A TUTORIAL APPROACH FOR TEACHING DATABASE CONCEPTS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Susbi Sharma

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

March 2012

Fargo, North Dakota

North Dakota State University

Graduate School

Title

A TUTORIAL APPROACH FOR TEACHING

DATABASE CONCEPTS

By

SUSBI SHARMA

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Kendall Nygard	
Chair (typed)	Chair (signature)
Dr. Gursimran Walia	
Dr. Juan (Jen) Li	
Dr. Jim Coykendall	

Approved by Department Chair:

April 03, 2012	Dr. Kenneth Magel
Date	Signature

ABSTRACT

Experts in the education and psychology fields have undertaken immense research in order to find better learning approaches. Educators, teachers, cognitive psychologists, and researchers eventually accepted and developed a two-dimensional framework, focusing on both knowledge as well as cognitive processes. In combination, this two-dimensional framework clearly defines what a student is expected to learn in school. This framework is well described as well as addressed in the six different levels of the Cognitive approach in the Revised Bloom's Taxonomy.

Taking into account one of the most commonly used learning approaches in the field of education and psychology, i.e., Bloom's Taxonomy, this paper emphasizes the teaching database development in a step-by-step, task-oriented approach. All task-sections of the paper accommodate the lower-order thinking levels to provide the basic knowledge required. Once the basic concepts are explained, the user is expected to work on his own, following the directions and instructions provided.

.

ACKNOWLEDGEMENTS

I would like to thank Dr. Kendall Nygard for his guidance, help, and support. My sincere thanks go to Dr. Juan Jen Li, Dr. Gursimran Singh Walia, and Dr. Jim Coykendall for serving on the graduate committee.

I would also like to thank my mother and my grandparents. They have always been the biggest support and encouragement throughout my life.

Finally, I would like to thank a very close friend of mine, Mr. Manu Bhogadi, for his continuous support and encouragement.

DEDICATION

I am dedicating my work to my beloved mother, Mrs. Asha Sharma, who has continually supported and encouraged me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1. INTRODUCTION

The study described in this paper concerns the design and development of tutorial materials to assist and support students who are interested in learning the fundamental principles of Database Management Systems. When working with students, it is required, for their learning process, that we recognize certain attributes about the kind of learning tasks we are giving them to do. We typically assign reading or written notes to them, followed by some discussion, and end up with some form of evaluation [1]. There has always been a need for a better learning system, but it is very difficult for teachers to uncover an efficient learning system that helps provide knowledge to the students in the best way. Without a proper structure, no matter what the learners are tending to learn or understand, things do not seem to change or improve. Similarly, beginners/students who do not have any basic fundamental knowledge in the area of database development undergo the same problem. There needs to be a proper structure, or framework, addressing different levels of learning for our learners; the framework needs to cover the knowledge of Database Management Systems, starting from the fundamentals and leading to an in-depth knowledge.

Therefore, the main issue that this paper addresses is to present better learning skills and techniques for our learners to understand and grasp the fundamental principles of databases. The paper has a tutorial-based approach to instruct users about how to operate with data from using flat files like Excel spreadsheets to using an Access database to maintain records. Further, the user is also exposed to the knowledge of querying data based on SQL-Structured Query Language commands. Here, the users will learn how resultant data extracted from the large database can be manipulated using different additions or modifications to the queries. As a whole, emphasis provides the user with a basic layout to understand as well as to write queries using Structured Query Language.

1.1. Objective and Tasks

The objective of the study is to design and develop a software system to support students in learning the principles of database management system. Educational learning theory is used as a guide.

Following is a brief outline of the tasks conducted to meet the objective:

- Tutorials specifically aimed at teaching database fundamental principles, definitions, operations, etc. are designed following an example-oriented approach.
- The first tutorial is designed to teach the rationale of using an Access database in contrast to Excel spreadsheets. The tutorial provides a detailed, task-oriented approach, offering users hands-on experience working with Access database operations.
- The second tutorial is comprised of definitions for the key terminologies of SQL concepts and commands; introduces the syntax for SQL commands; provides examples based on each of the SQL commands; and, last but not least, outlines each task in the tutorial with hands-on working exercises towards the end of the task.
- All components of the two tutorials designed in the paper are precisely and explicitly related to the revised Bloom's Learning Objectives.
- The paper signifies that an example-based approach, along with a detailed set of hands-on exercises at the end of each task in the tutorial, can provide an ideal learning environment for the user that settles evenly with the levels of the Revised Bloom's Taxonomy of Educational Objectives.

1.2. Significance of this Work

Among the various challenges that people face while learning a new concept is a proper structure of learning. It is difficult to grasp complex concepts in the initial phase without having any basic fundamental knowledge. Another important criterion here is the amount of content that individuals are trying to learn. Often, learners try to read back and forth in order to understand the key fundamentals, or concepts, but lack a proper structure [1].

Therefore, the idea is to propose a step-by-step, laid-out structure that more closely reflects information about how we organize information and ideas in relation to advanced concepts of developing, designing, and maintaining database system. This systematic, procedural approach illustrates better learning techniques for students from the Revised Bloom's Taxonomy in relation to database management. The paper includes, at first, the lower-order thinking skills to provide the basic knowledge to students as well as the basic learning skills. Once the basic concepts are explained, students are able to better understand concepts and to follow the task-oriented learning process. The tutorials designed in the paper not only provide knowledge to the users, but also offer the users some space to apply that knowledge while analyzing it in their own way.

Each task is laid out in the tutorial following the Revised Bloom's Taxonomy according to the level of complexity. The tasks in the tutorial are explained in such a fashion that, as the users progress in the tutorial, they get a chance to apply their own knowledge and understanding of what they have learned. The most important aspect of the tutorials in this paper is that they lay out all the basic concepts that concentrate on understanding as well as application and allow the user to play around with the concepts. The tutorials shall prove beneficial to any user, whether a student or not, in order to grasp the database fundamentals as well as to learn to work with databases, whether using Access or SQL.

1.3. Organization of the Document

Chapter 2 discusses the role of taxonomy in the field of learning. It discusses the various learning taxonomies being used from the past. Chapter 2 uncovers the history of Bloom’s Taxonomy as well as introduces the Original Bloom’s Learning Objectives. Chapter 3 illustrates improvisations in the original Bloom’s Taxonomy model and introduces the Revised Bloom’s Taxonomy. Chapter 3 also discusses the relationship between the Revised Bloom’s Learning Objectives and this present work. Chapter 4 describes the first step in developing a user’s database background utilizing spreadsheets as well as working with an Access database. This tutorial introduces a database’s fundamental principles with eight tasks covering both Excel spreadsheets as well as Access backgrounds. Also, Chapter 4 includes the mapping between eight of these tasks with the Cognitive levels in the Revised Bloom’s Taxonomy. Chapter 5 is the second and last step in the tutorial; this step extends the knowledge and understanding of the users further to the advanced concepts of databases as a transition from an Access database to an SQL database. Both Chapter 4 as well as Chapter 5 illustrates mapping between each of the task-sections in the tutorial with the Revised Bloom’s Taxonomy. Chapter 6 contains the conclusion and limitations of the paper. Chapter 6 also provides a glance of the future work.

CHAPTER 2. LEARNING TAXONOMIES AND BLOOM'S TAXONOMY

What is the first thought that comes to our mind when we hear the word "learn"? Some people might say "learn" means cramming. Others would say to first understand and then maybe even learn by heart. Some people may address "learn" as just understanding by examples or relating to some real-time situation [1]. Thus there can be several more classifications or distinctions about how an individual develops a learning technique or habit. It is, in fact, quite difficult to classify the learning techniques of every individual. Imagine yourself in a scenario where you can study and learn something completely novel to you. There may be many dilemmas that grow in your mind; some of them might even be just how you should start learning or even why you should learn that particular thing. Subsequently, your mind would wander to understand the rationale for learning something new that may or may not be accomplished with an existing system with which you are familiar. Thus, no matter what we need to learn or understand, there must be a technique or an approach that appeals and encourages us to learn something novel with a proper understanding (fundamental knowledge) of it. As we have already discussed, although it is difficult to enumerate learning techniques for different individuals, it is quite feasible to design a sequential and iterative structure that enables an individual to acquire a proper learning pattern.

2.1. What is Taxonomy?

Taxonomy is the science of characterizing and classifying. It is a structure-based approach to classify different species. Taxonomy deals with developing a classification scheme or arranging a particular class in a hierarchical structure. The taxonomy that I am emphasizing in this paper is that of educational objectives. This taxonomy deals with different objectives of teaching as well as learning. Learning taxonomies attempt to break down and categorize types of learning to help designers develop objectives and learning strategies that best match the specific type of learning that is targeted [2].

2.2. Learning Taxonomies: At a Glance

There have been several learning taxonomies developed and used. Some are well known and used very commonly, whereas the others are not very well known and not so commonly used. The following sections describe a few of these learning taxonomies, and give a short description about them [2].

2.2.1. Bloom's Taxonomy

Benjamin Bloom's taxonomy is probably the best known, and it breaks learning into the three commonly used categories of cognitive, psychomotor, and affective learning. Each category has a number of subcategories [2].

2.2.2. Robert Gagne's Taxonomy

Robert Gagne had significant influence with his taxonomy which he labeled "learned capabilities." Gagne's highly developed system of learning design included "conditions of learning," or unique instructional strategies, that were required to develop each learned capability [2].

2.2.3. David Merrill's Component Display Theory

David Merrill's Component Display Theory (CDT) classified learning along two dimensions: content (facts, concepts, procedures, and principles) and performance (remembering, using, and generalities). David Merrill's Content/Performance matrix was found to be quite helpful. Its most contributing feature was the distinction between knowing and doing [2].

2.2.4. A.J. Romiszowski

He developed the knowledge/skill distinction even further and added the dimension for a variety of categories of skilled performance. He designed an approach to clarify skilled behavior [2].

2.3. Why Bloom's Taxonomy?

Learning, teaching, identifying educational goals, and thinking are all complicated concepts interwoven into an intricate web [3]. Benjamin Bloom believed that learning is not something that merely aims at giving information to our learners, but in fact, it is a detailed design and structure to identify the educational goals that need to be met [3] . Let us say that, when a particular course is created for students, it is designed keeping in mind the learning objectives which indicate what students will be able to do when the course ends. For example, "After you have completed two sessions of this course, you shall be able to"

- Identify positive and negative integers in the number line.
- Perform operations like addition and subtraction on the integers from the number line.

Notice that the verbs "identify" and "perform" are certain criteria that could be measured in some way, whether it is an exam or by observation. This measurement criteria is the most crucial and elemental factor when stating the learning objectives. In principle, we are making an attempt to explain how students would self-realize the fact that they achieved or learned something on the basis of the assessment they underwent.

