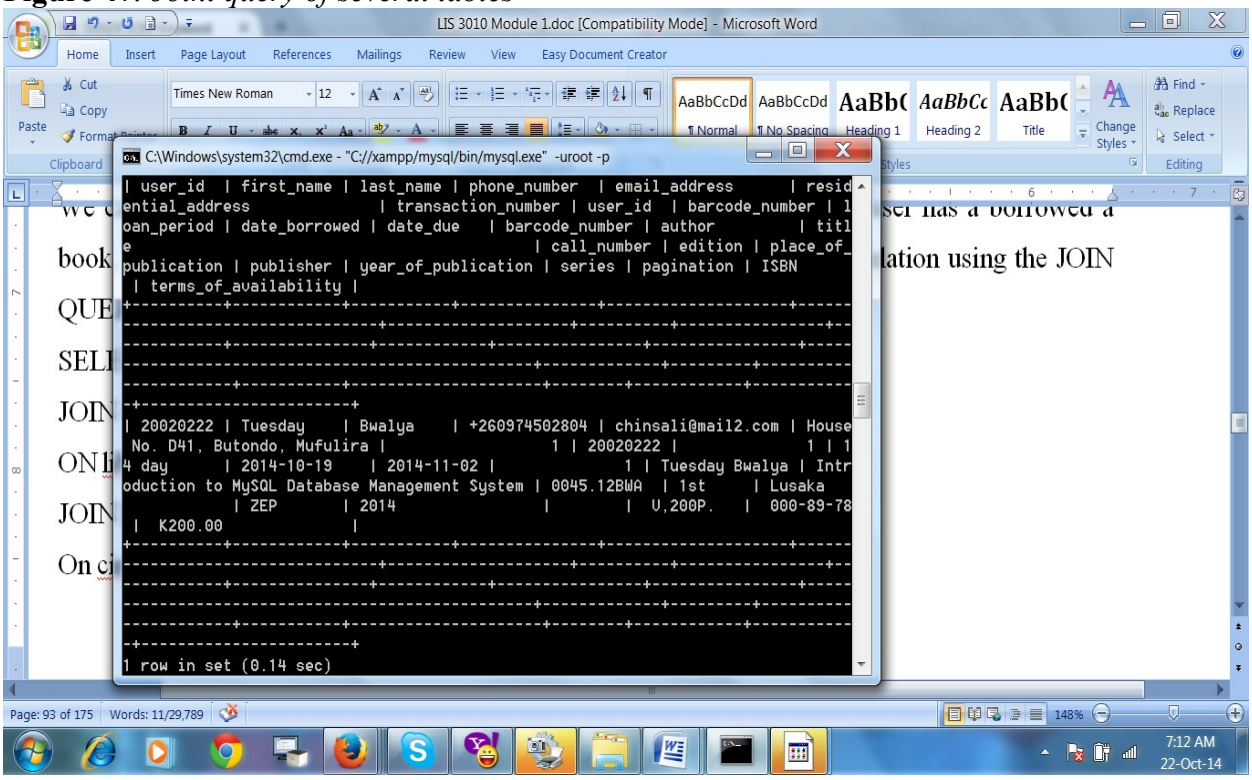
In the above query, you see that we are joining the three tables on the basis on primary and foreign keys. For instance, when we say:

JOIN circulation **ON library_users.user_id=circulation.user_id**. We are simply telling MySQL to join the library_user table and circulation table on the basis of the common field

called *user_id*. This user_id is a primary key for the library_users table but it is a foreign key in the circulation table. The same principle is applied when we say:

JOIN library_materials ON circulation.barcode_number=library_materials.barcode_number.

Figure 47*: Joint query of several tables*



If you are accurately typed the query, you see the above results. You can also retrieve data just from two tables. Let say library_users and circulation. Always remember that for you to retrieve data from more than one table, these tables should share a common field.

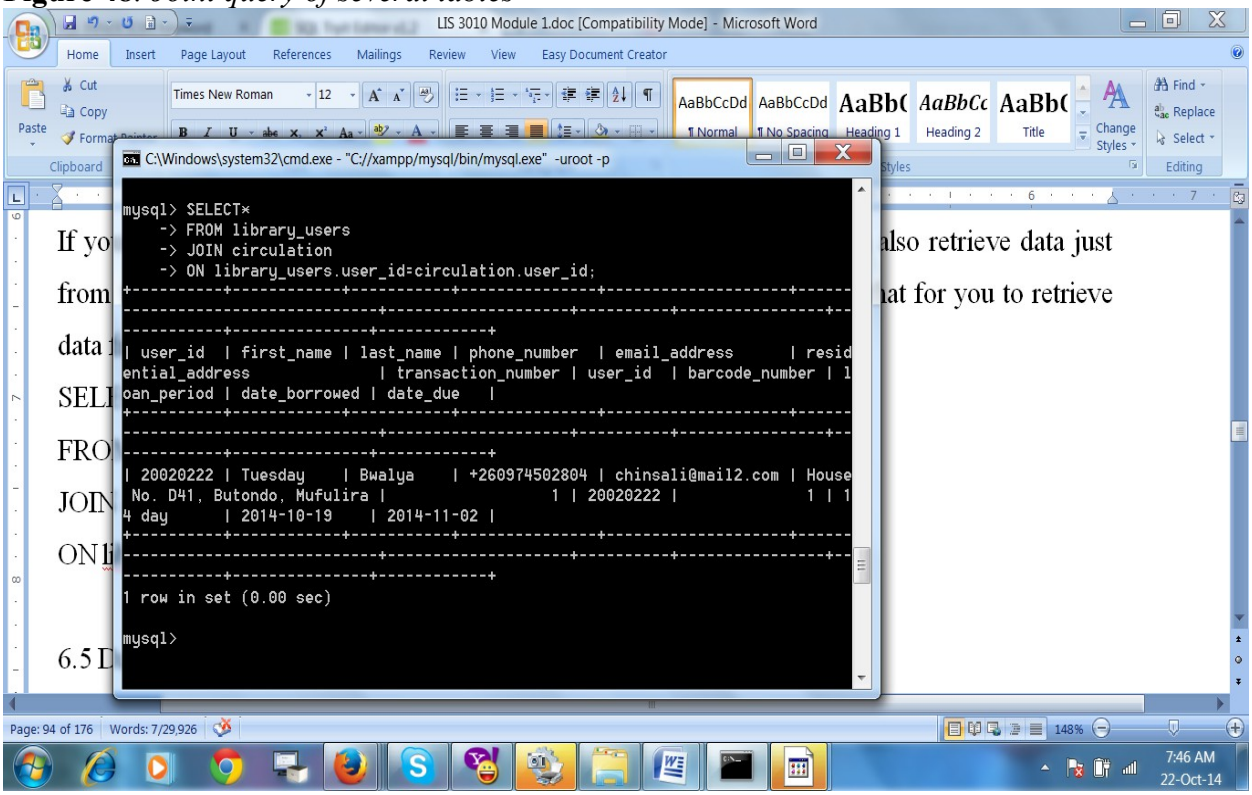
SELECT*

FROM library_users

JOIN circulation

ON library_users.user_id=circulation.user_id;

Figure 48: *Joint query of several tables*



There are many types of JOIN queries that you can execute in MySQL. There are also union queries that wok similar to join queries. You can read about them as well.

6.5 Data manipulation

There are situations when we discover that the data entered into the system is wrong, therefore it needs to be changed. You can do that with the UPDATE command.

UPDATE ***table_name***

SET ***field1=”value1”, field2= “value2”***

WHERE ***some_field=some_value*****;**

In the above command, we begin by stating the command which is UPDATE. Then we proceed to declare the table in which we want to do the update. Thereafter, we declare the fields that we want to have data changed and state the new data to be entered. Remember to put open and closed inverted commas (“”) to the field values if they are not integers. Lastly, we need to create a condition so that MySQL changes data in only the needed field where the condition will be met. In most case the field to use to create is a condition is preferably a primary key unless it the

field in which you want to change data. Let us now replace the above command with our table details.

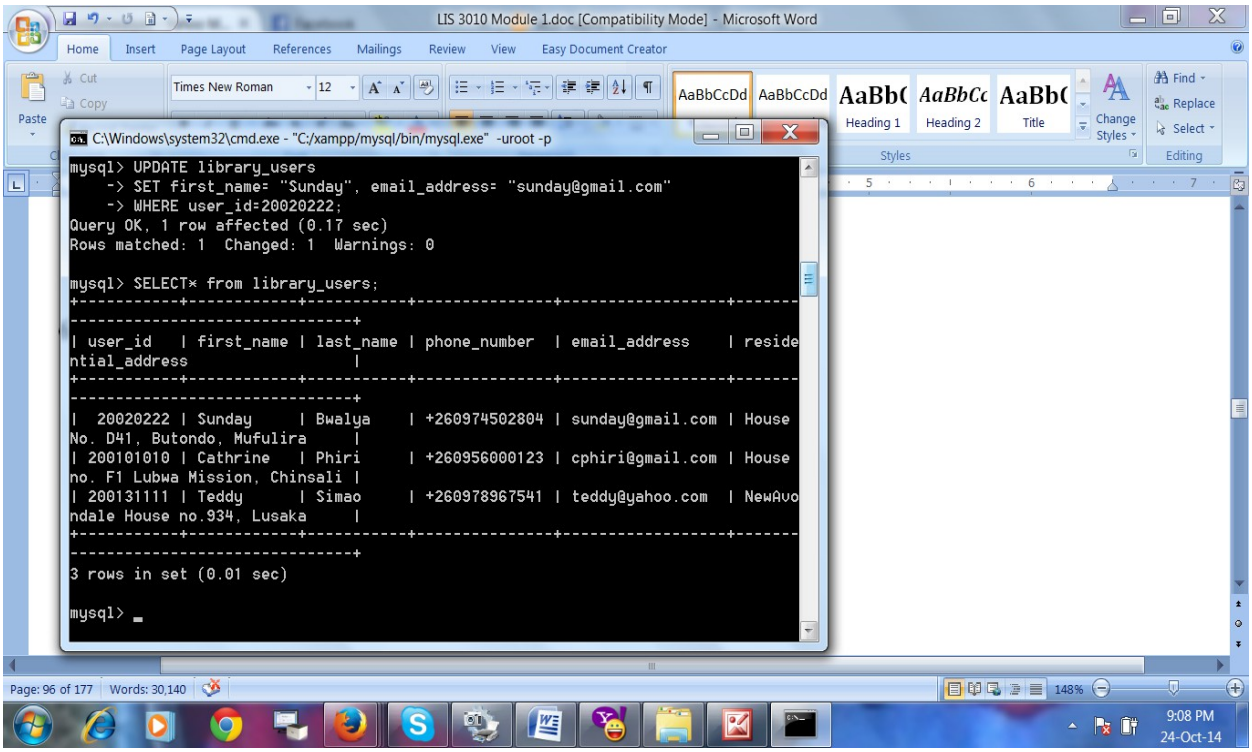
UPDATE library_users

SET *first_name*= "*Sunday*", *email_address*= "*sunday@gmail.com*"

WHERE *user_id*=20020222;

After executing the query successfully, you see a message saying query OK. Then SELECT * from the library_user table, you have the following results.