Bloom's taxonomy provides a very clear formula for thinking about instructional design, i.e., objectives, class activities, and assessments. Using Bloom's Taxonomy to design our courses does the following things. It informs the students [4]

- What they should study

- How they will be assessed

It guides the instructor [4]

- In assessment strategies
- In teaching strategies

It tells the instructor (and accreditation agencies) [4]

- If teaching strategies worked
- If assessment strategies worked

2.4. Bloom's Taxonomy: History at a Glance

Bloom's Taxonomy, emphasized and explained that any given task favors one of the three psychological domains: Cognitive, Affective, and Psychomotor. The original intent in creating the taxonomy was to focus on these three major domains of learning [5]. The Cognitive Domain covered "the recall or recognition of knowledge and the development of intellectual abilities and skills"; the affective domain covered "changes in interest, attitudes, and values, and the development of appreciations and adequate adjustment"; and the psychomotor domain encompassed "the manipulative or motor-skill area" [5].

2.5. Bloom's Taxonomy: Original Model

The original Bloom's Taxonomy was comprised of six Cognitive development categories: knowledge, comprehension, application, analysis, synthesis, and evaluation as shown in Figure 1. Among these six different categories of learning styles, the first two, knowledge and comprehension, were explained by Benjamin Blooms as the lower order of learning [5]. The remaining four learning styles were considered and classified by Benjamin Bloom as higher-order learning styles in the Cognitive Domain hierarchy [5]. The reason behind this classification was that these higher-order thinking skills required motor-skill techniques.

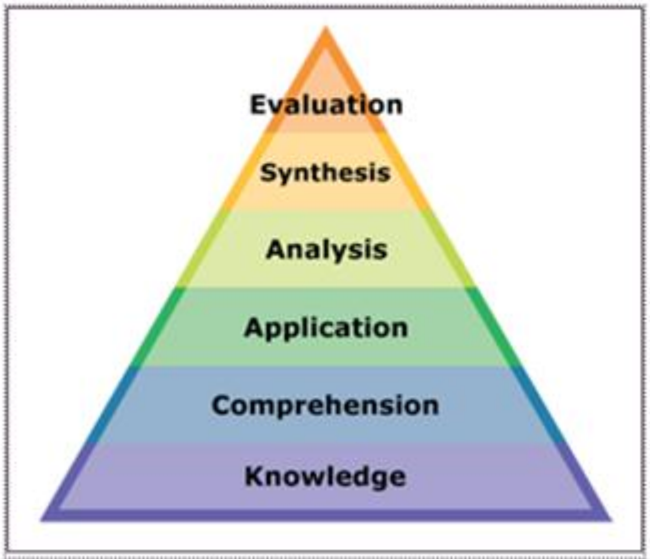


Figure 1. Original Bloom's Taxonomy Hierarchy.

2.5.1. Knowledge Level

This is a lower-order learning process. It mainly requires recalling memorized information. From a small to a wide range of information, the learner is mainly involved in remembering it [6].

2.5.2. Comprehension Level

Another lower-order learning process in the Cognitive Domain, comprehension underlines an individuals or a learner's ability to grasp the meaning of particular material. In other words, this level is one step ahead of the knowledge level where an individual only aims at remembering material [6].

2.5.3. Application Level

Moving up in the hierarchy of learning levels in the Cognitive Domain, the first higher-order learning level that appears is the application level. The application level involves applying knowledge and understanding in a real-time situation. In other words, the application level includes applying new theories, rules, methods or concepts, laws, or principles in a concrete situation [6].

2.5.4. Analysis Level

To be able to analyze is to be able to break material into its substituent parts. In analysis, one needs to identify and recognize the principles involved. Analysis represents a higher intellectual level than comprehension and application because the analysis level requires an understanding of both the content and the structural form of the materials [6].

2.5.5. Synthesis Level

Synthesis is nothing but putting together or, rather, assembling ideas into a new or unique product or plan. Synthesis represents a rather high intellectual learning level because it extends learner’s thinking and allows them to create new ideas from the older ones [6].

2.5.6. Evaluation Level

The evaluation level is the highest hierarchy level; the learner’s makes necessary decisions about the value of ideas, items, materials, etc. It is eventually at this stage where the learner needs to put forward all the information, knowledge, analyzing ability, and application that he/she has acquired about particular material [6].

CHAPTER 3. BLOOM'S TAXONOMY: REVISED MODEL

3.1. Introduction

As the years passed, with a growing pattern of knowledge and understanding, there was need to reorganize Bloom's Taxonomy. The new era of learners was emerging, and their needs and abilities varied significantly. Hence, there was a call for alterations and modifications in the patterns of learning styles [7]. With the required revisions in Bloom's Taxonomy being implemented, the need of the hour could be met, i.e., changes in the growing needs and abilities of the new-age learners could be addressed. Therefore, in the 1990s, significant revisions were made to the taxonomy. These revisions included numerous apparently minor, yet quite substantial, changes. The modifications were made in three broad categories: terminology, structure, and emphasis [7]. In the modified version of Bloom's Taxonomy, the names of the major cognitive process categories were altered, keeping in mind that the learner's action as thinking involves bringing in manipulations and physical actions. Figure 2, shows the learning levels in the Revised Bloom's Taxonomy.

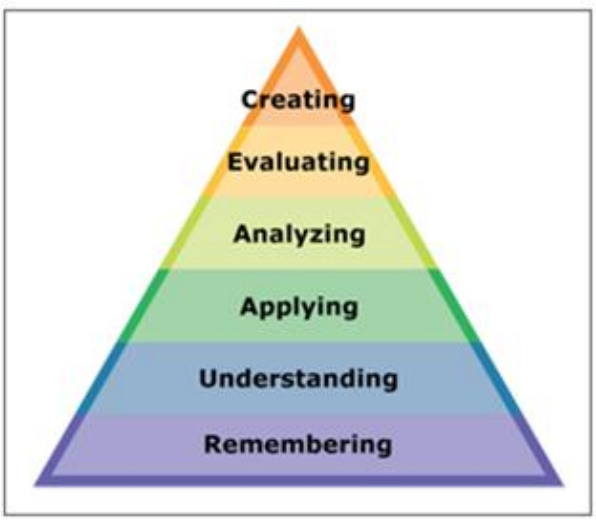


Figure 2. Revised Bloom's Taxonomy Hierarchy.

3.1.1. Remembering Level

Remembering involves retrieving, recognizing, and recalling relevant knowledge from long-term memory, e.g., find out as well as learn terms, facts, methods, procedures, and concepts [8].

3.1.2. Understanding Level

This level includes constructing a meaning from oral, written, and graphic messages through interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining. People need to understand the uses and implications of terms, facts, methods, procedures, and concepts [8].

3.1.3. Applying Level

Applying encompasses carrying out or using a procedure through executing or implementing. People need to make use of, apply practice theory, solve problems, and use information in new situations [8].

3.1.4. Analyzing Level

Analyzing, in simpler terms, can be described as breaking material into constituent parts as well as determining how the parts relate to one another and to an overall structure or purpose through differentiating, organizing, and attributing. People need to take concepts apart, break them down, analyze structure, recognize assumptions and poor logic, and evaluate relevancy [8].

3.1.5. Evaluating Level

This level associates with making judgments based on criteria and standards through checking and critiquing. People need to set standards, judge using standards, and evidence, accept or reject on basis of criteria [8].

3.1.6. Creating Level

Creating includes putting elements together to form a coherent or functional whole, reorganizing elements into a new pattern or structure through generating, planning, or producing. People put things together; bring together various parts, write themes, present speeches, plan experiments, and put information together in a new and creative way [8].

3.2. Why Modern Taxonomy?

Since 1948 when the taxonomy was originally created, dramatic changes in society have occurred. As a result, modifications have been made to the structure for which the taxonomy was first written. Several attempts have been made in the past and continue to be made in order to reorganize the taxonomy to meet the needs of new-age learners as well as to satisfy the call of teachers demands.

The current structure designed for the revised taxonomy aims at providing a clear and succinct illustration of the orientation between the standards and educational goals as well as the objectives to be met as far as learning goes. The main purpose of using the revised taxonomy with respect to this paper is because the revised taxonomy is designed by keeping the present education system and learners in mind.

3.3. Relating the Present Work with Modern Taxonomy Levels

3.3.1. Remembering Basic Concepts

The user/learner is provided with a tutorial-based learning approach. The user is asked to recollect basic concepts and an understanding of Excel spreadsheets and Access databases. Intermittently, the user is provided with notes about a particular topic being introduced. The notes are written and the information presented such that the user/learner is offered only the basic optimum information required, and henceforth, the concentration of the user is expected to be more on understanding those concepts thoroughly. Thus, the information is kept precise as the focus is more on the Cognitive Domain levels of understanding, analyzing, applying, and creating.

3.3.2. Understanding the Work Flow

To understand the work flow better no matter whether it is a course, any instruction set, etc., the best way would be to raise questions about the student's knowledge of the concepts relevant to that particular topic. This paper provides hands-on exercises for students to test their own understanding of concepts. There is a set of questions at the end of each topic/task which serves the purpose of testing a student's knowledge on that particular topic/task. The student is given a sample database and various operations. Tasks are performed from scratch to show the user how a better designed database, which is more efficient, can be created from what it was earlier.

3.3.3. Applying the Knowledge

The tutorial is designed, keeping in mind that it involves application. As I have already mentioned, the tasks are set, keeping in consideration that, after the end of every task, the user is required to undergo real-time application of the knowledge gained from the information pool provided in that respective task. It is as though you provide a map for a student with directions and then ask him/her to narrate the route from a particular place to another place. Similarly, the approach here is to provide the student with knowledge about how to perform various operations using databases, i.e., how to design, create, update, delete data from database, etc., and then ask him/her to perform some manipulations and physical activity in order to exercise his/her mind and furnish his/her concepts.

3.3.4. Analyzing the Details

As the tutorial proceeds through the higher levels of learning objectives, the student undergoes different tasks, and while performing various operations as a part of these tasks, the student gains the understanding as well as the analyzing ability to evaluate what he/she has performed so far and what needs more attention. The student shall be able to assess and analyze his/her scale of knowledge in working with databases while he/she proceeds through these tasks.

CHAPTER 4. DATABASE BACKGROUND

This paper emphasizes a tutorial-based approach. This approach is based on a few pre-assumptions. Among those pre-assumptions, one major assumption is the student's exposure to the use of Microsoft Excel and Microsoft Access. Although the student is not expected to know all the operations or functions that MS Access and MS Excel provide, it is assumed that the student should at least be familiar with what MS Excel and Access are, know why they are used, and have some basic use of them before. The tutorial is focused to bring some hands-on experience for the students in MS Access.