Figure 49: *Updating data in the table*

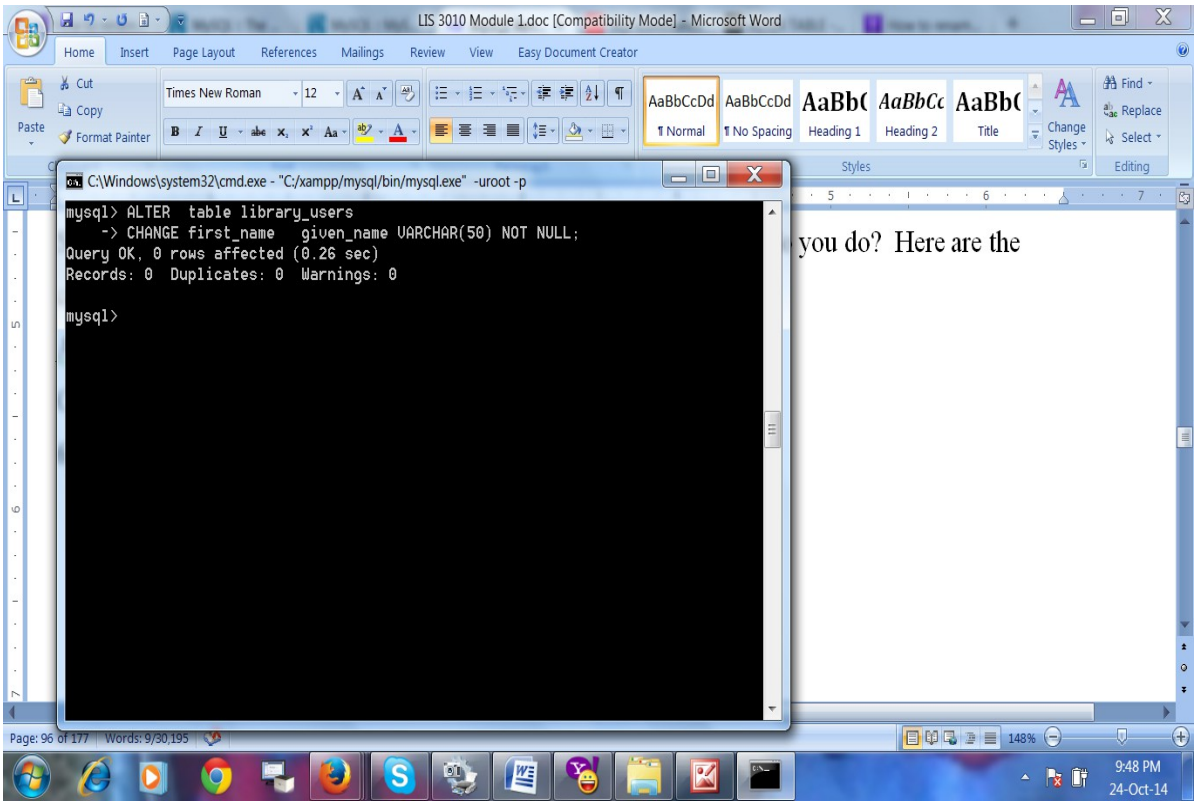


Suppose, you want to change the table name or the field name, what do you do? Here are the commands to use.

ALTER table library_users

CHANGE first_name given_name VARCHAR(50) NOT NULL;

Figure 50: *Alter some fields of a table*

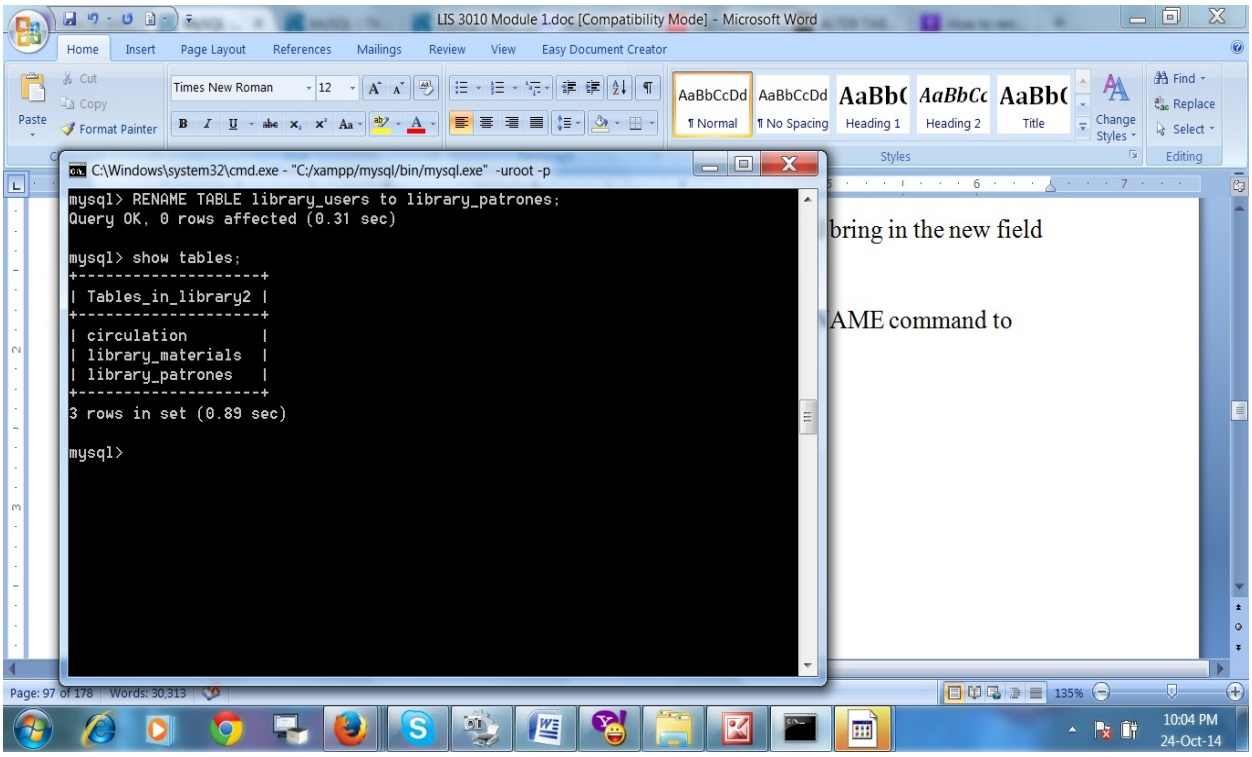


In the above example, you have learnt how to use two commands. These are ALTER and CHANGE commands. In the ALTER command, we declare what part of the database we want to alter. In this case, it is the table and we proceed to mention table name. Thereafter, we introduce another command called CHANGE which is followed by the name of the filed or column we want to change or rename. Then we press tab (create space) and bring in the new field name and its data type.

If you want to rename the database or table, you need to use the RENAME command.

RENAME table library_users to library_patrones;

Figure 51: *Change the name of the table*



When you send a SHOW tables command to MySQL, you see that indeed the table library_users has been renamed to library_patrones.

You can also delete data from the tables, and delete both the tables and tables with the following commands.

DELETE from table_name

WHERE (condition) feild1=value;

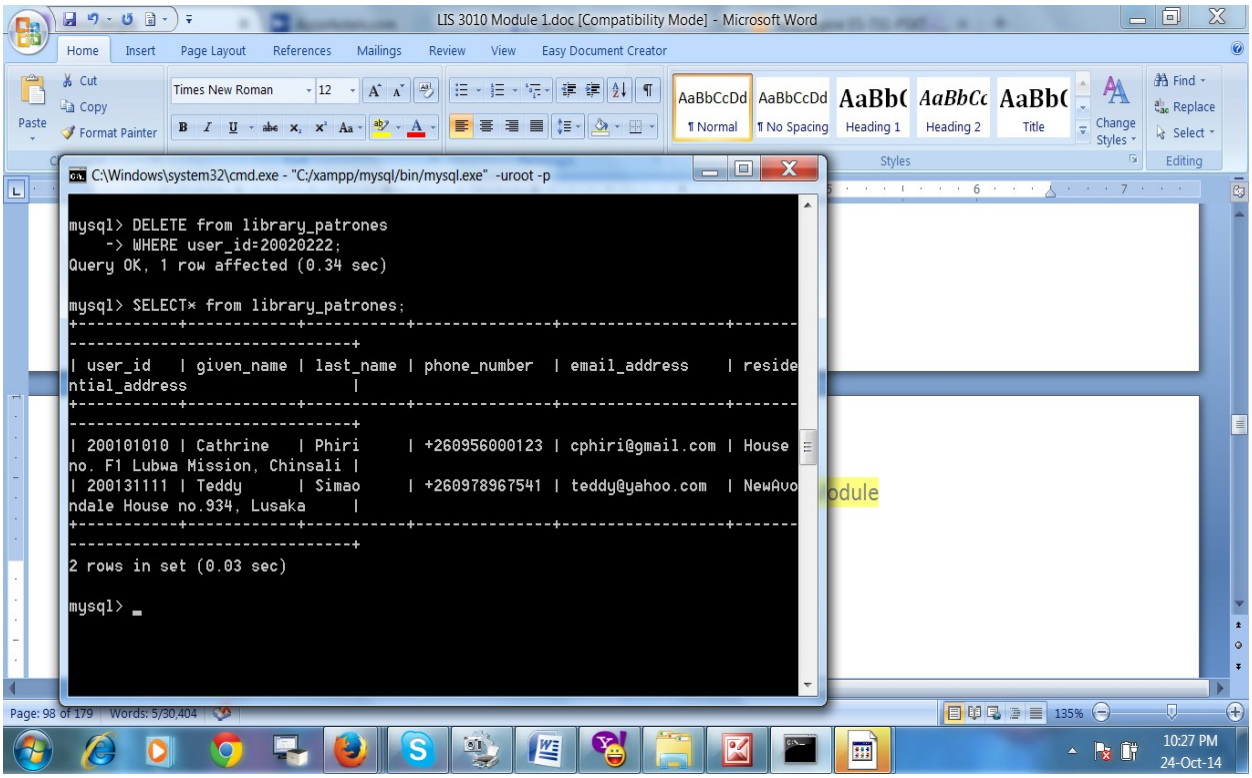
This command implies that you want to delete the entire record from the table which will meet the condition set in the query.

So if we use our renamed table called library_patrones and say:

DELETE from library_patrones

WHERE user_id=20020222;

Figure 52: *Delete data from a table*



Once you use a SELECT command, you see that one record with user_id 20020222 has been deleted.

If you want to delete the table or database, use the DROP command, everything will go.

DROP database library2;

DROP table circulation;

If you don't want to delete the database or table don't use this command unless when you were login onto MySQL just before starting transacting with it, you had set InnoDB **rollback** settings to zero as follows:

SET autocommit=0;