4.1. Database Concepts

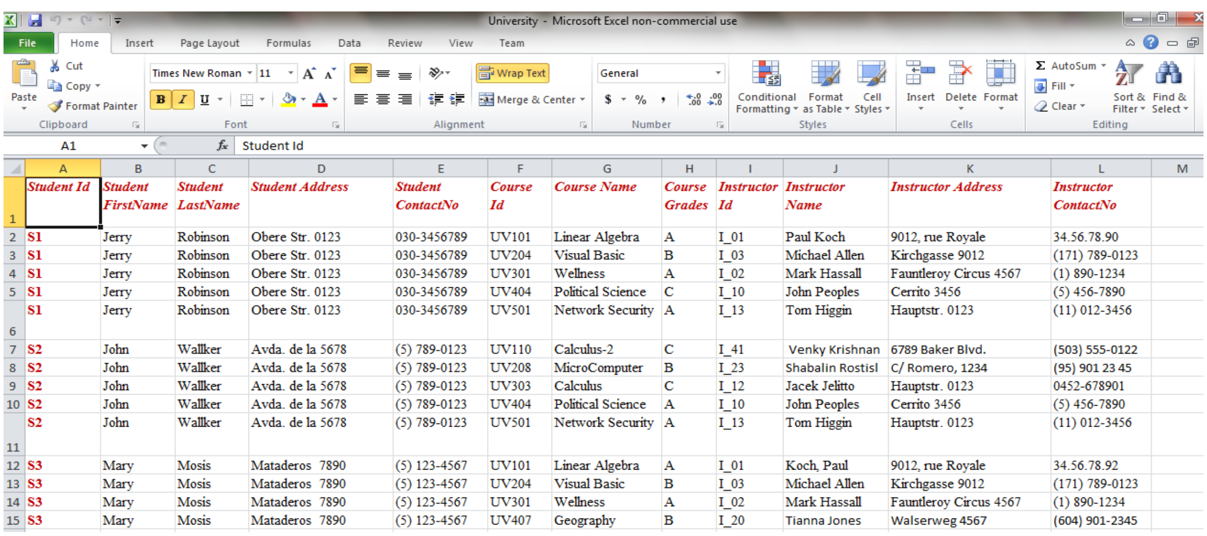
Under the assumption that the user/student possesses basic knowledge of databases and computer operations, the first step provides hands-on exercises for understanding the fundamental database concepts [9]. The user is provided with a raw database and is asked to perform operations on the database. In the initial phase, MS Excel is chosen to create the raw database file, which is also called a "flat file." Eventually, the user is asked to undergo a transition from MS Excel to MS Access on the basis of the explanations and the requirements that need to be met. Also, if the user does not have any pre-exposure to using MS Access before, there is a step-by-step instruction for every task to direct the user.

The tutorial based on MS Excel and MS Access is designed in such a manner that it is comprise of eight essential tasks which encompass a journey from MS Excel to MS Access, and offer knowledge as well as hands-on experience with the two Microsoft products used to handle the database and its operations [9]. There is no particular architecture for this task-oriented tutorial design because it is only concentrated on basic operations, because the operations are performed with MS Access, and because no other application is introduced.

This example-based tutorial assumes that you have access to the Microsoft Access database system. Access is part of the Microsoft Office suite, and it is available on computers in the clusters at NDSU. The instructions assume Access 2010 but can be modified slightly for Access 2007.

For use in this tutorial, there is a small sample database that stores information about university data records. The Excel file named “University” contains all the related data for the student, courses enrolled, grades, and information about the faculty. This file is posted for downloading on the Blackboard site. Save the given Excel file, “University,” from the Blackboard site. Figure 3 is a screenshot of the “University” spreadsheet.

Download the “Project 1” zip folder. Extract all the files from the folder. Once you extract all the files, you can see four files in the folder. Among the four files, two of them are Excel spreadsheets named “University” and “University_New.” There is another file named “Access_Help_Screenshots” which contains images that explain how to perform various tasks while working with Access database operations. This document is prepared as a guide to help users perform the tasks in the tutorial.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Student Id	Student FirstName	Student LastName	Student Address	Student ContactNo	Course Id	Course Name	Course Grades	Instructor Id	Instructor Name	Instructor Address	Instructor ContactNo
2	S1	Jerry	Robinson	Obere Str. 0123	030-3456789	UV101	Linear Algebra	A	I_01	Paul Koch	9012, rue Royale	34.56.78.90
3	S1	Jerry	Robinson	Obere Str. 0123	030-3456789	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 9012	(171) 789-0123
4	S1	Jerry	Robinson	Obere Str. 0123	030-3456789	UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Circus 4567	(1) 890-1234
5	S1	Jerry	Robinson	Obere Str. 0123	030-3456789	UV404	Political Science	C	I_10	John Peoples	Cerrito 3456	(5) 456-7890
6	S1	Jerry	Robinson	Obere Str. 0123	030-3456789	UV501	Network Security	A	I_13	Tom Higgin	Hauptstr. 0123	(11) 012-3456
7	S2	John	Wallker	Avda. de la 5678	(5) 789-0123	UV110	Calculus-2	C	I_41	Venky Krishnan	6789 Baker Blvd.	(503) 555-0122
8	S2	John	Wallker	Avda. de la 5678	(5) 789-0123	UV208	MicroComputer	B	I_23	Shabalin Rostisl	C/ Romero, 1234	(95) 901 23 45
9	S2	John	Wallker	Avda. de la 5678	(5) 789-0123	UV303	Calculus	C	I_12	Jacek Jelitto	Hauptstr. 0123	0452-678901
10	S2	John	Wallker	Avda. de la 5678	(5) 789-0123	UV404	Political Science	A	I_10	John Peoples	Cerrito 3456	(5) 456-7890
11	S2	John	Wallker	Avda. de la 5678	(5) 789-0123	UV501	Network Security	A	I_13	Tom Higgin	Hauptstr. 0123	(11) 012-3456
12	S3	Mary	Mosis	Mataderos 7890	(5) 123-4567	UV101	Linear Algebra	A	I_01	Koch, Paul	9012, rue Royale	34.56.78.92
13	S3	Mary	Mosis	Mataderos 7890	(5) 123-4567	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 9012	(171) 789-0123
14	S3	Mary	Mosis	Mataderos 7890	(5) 123-4567	UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Circus 4567	(1) 890-1234
15	S3	Mary	Mosis	Mataderos 7890	(5) 123-4567	UV407	Geography	B	I_20	Tianna Jones	Walserweg 4567	(604) 901-2345

Figure 3. Screenshot of University Spreadsheet.

4.2. Task 1: Examining the Spreadsheet

4.2.1. What is a Database?

A database is a collection of data describing the activities of one or more related organizations [10]. For example, a “University Database” might contain a stored data collection of the following items:

- Entities such as students, courses, classes, grades, faculty, and classrooms.
- Relationships between these entities, such as the students enrolled in courses, grades for each student, etc.

4.2.2. What is a Database Management System (DBMS)?

A database management system, or DBMS, is software designed to assist in maintaining and utilizing a large collection of data. It is a collection of programs that initiate to store, modify, and extract data/information from a database [11].

4.2.3. What is a Primary Key?

A Primary key is a key in a relational database that is unique for each record. It is a unique identifier in a data record table [11].

4.2.4. What is Redundancy?

Redundancy can be defined as having particular content or information repeated a number of times. In data redundancy, the same field is repeated in two or more tables [11].

4.2.5. Examining the Spreadsheet

Spreadsheet software like Excel is capable of storing and manipulating tables of data. In the spreadsheet named “University”, carefully notice that all of the relevant information for each student is included, so we do have completeness. When and why should we use a spreadsheet for our data and not bother with using a DBMS? This task explores this question. For this particular task, the user should be using the Excel file named "University.xls." The Excel file named "University" contains all the related data of the student, courses enrolled, grades, and also information about the faculty. This file is posted for downloading on the Blackboard site. Save the given Excel file, "University," from the Blackboard site.

Task 1 Questions

i. What is the difference between a spreadsheet and a DBMS? Explain based on your understanding.

ii. Consider editing the Excel file "University." Also, see if there are any redundancies in the spreadsheet. If you are asked to perform operations on the Excel file, such as adding, deleting, and modifying data, describe how redundancy would lead to problems, inconveniences, or unnecessary computer operations.

Figure 4, shows the mapping between "Task1-Examining the Spreadsheet" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 1.

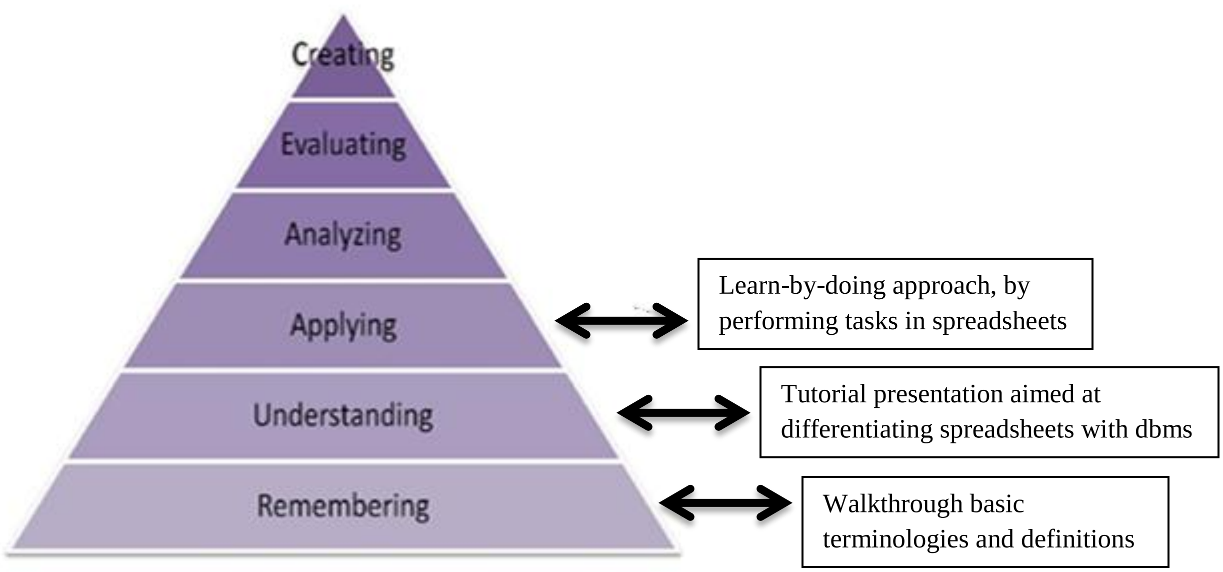


Figure 4. Revised Taxonomy Levels for Examining the Spreadsheet.

Table 1. Revised Taxonomy Levels for Examining the Spreadsheet.

Taxonomy Level	**Description**
Remembering	This task is designed with an approach to provide a walkthrough to the users with all the basic terminologies, definitions, and concepts.
Understanding	The user is given a tutorial presentation aimed at differentiating spreadsheets from a database. The concepts are referenced using an example University data spreadsheet.
Applying	This task offers a "learn-by-doing approach" to the users. The user is given a task to perform operations on the Excel spreadsheet which include data addition, modification, data deletion, etc.