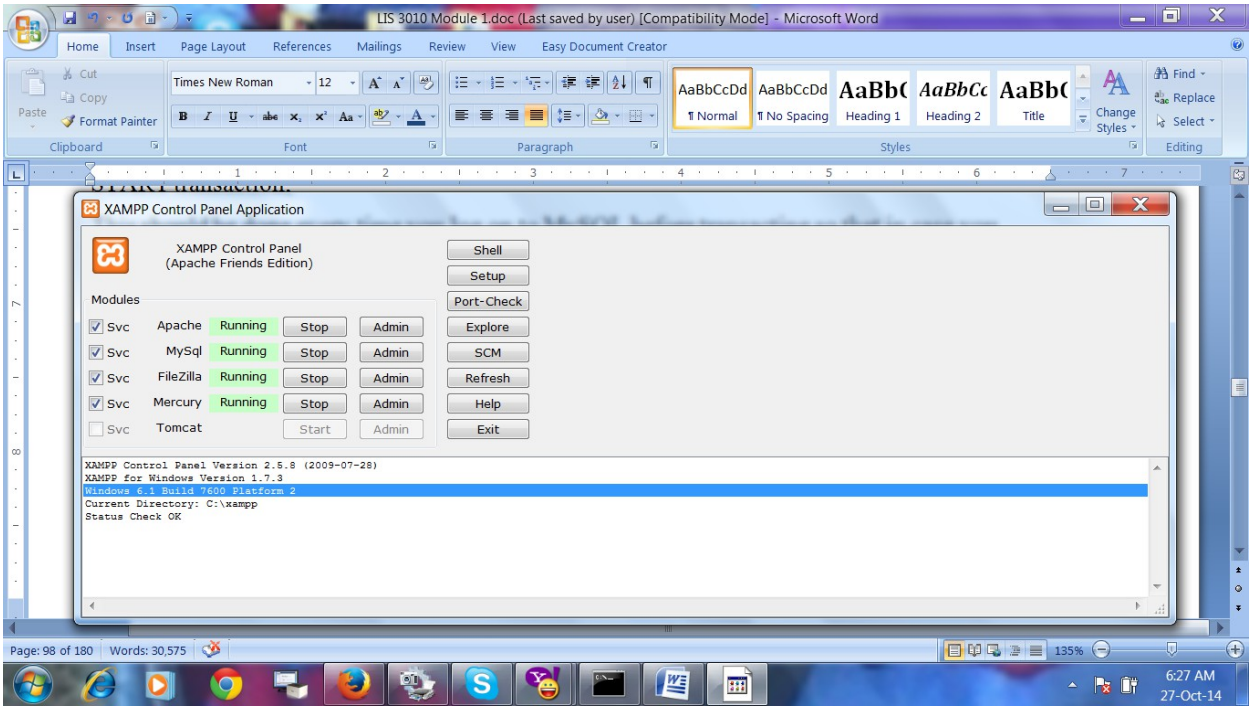
START transaction;

This should be done every time you log on to MySQL before transacting so that in case you delete some data or table unknowingly you can issue the rollback command after realizing the mistake made.

6.6 Introduction MySQL graphical user interface

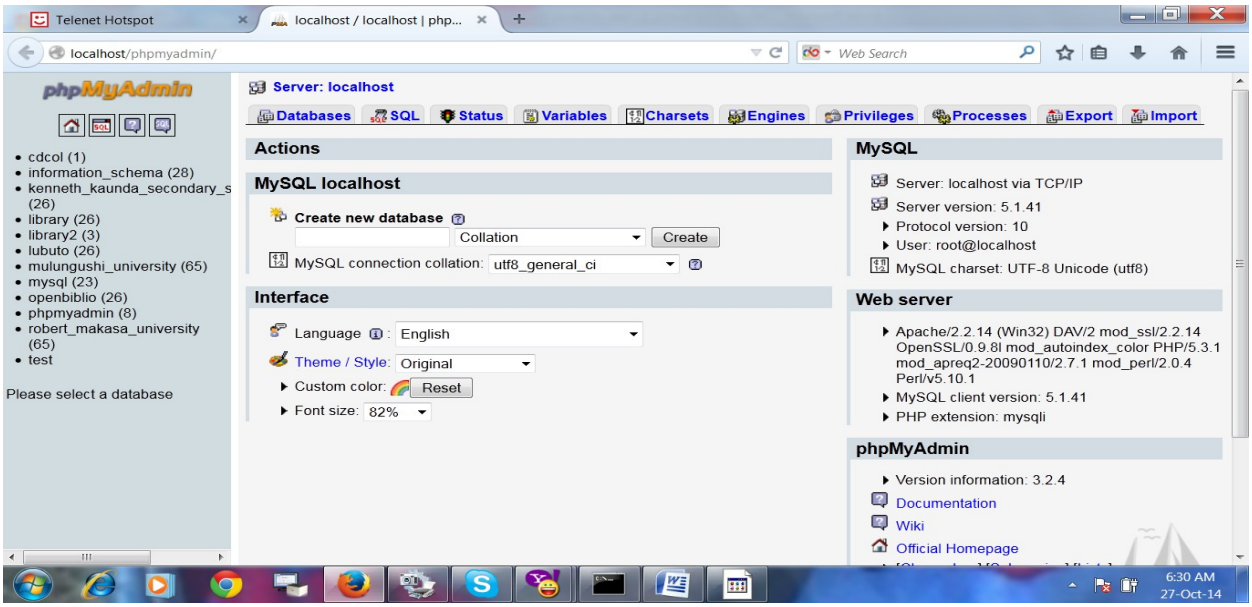
Congratulations for doing everything using the command line. You can also do all we have done in the graphical user interface of MySQL. But it is good idea to use the command line. We can now go to the graphical user interface by starting our Xampp if it was off and start all the programs. Having started the programs, click the **MySQL admin** icon, the graphical user interface will open.

Figure 53: *Starting Apache and MySQL*



In the graphical user interface, you will see the database that we have created. You can even browse the tables and their contents in the graphical user interface.