4.3. Task 2: Examining the Database

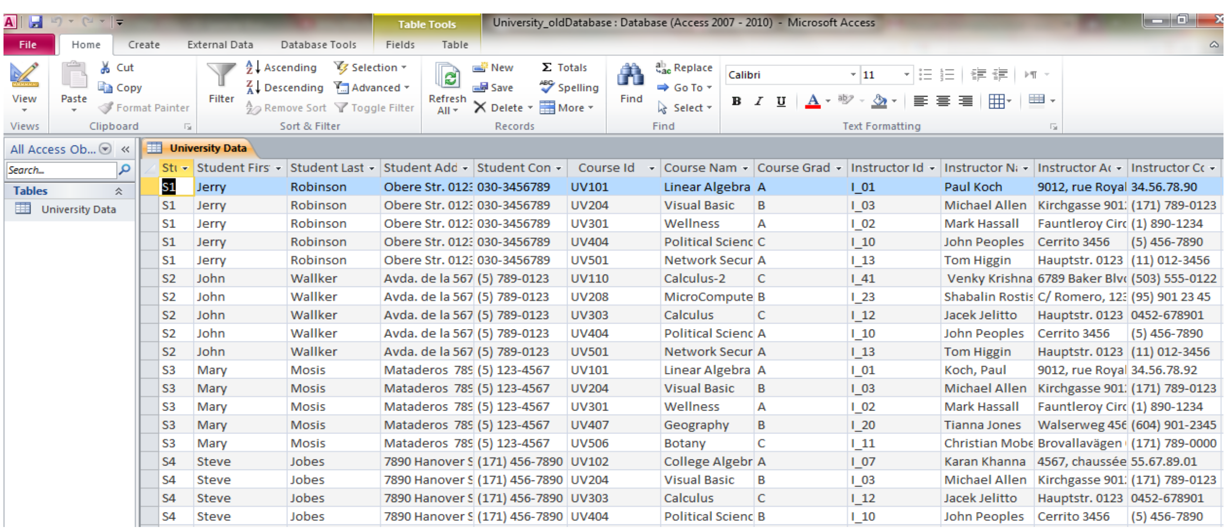
4.3.1. Why do we need to use a Database Management System?

So far in the first task, the user must have gained some knowledge about why we need to use a database rather than a flat file (Excel file). As we have seen, Figure 1 shows a flat file (Excel file) of information about nine students enrolled at a university where each student is registered in five different courses under five different faculty members. Those of us who are not quite aware of relational databases might try to use the spreadsheet, such as Microsoft Excel. For a start, this is not a bad approach because it provides all information in regards to the students enrolled at the university. There might be a scenario where there could be questions raised, such as [12]:

- How many students scored a grade of “A” in the “Linear Algebra” course?
- The course named “Botany” is taught by which faculty?
- What is the count of students enrolled in course Id “UV204”?

Also, for instance, at this point, the spreadsheet contains the records of only nine students, which is not a big count, but as the spreadsheet grows, it becomes difficult to answer these questions. In an attempt to present the university data in a comparatively better structure and platform where such complex questions could be handled, we need to use a database. Therefore, we create a database using Microsoft Access Database 2010.

Launch Microsoft Access, and open the database file named “University_oldDatabase”. The database is posted for downloading on the Blackboard site. Save the given database file, “University_oldDatabase,” from the Blackboard site. Using a right mouse click and choosing the option to “Save Target as” will allow you to save a file to your local machine or flash memory unit. There should be one table on the left pane titled “University Data.” For use in this task, there is this table named “University Data,” that stores all the information, as we can see in Figure 5.



Stu	Student Firs	Student Last	Student Add	Student Con	Course Id	Course Nam	Course Grad	Instructor Id	Instructor N	Instructor Ad	Instructor Co
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV101	Linear Algebra	A	I_01	Paul Koch	9012, rue Royal	34.56.78.90
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 901	(171) 789-0123
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Circ	(1) 890-1234
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV404	Political Scienc	C	I_10	John Peoples	Cerrito 3456	(5) 456-7890
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV501	Network Secur	A	I_13	Tom Higgin	Hauptstr. 0123	(11) 012-3456
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV110	Calculus-2	C	I_41	Venky Krishna	6789 Baker Blv	(503) 555-0122
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV208	MicroCompute	B	I_23	Shabalin Rostis	C/ Romero, 12	(95) 901 23 45
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV303	Calculus	C	I_12	Jacek Jelitto	Hauptstr. 0123	0452-678901
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV404	Political Scienc	A	I_10	John Peoples	Cerrito 3456	(5) 456-7890
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV501	Network Secur	A	I_13	Tom Higgin	Hauptstr. 0123	(11) 012-3456
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV101	Linear Algebra	A	I_01	Koch, Paul	9012, rue Royal	34.56.78.92
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 901	(171) 789-0123
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Circ	(1) 890-1234
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV407	Geography	B	I_20	Tianna Jones	Walserweg 45	(604) 901-2345
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV506	Botany	C	I_11	Christian Mobe	Brovallavägen	(171) 789-0000
S4	Steve	Jobes	7890 Hanover S	(171) 456-7890	UV102	College Algebr	A	I_07	Karan Khanna	4567, chaussée	55.67.89.01
S4	Steve	Jobes	7890 Hanover S	(171) 456-7890	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 901	(171) 789-0123
S4	Steve	Jobes	7890 Hanover S	(171) 456-7890	UV303	Calculus	C	I_12	Jacek Jelitto	Hauptstr. 0123	0452-678901
S4	Steve	Jobes	7890 Hanover S	(171) 456-7890	UV404	Political Scienc	B	I_10	John Peoples	Cerrito 3456	(5) 456-7890

Figure 5. Screenshot of University_oldDatabase.

Databases typically have many parts, and Access will display them in separate sub windows, all within a main window titled "Microsoft Access." If you cannot see the contents under each table, click on the title bar labeled "All Access Objects," and select the option "Tables and Related views."

Task 2 Questions

i. Write the names of the all the fields in the University Database.

ii. From the University_oldDatabase, indicate which fields have unique values across all the tuples in the table and which do not.

iii. Identify and write down the name of primary key of the "University Data" table.

Figure 6, shows the mapping between "Task 2-Examining the Database" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 2.

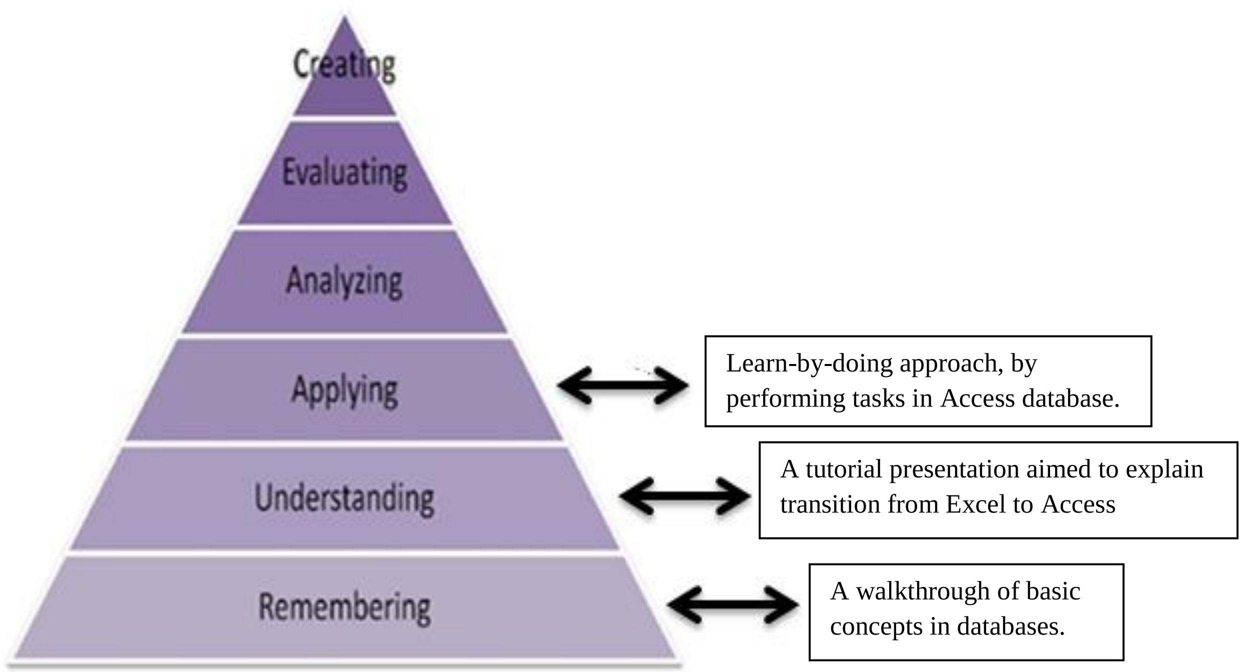


Figure 6. Revised Taxonomy Levels for Examining the Database.

Table 2. Revised Taxonomy Levels for Examining the Database.

Taxonomy Level	**Description**
Remembering	Task 2 offers the users more familiarity with the Access database. The users are given a tutorial representation of the rationale to work with a database as well as to simultaneously perform various database operations using Access.
Understanding	This task is designed to make the user understand the transition from Excel to Access. On the basis of a case study, an effort has been made to provide users with the understanding about the fundamentals of databases in Access as well as various database concepts such as fields, redundant values, primary key, etc.
Applying	Following an example-oriented approach, the user is given a task to detect the primary key in the table, which implies a user's knowledge of database concepts (primary key detection, unique values, etc.).

4.4. Task 3: Examining Database Design

4.4.1. What is a Relational Database Management System (RDBMS)?

A Relational Database Management System (RDBMS) is a type of Database Management System (DBMS) where data are stored in the tables as well as the relationships among the data being stored in the table [11].

4.4.2. What is Normalization?

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring that data dependencies make sense (only storing related data in a table) [13].

4.4.3. Why do we need to perform Normalization?

As we have seen in Task2, our database consists of redundant values. There exists a duplicate value in the attributes of the table. Therefore, in order to make the "University Data" table less redundant as well as to develop relationships between the fields of the University Data table, we need to split the University Data table into separate new tables. These tables would be generated by using the Normalization method [13].

4.4.4. Normalization Includes Normal Forms

4.4.4.1. First Normal Form

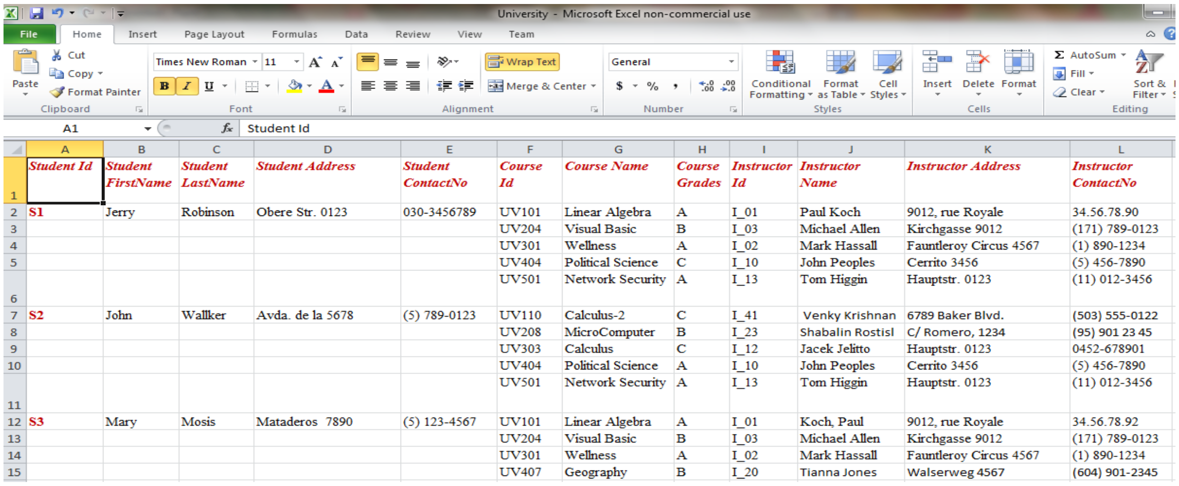
The First Normal Form abides by the following two rules [12]:

- A row of data in a table cannot contain repeating groups of similar data (atomicity).
- Each row of data in a table must have a unique identifier (or Primary Key).

The "University Data" table consists of two flaws:

4.4.4.1.1. Atomicity

As we have already seen and discussed, the "University Data" table consists of repeating values for Student id, Student first name, last name, address, and contact. Therefore, by separating each item in these lists into its own row, the condition for atomicity in the table is met. See Figure 7.

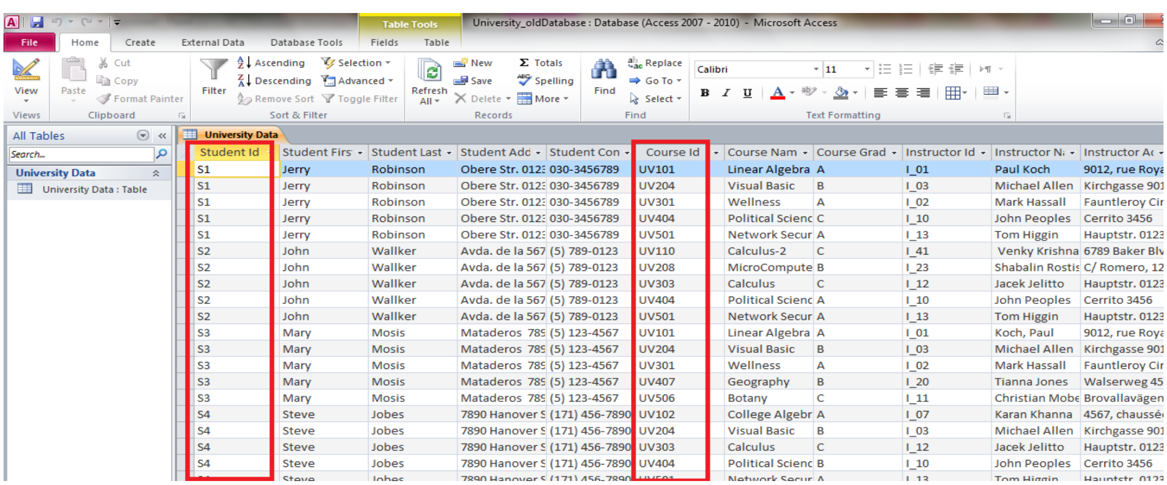


	A	B	C	D	E	F	G	H	I	J	K	L
1	Student Id	Student FirstName	Student LastName	Student Address	Student ContactNo	Course Id	Course Name	Course Grades	Instructor Id	Instructor Name	Instructor Address	Instructor ContactNo
2	S1	Jerry	Robinson	Obere Str. 0123	030-3456789	UV101	Linear Algebra	A	I_01	Paul Koch	9012, rue Royale	34.56.78.90
3						UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 9012	(171) 789-0123
4						UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Circus 4567	(1) 890-1234
5						UV404	Political Science	C	I_10	John Peoples	Cerrito 3456	(5) 456-7890
6						UV501	Network Security	A	I_13	Tom Higgin	Hauptstr. 0123	(11) 012-3456
7	S2	John	Wallker	Avda. de la 5678	(5) 789-0123	UV110	Calculus-2	C	I_41	Venky Krishnan	6789 Baker Blvd.	(503) 555-0122
8						UV208	MicroComputer	B	I_23	Shabalin Rostisl	C/ Romero, 1234	(95) 901 23 45
9						UV303	Calculus	C	I_12	Jacek Jelitto	Hauptstr. 0123	0452-678901
10						UV404	Political Science	A	I_10	John Peoples	Cerrito 3456	(5) 456-7890
11						UV501	Network Security	A	I_13	Tom Higgin	Hauptstr. 0123	(11) 012-3456
12	S3	Mary	Mosis	Mataderos 7890	(5) 123-4567	UV101	Linear Algebra	A	I_01	Koch, Paul	9012, rue Royale	34.56.78.92
13						UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 9012	(171) 789-0123
14						UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Circus 4567	(1) 890-1234
15						UV407	Geography	B	I_20	Tianna Jones	Walserweg 4567	(604) 901-2345

Figure 7. Screenshot of Un-Normalized University Data Table.

4.4.4.1.2. Primary Key

We need to provide a Primary Key for the University Data table. If we consider Student id as the Primary Key, it will violate the rules as each data must be unique, whereas Student id is not unique. Therefore, the Primary Key for the "University Data" table shall be a combination of two, i.e., Student id , Course id. If we consider (Student id, Course id) to be the Primary Key for the "University Data" table, then we can be sure that we do not have any repeating values as well as that all values are unique. See Figure 8.



Student Id	Student Firs	Student Last	Student Adc	Student Con	Course Id	Course Nam	Course Grad	Instructor Id	Instructor N	Instructor Ac
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV101	Linear Algebra	A	I_01	Paul Koch	9012, rue Roya
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 901
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Cir
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV404	Political Scienc	C	I_10	John Peoples	Cerrito 3456
S1	Jerry	Robinson	Obere Str. 012	030-3456789	UV501	Network Secur	A	I_13	Tom Higgin	Hauptstr. 012
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV110	Calculus-2	C	I_41	Venky Krishna	6789 Baker Bl
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV208	MicroCompute	B	I_23	Shabalin Rostis	C/ Romero, 12
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV303	Calculus	C	I_12	Jacek Jelitto	Hauptstr. 012
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV404	Political Scienc	A	I_10	John Peoples	Cerrito 3456
S2	John	Wallker	Avda. de la 567	(5) 789-0123	UV501	Network Secur	A	I_13	Tom Higgin	Hauptstr. 012
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV101	Linear Algebra	A	I_01	Koch, Paul	9012, rue Roya
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 901
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV301	Wellness	A	I_02	Mark Hassall	Fauntleroy Cir
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV407	Geography	B	I_20	Tianna Jones	Walserweg 45
S3	Mary	Mosis	Mataderos 78	(5) 123-4567	UV506	Botany	C	I_11	Christian Mobe	Brovallavägen
S4	Steve	Jobes	7890 Hanover	(171) 456-7890	UV102	College Algebr	A	I_07	Karan Khanna	4567, chaussé
S4	Steve	Jobes	7890 Hanover	(171) 456-7890	UV204	Visual Basic	B	I_03	Michael Allen	Kirchgasse 901
S4	Steve	Jobes	7890 Hanover	(171) 456-7890	UV303	Calculus	C	I_12	Jacek Jelitto	Hauptstr. 012
S4	Steve	Jobes	7890 Hanover	(171) 456-7890	UV404	Political Scienc	B	I_10	John Peoples	Cerrito 3456

Figure 8. Screenshot of Normalized University Data Table.

4.4.4.2. Second Normal Form

The Second Normal Form abides by the following two rules [12]:

- All requirements for First Normal Form are met.
- Every non-key attribute is fully functional dependent on the multi-valued Primary Key. If required, remove all partial dependency on a multi-valued key.

Next, we test the University Data table for partial dependencies on the multi-valued Primary Key. Testing for partial dependencies- means that, for a table that has a concatenated/multi-valued Primary Key (Student id, Course id), each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column only depends upon one part of the concatenated key, then we say that the entire table has failed the Second Normal Form, and we must create another table to rectify the failure. Let us see how the Second Normal Form works. Table 3, shows all the functional dependencies on the multi-valued Primary Key (Student id, Course id).

Table 3. Functional Dependencies on Primary Key.

Column Name	**Dependency on Primary Key**
Student first name	No (It depends only on StudentId.)
Student last name	No (It depends only on StudentId.)
Student Address	No (It depends only on StudentId.)
Student Contact	No (It depends only on StudentId.)
Course Name	No (It depends only on CourseId.)
Course Grades	Yes (Depends on both StudentId , CourseId.)
Instructor Id	No (It depends only on CourseId.)
Instructor Name	No (It depends only on CourseId.)
Instructor Address	No (It depends only on CourseId.)
Instructor Contact	No (It depends only on CourseId.)

Therefore, based on all the fields that depend upon the entire concatenated key, we have generated the following tables:

Student Table: {StudentId, StudentFirstName, StudentLastName, StudentAddress, StudentContact}

Course Table: {CourseId, Course name, InstructorId, InstructorName, InstructorAddress, InstructorContact}

Student Details Table: {StudentId, CourseId, CourseGrades}

Note: You can refer to the screenshots in Figures 9-11.

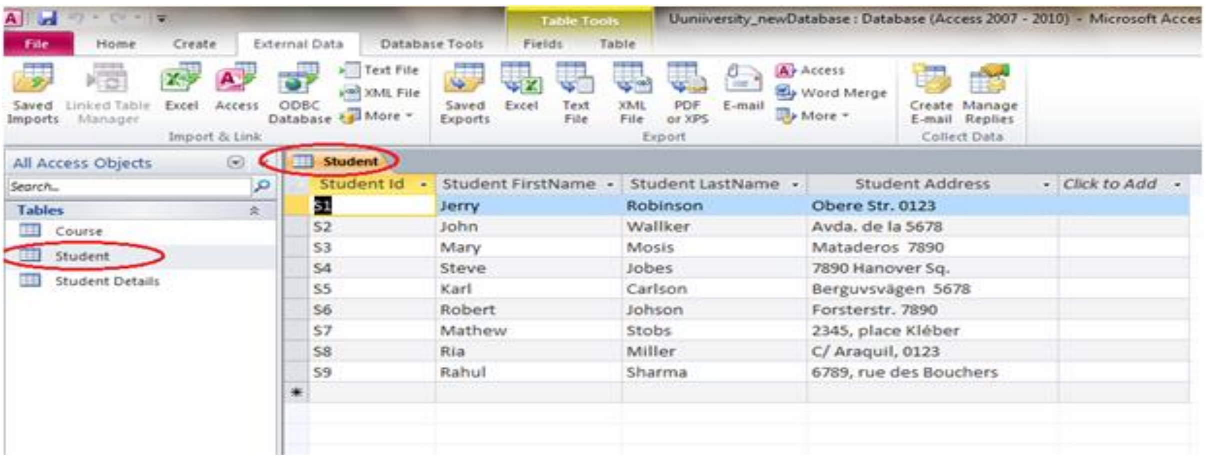


Figure 9. Screenshot of Student Table.