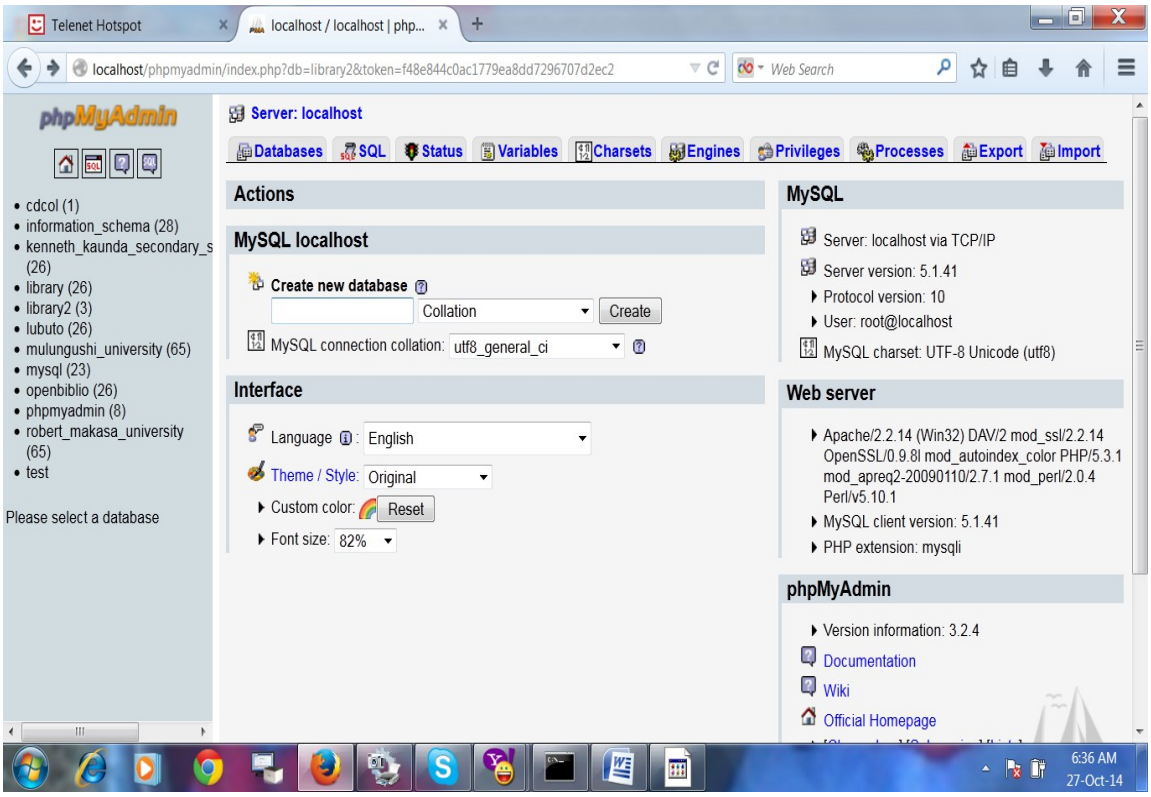
Figure 54: *Graphical user intarface of MySQL*



You can see that the database called **library2** is indicating having three tables. This is correct. You can click on it and see the tables.

It is easy to create a database in this interface by typing the name of the new database in the combo box for **create new database**.

Figure 55: *Creating a database in graphical user intarface of MySQL*

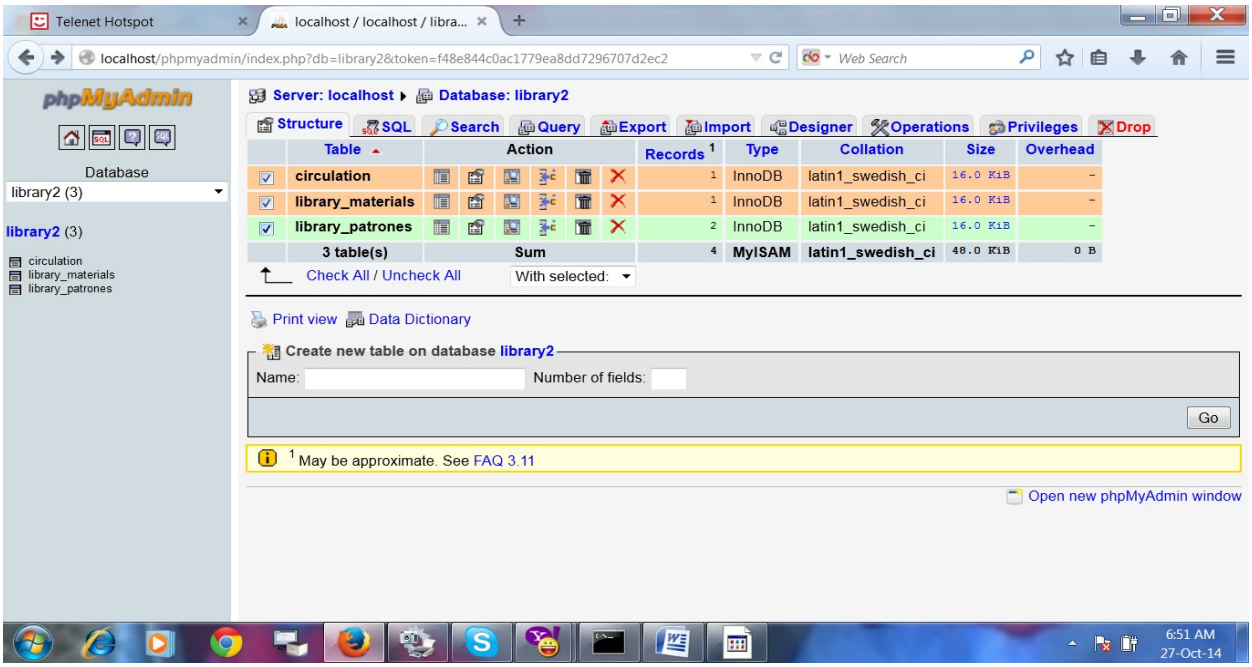


After naming your database, you can proceed to define the tables inside it like we did in the command line. Inserting of data into the tables should however be done using the INSERT SQL commands.