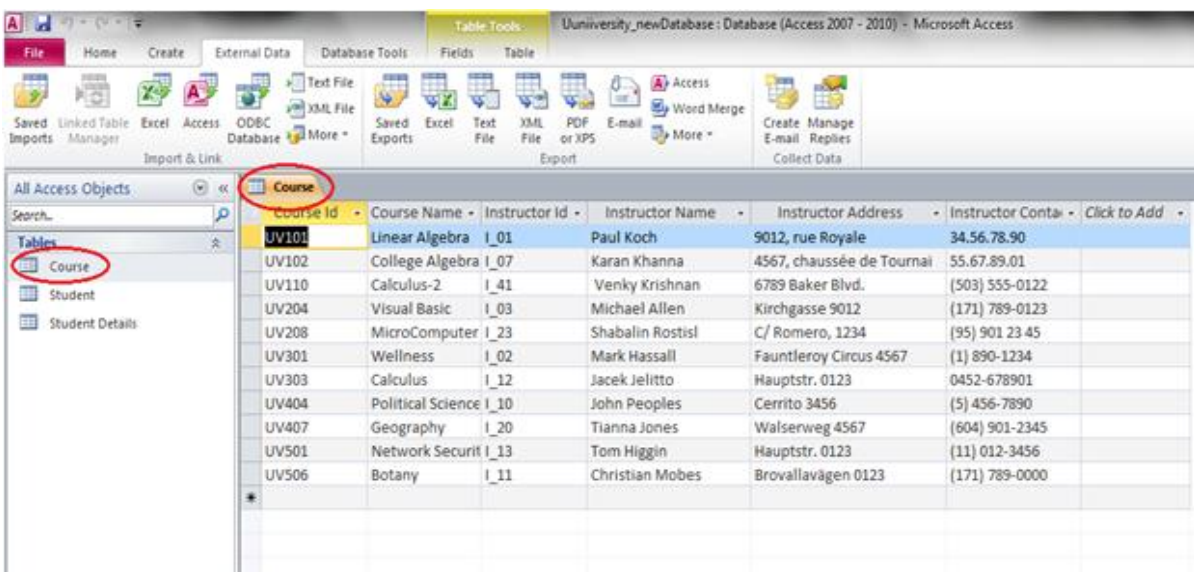


Figure 10. Screenshot of Course Table.

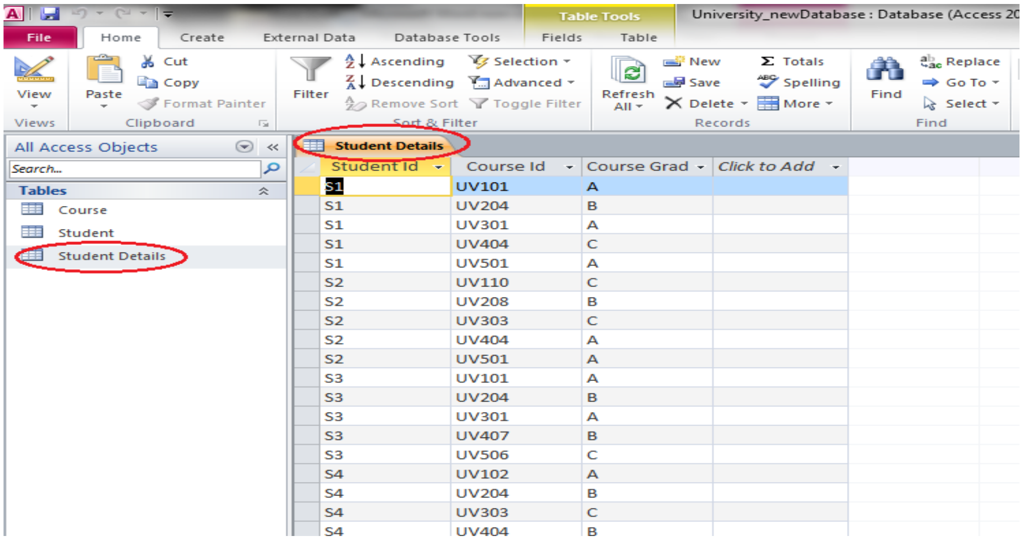


Figure 11. Screenshot of Student Details Table.

4.4.4.3. Third Normal Form

The Third Normal Form abides by the following two rules [12]:

- All requirements for the Second Normal Form must be met.
- Remove all transitive dependencies. Eliminate fields that do not depend directly on the primary key. That is, any field that is dependent not only on the primary key, but also on another field, must be moved to another table.

In the Course table, Course id is the Primary Key, but field attributes such as InstructorName, InstructorAddress, and InstructorContact are not only dependent on Course id, but also on Instructor Id. Therefore, according to the Third Normal Form, it is suggestible that we create a separate table named “Instructor” table to store all the instructor information; the Primary Key of that table would be Instructor Id. Figure 12, shows the screenshot of the “Instructor” table. Hence, we have the following four tables in our new database:

- Student Table
- Student Details Table

- Course Table
- Instructor Table

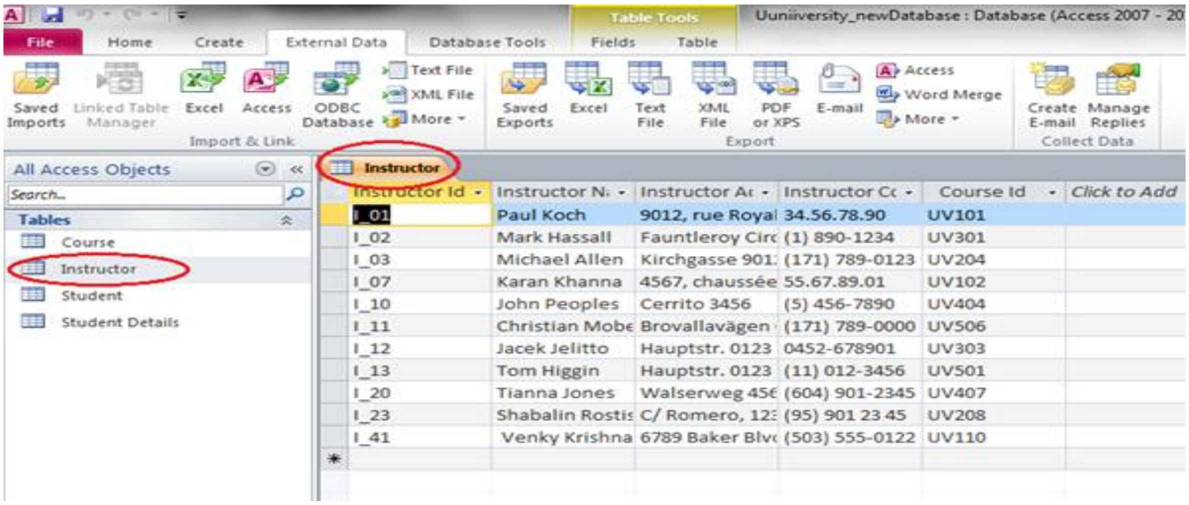


Figure 12. Screenshot of Instructor Table.

Although we reached the end of the Third Normal Form, normalization does not end here. Even after the database is normalized up to Third Normal Form, there is more to do. While undergoing the rules of the Third Normal Form, we ensure that all fields in the table are solely and entirely dependent on the primary key. When there is more than one candidate key within a given table, data inconsistencies may occur even though the database is in Third Normal Form. A candidate key is nothing but a combination of unique attributes from the table. A table may have one or more candidate keys, and one among them is chosen to be the primary key. In such a scenario, Third Normal Form does not work adequately. Therefore, we need to apply more measures to ensure that the database is normalized to the maximum.

In such a scenario, we bring higher levels of normal forms into implementation: Boyce Cord, Fourth Normal Form, and Fifth Normal Form. Although as we discovered that there is more to explore in the field of normalization, the scope of this tutorial is limited to normalization up to the Third Normal Form.

4.4.5. Examining the Database Design

Open the database file named “University_newDatabase.” There should be four tables on the left pane titled “Student Table,” ”Course Table,” ”Instructor Table,” and “StudentDetails Table.” So far, we have seen how we can remove redundant data from a given table by normalizing the table and extracting separate, new tables from the old table. These new tables are less redundant, more efficient, and related to each other.

Task 3 Questions

i. Why do we need to normalize a database?

ii. Differentiate between the following:

- First Normal Form and Second Normal Form
- Second Normal Form and Third Normal Form

iii. Normalize the following table named “Orders” table up to the three Normal Forms (First Normal Form, Second Normal Form, and Third Normal Form).

Table 4. Orders Table.

Order ID	Order Date	Cust. Id	Cust. Name	Cust. Address	Item Id	Item Description	Item Qty	Item Price
101	3/4/10	34	Adolphi, Stephan	2345 Gilbert St.	123	Soft drinks	100	12
103	3/18/10	67	Parovszky, Alfons	1234 Oxford Rd.	345	Cheeses	12	13
104	4/4/10	56	Balázs, Erzsébet	Calle del Rosal	123	Soft drinks	50	12
105	4/15/10	12	Holm, Michael	8901 Setsuko	645	Dried fruits	100	32
110	5/5/10	33	Popkova, Darya	5678 Rose St.	123	Soft drinks	24	12
110	5/5/10	33	Popkova, Darya	5678 Rose St.	345	Cheeses	55	13
125	5/15/10	21	Ræbild, Jesper	9012 King's Way	879	Prepared meats	50	11
125	5/15/10	21	Ræbild, Jesper	9012 King's Way	254	Breads, pasta	120	16

Figure 13, shows the mapping between "Task 3-Examining Database Design" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 5.

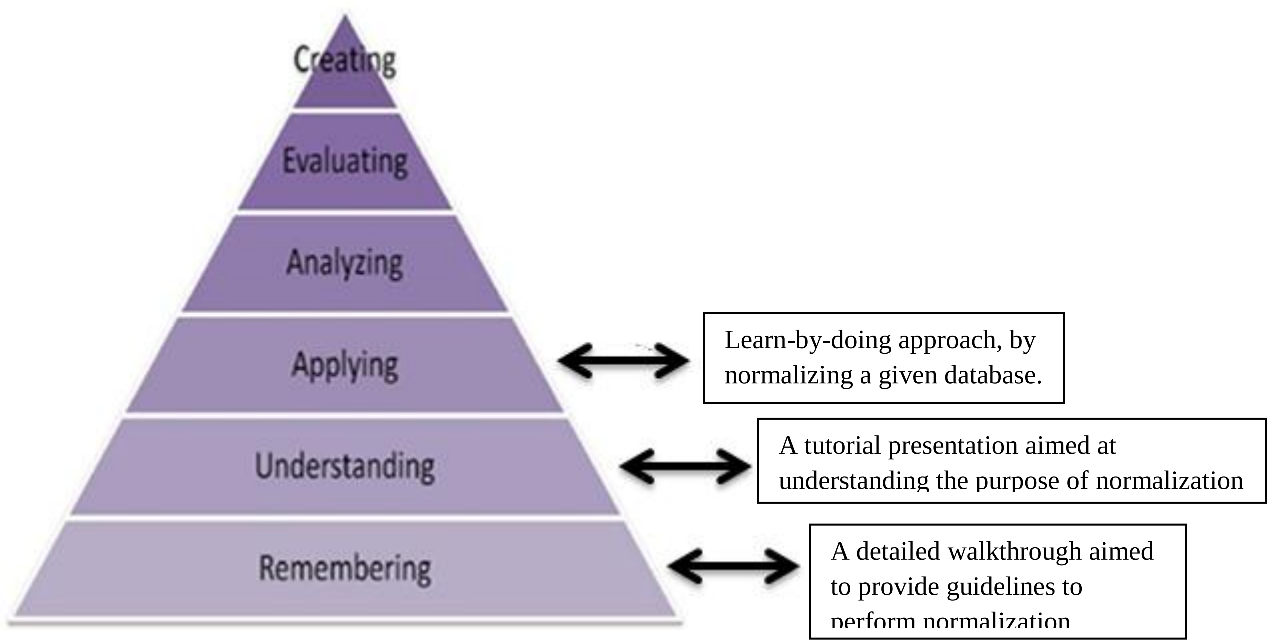


Figure 13. Revised Taxonomy Levels for Examining Database Design.

Table 5. Revised Taxonomy Levels for Examining Database Design.

Taxonomy Level	**Description**
Remembering	The user is provided with a detailed explanation as well as the rules to normalize a database with an example database.
Understanding	This task is designed with an approach to offer the user a tutorial presentation aimed at understanding the purpose of normalization as well as the procedure to perform normalization.
Applying	The task aims at teaching normalization by allowing the user to perform tasks using the three Normal Forms.

4.5. Task 4: Examining the Relationship Model

4.5.1. Examining Access Table View/Relationship Model

Access provides two kinds of view for a table. They are the "Datasheet View" and the "Design View" [9]. "Datasheet View" shows you both the information stored in the table as well as some metadata information (or database schema) which refers to the structure of the information. "Design View" is specialized for working with the metadata. A direct way to see the "Design View" is to right click the table title on the left panel and then choose the "Design View" option. "Design View" is focused on metadata [9]. "Design View" has rows and columns, but this time, each row contains information about a field. For now, focus on examining the table design without making changes to table. In this view, we see that, in addition to a name, each field has a data type which specifies what kind of information the field's value must be (e.g., a number, some text, a currency amount, etc.)

4.5.2. What is an Entity-Relationship Model?

The Entity-Relationship model (ER Model) allows us to describe the data involved in the real-world enterprise in terms of objects and their relationships. The model is widely used to develop an initial database design [9].

4.5.3. What is a One-to-One Relationship?

In a one-to-one relationship model, there exists one record in the first table that corresponds to exactly one record in the related table. In a relational database, a one-to-one relationship is enforced by a foreign key/primary key relationship [9].

4.5.4. What is a One-to-Many Relationship?

In a one-to-many relationship, each row in the related table can be related to many rows in the relating table. In a relational database, a one-to-many relationship is enforced by a foreign key/primary key relationship [9].

4.5.5. What is a Many-to-One Relationship?

In a many-to-one relationship, one entity (typically a column or set of columns) contains values that refer to another entity (a column or set of columns) that has unique values. In a relational database, a many-to-one relationship is enforced by a foreign key/primary key relationship [9].

4.5.6. What is a Many-to-Many Relationship?

In a many-to-many relationship, each record in one table may or may not relate to multiple records in the other table. In a relational database, a many-to-one relationship is enforced by a foreign key/primary key relationship [9].

Under the Database Tools tab, choose the Relationship option. The new sub window is an illustration of what is called an entity-relationship diagram. Also, in the Navigation Pane under the "All Tables" section, in the "Unrelated objects" category, you will find a Relationship Model. Open that Relationship Model, and examine it. If you observe carefully, you see "Student id" is a common field in both tables, Student and Student Details, so it allows each tuple in the Student table to be associated with a tuple in Student Details table. Similarly, there are common fields between other tables as well. Figure 14 is a relationship model for the "University_new Database".

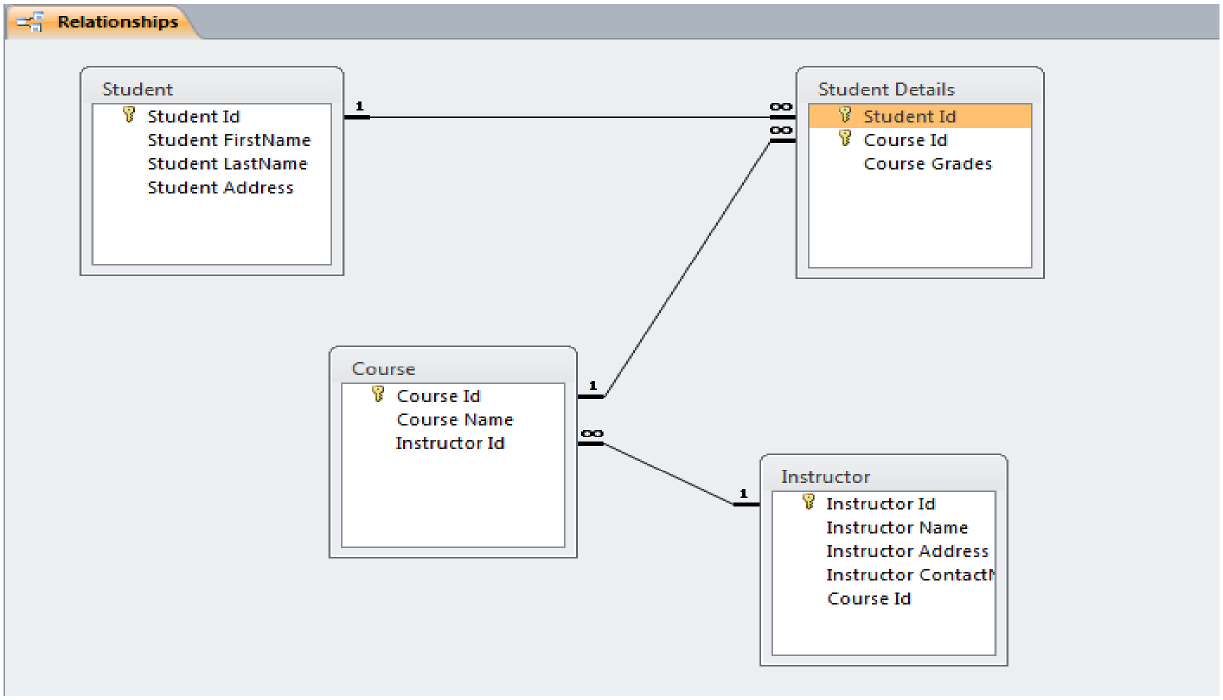


Figure 14. Relationship Model.

Task 4 Questions

i. We have already discussed that “Student id” is the common field in both the Student and Student Details tables. Explore the other common fields in the entity-relationship diagram; mention the names of the common fields as well as the tables that contain them.

ii. Referring to the entity-relationship diagram, by examining the titles and contents, explain what each small window represents.

iii. Each small window shows a list of fields. How is the Primary Key distinguished in these lists?

iv. Explain which fields from which tables are connected by black lines. Explain precisely what the 1 and infinity symbols mean in terms of connecting the information.

Figure 15, shows the mapping between "Task 4-Examining the Relationship Model" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 6.

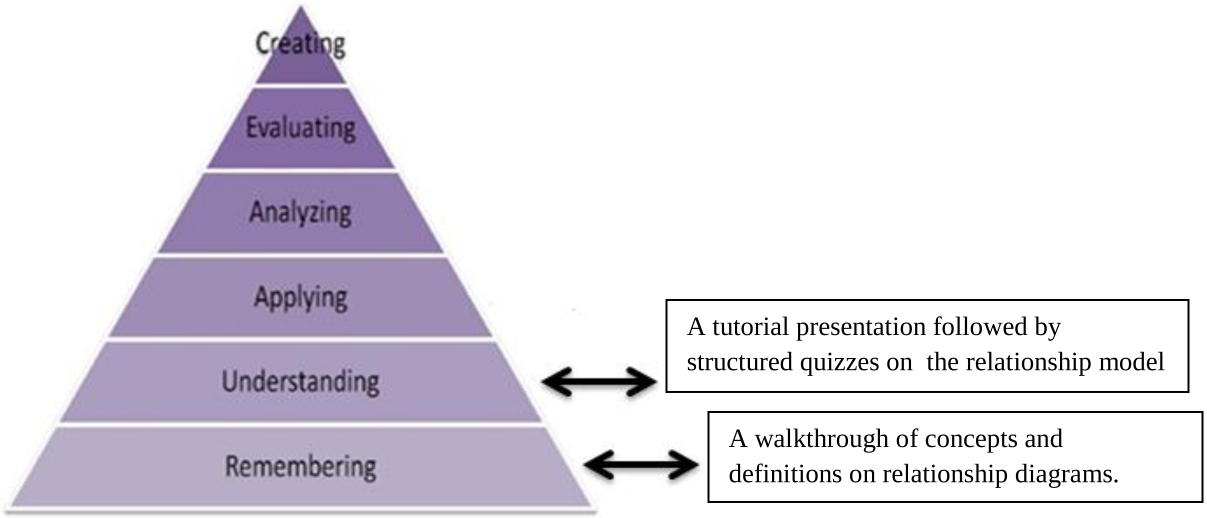


Figure 15. Revised Taxonomy Levels for Examining the Relationship Model.

Table 6. Revised Taxonomy Levels for Examining the Relationship Model.

Taxonomy Level	**Description**
Remembering	The user is provided with a detailed walkthrough of basic concepts and definitions based on relationship diagrams. This task is designed to offer user familiarity with creating relationship diagrams and all the different components included in a relationship model, with the help of a sample relationship model.
Understanding	A tutorial presentation followed by structured quizzes, where the user answers questions based on relationship diagrams. The tasks consist of exercises designed in order to test the user's understanding and affluence towards creating relationship diagrams.

4.6. Task 5: Examining an Existing Query

Queries, like tables, can be viewed in Datasheet or Design View. In the navigation pane on the left, the stored queries can be viewed [9]. Now, open the "Student_Course_Geography" query by double clicking on it. You can see this query under three tables, i.e., Student, StudentDetails, and Course Tables. Figure 16 shows a screenshot of "Student_Course_Geography" query in the "Design View".