6.7 Backing up and restoring database

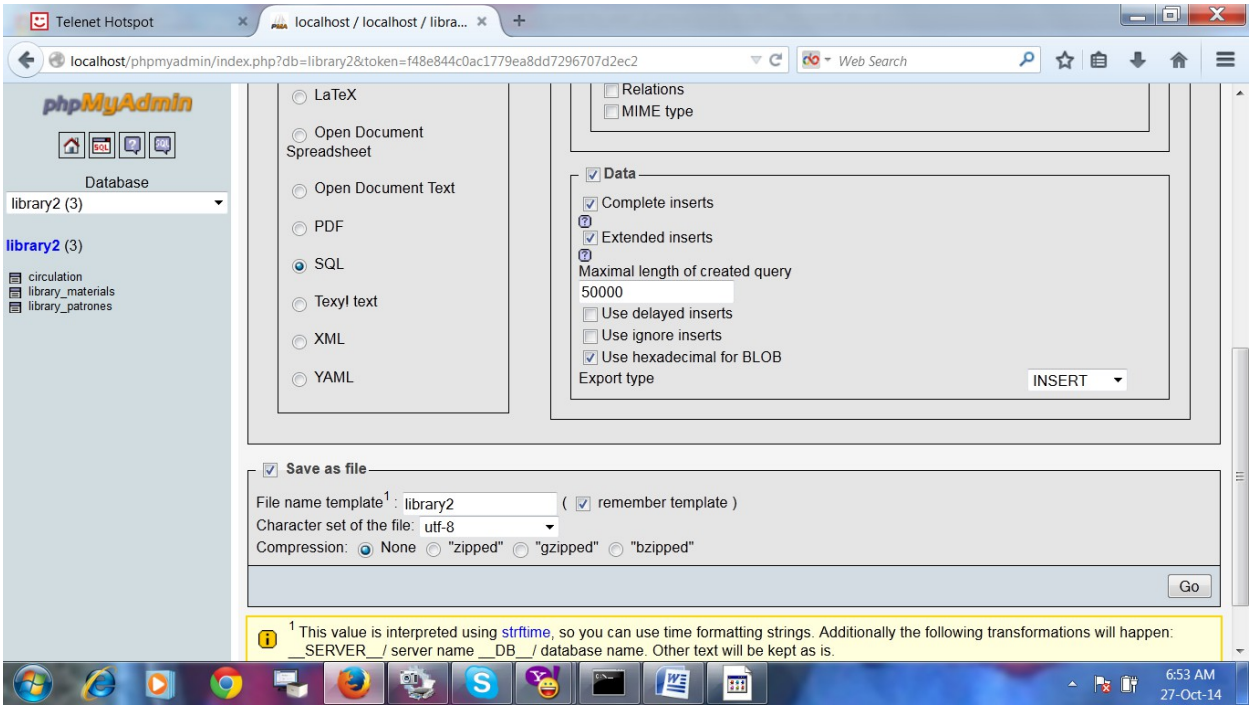
To back up MySQL database is easy. Go to the graphical user interface of MySQL and select the database you want to backup, click on it. Let us for example backup our library2 database. Once I click on it, the three tables inside will be shown. Tick all the three tables. Then look for the top icon called **export** and click because we want to send data to another storage environment.

Figure 56: *Selecting database to back up*



Once you click on **export**, a window will open where you need to type the name of the database where the three tables will be stored as a single database.

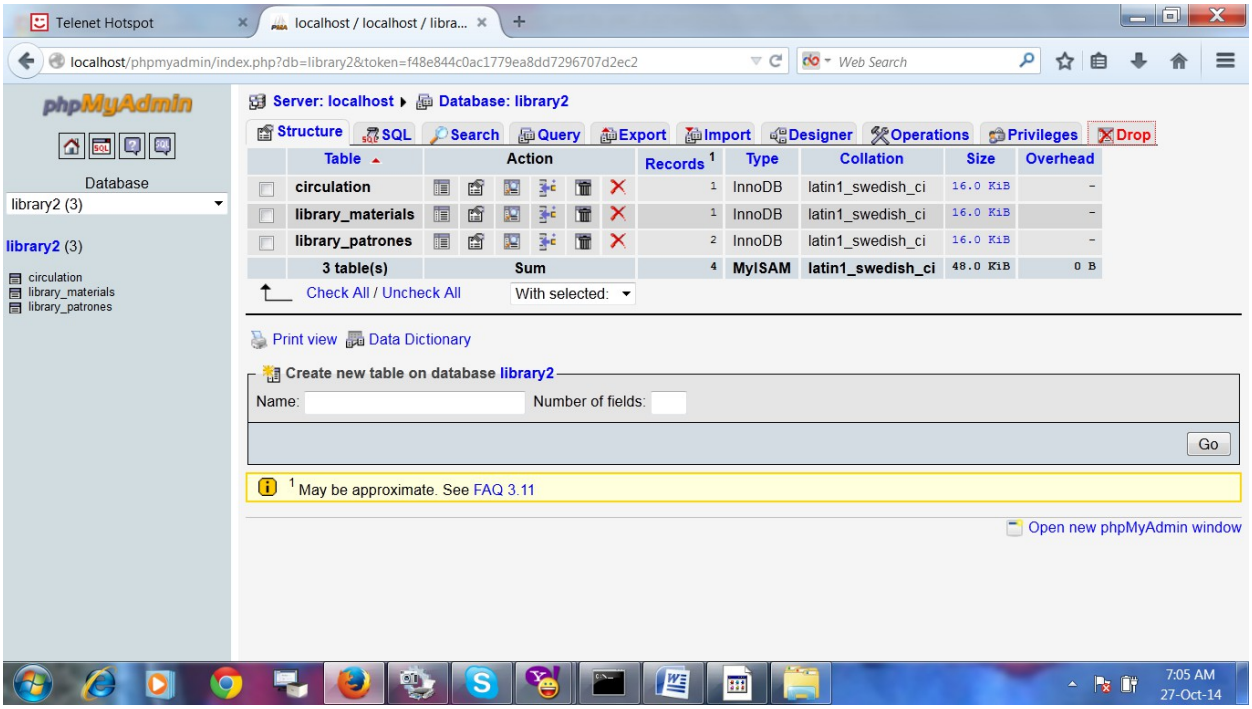
Figure 57: *Selecting file format of database to back up*



Remember to leave the file type as SQL. Do not un-tick this selection of file type. Then click on **Go** icon in the bottom corner, your database will be exported and downloaded. Copy the downloaded MySQL database file and save it on any backup device you are using. It is advisable to save it on external storage devices such as **external hard drive**. In my case the database is very small. It is in kilobytes. For me, I will keep it on the desktop because we will use it for restoration once our database is deleted.

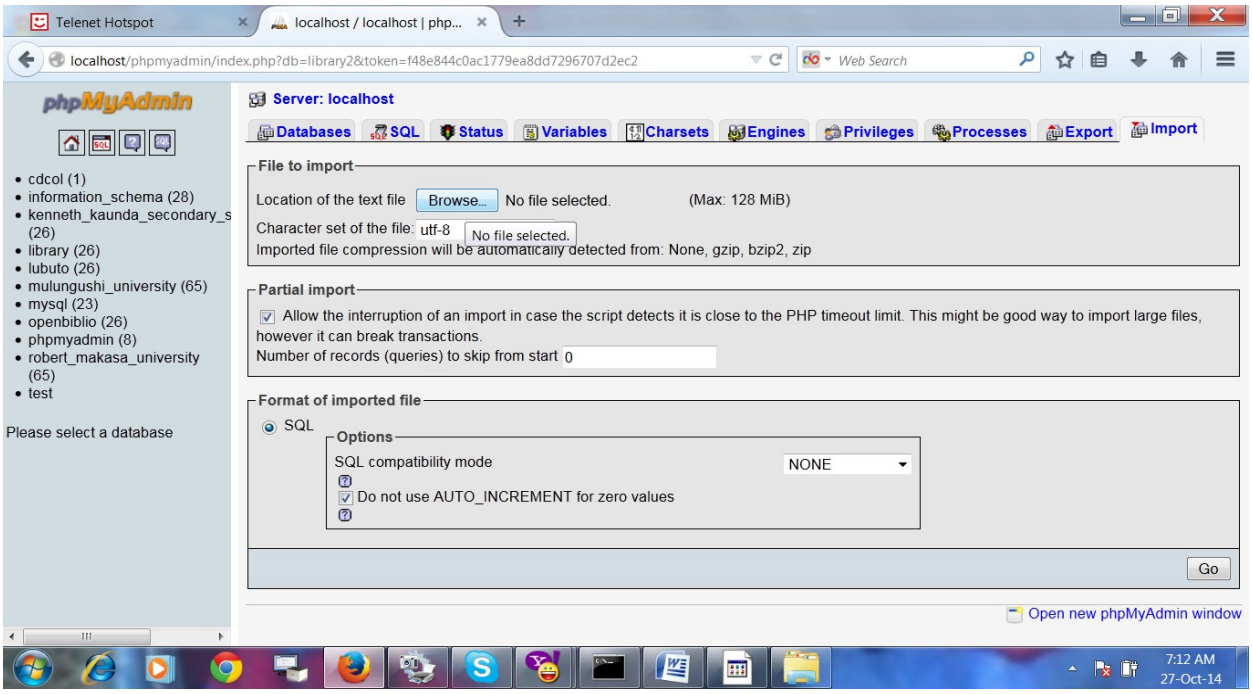
Let us now delete our database called library2 and restore it using the backup database. To delete the database, just go back to MySQL Home, where all the database are found, click on the database and in the window that will open, click the icon **drop,** MySQL will ask you to confirm if indeed you want to destroy the database. Just click **Ok** to proceed and you database will be gone.

Figure 58: *Delete a database from MySQL*



Go back now to MySQL home, you will not see our database library2. To recreate type the database name in the **new database** combo box and click on create icon. After creating an empty database called library2, then click on import, the new window will open in which you need to browse for the database that you want to import.

Figure 59: *Importing database into MySQL*



Click on browse and choose the storage area where you have saved the database to import. In my case, I will go to desktop, **click** my library2.sql file and finally click **ok**. It will be imported and a database with the three tables will be created again. **Note** that if the database file name and the name of the database to be created are not the same, you will see errors.

Summary

In this unit we have learnt how to log onto MySQL using the command line and use the command line to create the database and its tables using SQL statements. We learnt that to check the databases or tables that are already existing, we need to use the SHOW statement. To create the database and tables, we use the CREATE statement. To put data into the tables, we use the INSERT statement. To make changes to table name or fields, we learnt to use the ALTER statement. Further we learnt how to retrieve data from the tables using a variety of queries. One the queries or statements learnt is the SELECT* which retrieves data on all the fields of a specified table. We also learnt how to retrieve data using the WHERE condition in the SQL statement. This unit has also shown us how to use the MySQL graphical interface to create a

database. We also learnt how to back up and restore our database through the graphical interface using the click, click way.

Activities

a) Create a database with three tables for an organization of your choice and practice how to query them.

- ✓ Table1 should employees biodata
- ✓ Table 2 should have information about employees departments
- ✓ Table 3 must have employees' supervisors.

References

Beighley. L and Morrison. M (2009). *Head First PHP &MySQL*. Cambridge: O'Reilly.

Chao.L (2006). *Database Development and Management*. Boca Raton: Taylor &Francis Group.

Haag. S, C. M and Philips. A (2007). *Management of Information Systems for the Information Age*. Beijing: China Machine Press

Hoffer, J.A, Ramesh. A and Topi. H (2011). *Modern Database Management*. New Delhi; Chennai: Pearson; Dorling Kindersley

Kifer. M, Bernstein.A and Lews, M.P (2005). *Database Systems: An Application-Orientation Approach*. 2rd Ed. New York: Pearson Education

Kochtanek, T.R and Mathews, J.R (2002).*Library Information Systems: From Library Automation to Distributed Information Access Solutions*. Westport: Libraries Unlimited

Kroenke, D.M (2005). *Database Processing: Fundamentals, Design, and Implementation*. 10th Ed. Beijing: Publishing House of Electronics Industry

Laudon, K.C and Laudon, J.P (2006). *Management of Information Systems*. 6th Ed. Beijing: Pearson Education

Malaga, R.A (2006). *Information Systems Technology*. Beijing: Publishing House of Electronics Industry

Narang, R (2009). *Database Management Systems.* New Delhi: PHI Learning

Scholtiz, M (2006). *Application of Executive information systems in South Africa***.** [Online]Available:http://libserv5.tut.ac.za:7780/pls/eres/wpg_docload.download_file?p_filename=F2058583165/mariusscholtz.pdf. [Accessed on 16/11/2013]

Stephens. R, Plew. R and Jones, Arie. D (2012). *Sams Teach Yourself SQL*.4th Ed. New Delhi: Pearson; Dorling Kindersley