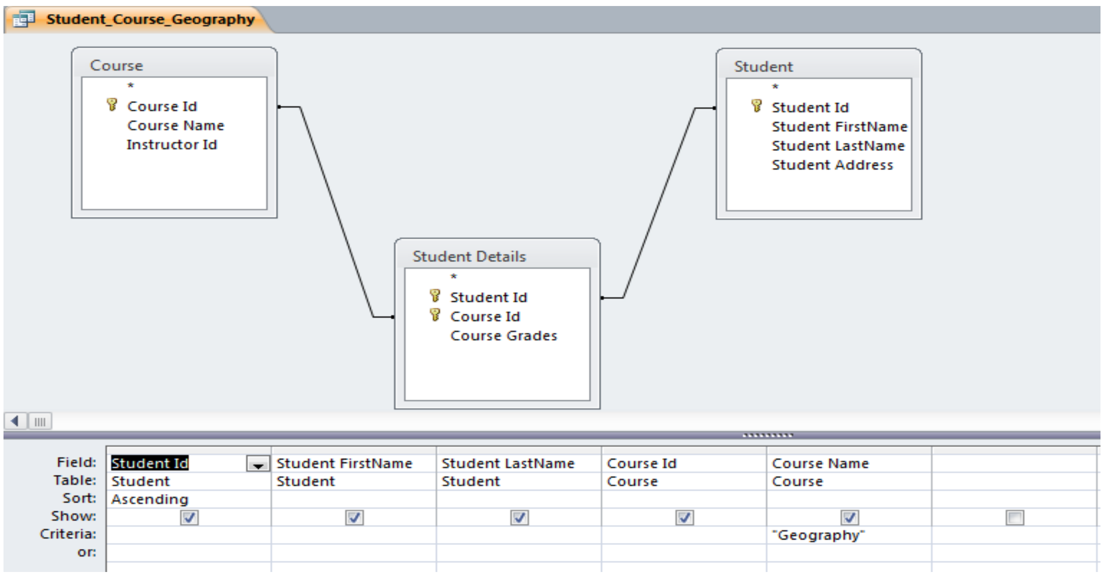


Figure 16. Student_Course_Geography Query.

Task 5 Questions

i. Open the "Student_Course_Geography" query by double clicking on it. Write the names of the fields shown in the columns. In what order do the tuples appear?

ii. Open the "Design View" of the "Student_Course_Geography" query by selecting the option of "Design View." (Right click on the tab, and you can see the various options.) In "Design View", you can see two parts; the upper one has two small windows related to each other, and the lower one has a grid. For this query, write down the field which has the "Criteria" option filled in the lower part.

Figure 17, shows the mapping between "Task 5-Examining an Existing Query" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 7.

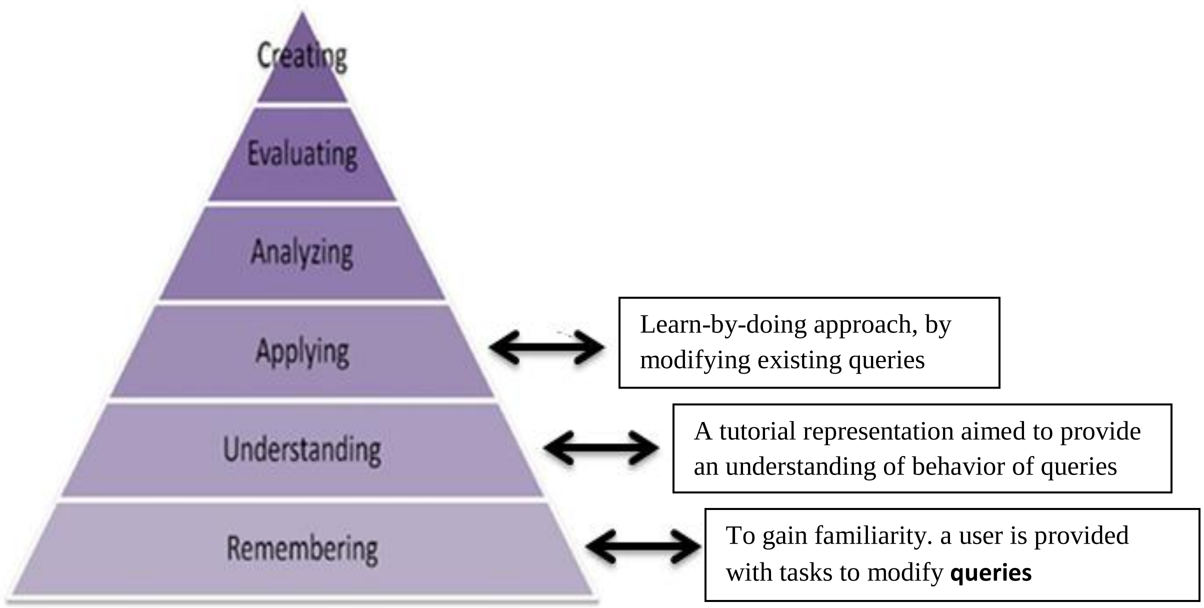


Figure 17. Revised Taxonomy Levels for Examining an Existing Query.

Table 7. Revised Taxonomy Levels for Examining an Existing Query.

Taxonomy Level	**Description**
Remembering	The user is given tasks to perform modifications in the pre-existing queries, thereby making an effort for the user to gain more acquaintance with queries.
Understanding	This task provides a tutorial representation aimed at offering a deeper understanding of the behavior of queries to the user.
Applying	"Learn-by-doing" approach is followed in this task, where the users are asked to modify already-existing queries.

4.7. Task 6: Creating a New Query

Until now, we worked with queries that already existed. In the next steps, you will create a new, one-table query for viewing information. Your goal is to generate a query based on a single table to show some information. Figure 18, shows a screenshot of the "Student_Grades" query in the "Design View".

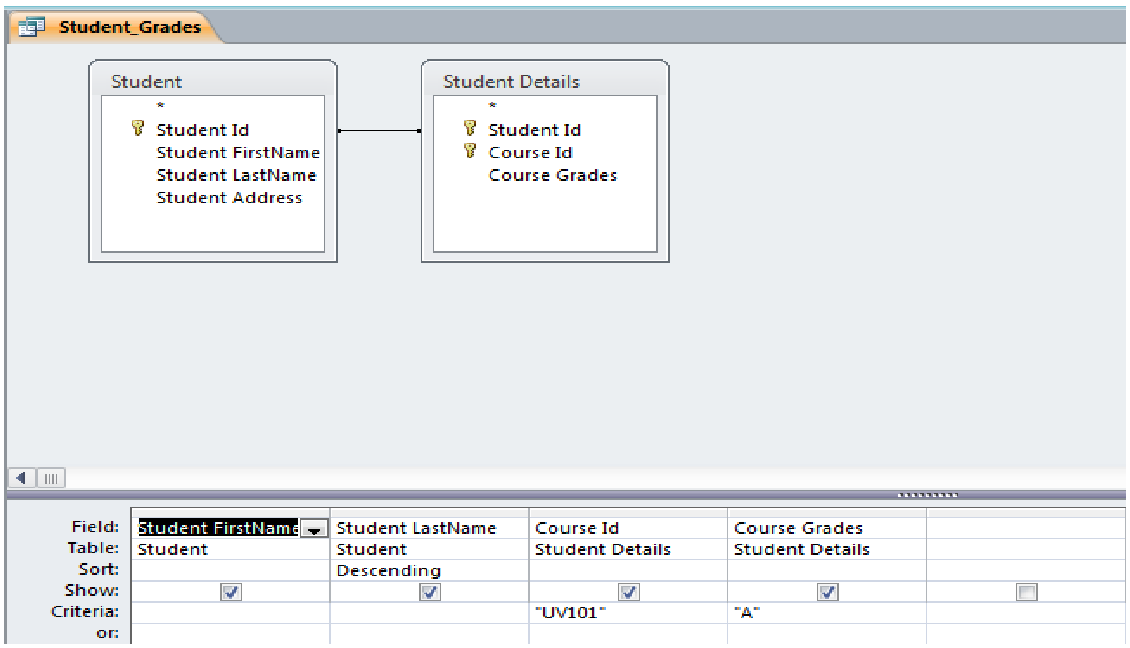


Figure 18. Student_Grades Query.

Task 6 Questions

i. Create a query to show the Student ID, last name, and first name of all the students whose last name starts with the letter "S." You can save this query as" Student_Lastname" query. Paste a screenshot of the "Design View" and "Datasheet View" of the generated query.

ii. Create a query to show the Student Id, StudentFirstName, and StudentLastName for all the students who are enrolled in the CourseID= "UV101." You can save this query as "Student_CourseID" query. Paste a screenshot of the "Design View" and "Datasheet View.

Figure 19, shows the mapping between “Task 6- Creating a New Query” and the levels of the Revised Bloom’s Taxonomy. A detailed explanation of the mapping is provided in Table 8.

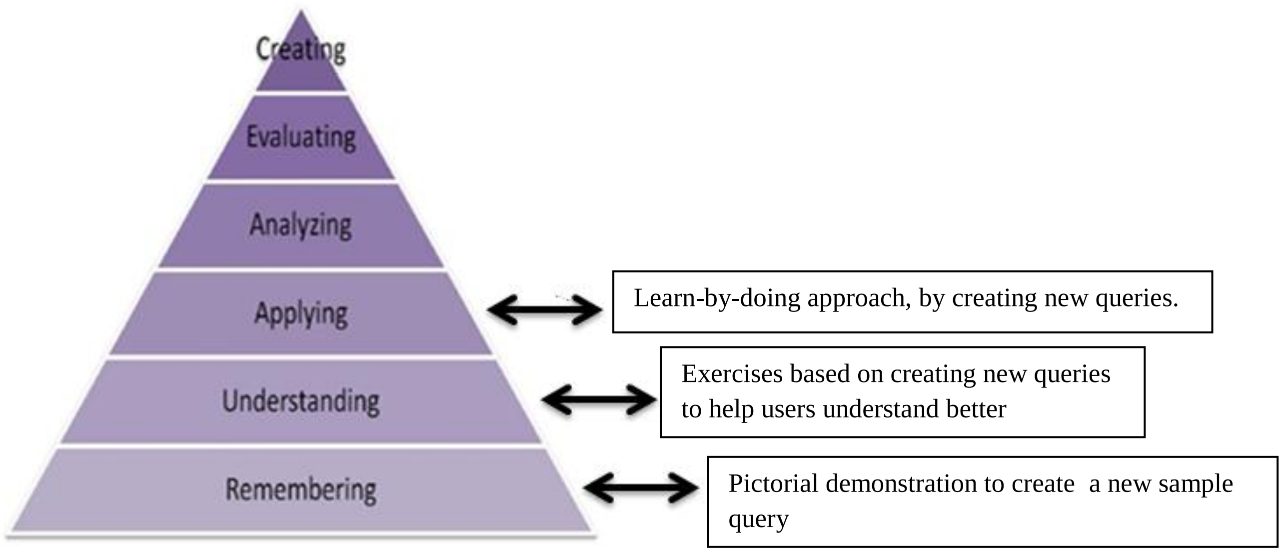


Figure 19. Revised Taxonomy Levels for Creating a New Query.

Table 8. Revised Taxonomy Levels for Creating a New Query.

Taxonomy Level	**Description**
Remembering	The user is given a detailed procedure and a pictorial demonstration to create a new sample query.
Understanding	The user is exposed to exercises to create queries for this task. Creating a new query, would help the user better understand the entire mechanism of generating queries.
Applying	Again the “learn-by-doing approach” is followed here. The user is given instructions and procedure to direct him/her to create a query and is asked to create a new query following the given set of instructions.

4.8. Task 7: Creating a Form

A form is like a table, but it allows you to be flexible with database modifications, such as add, update, and delete, within MS Access without using additional software. Also, when something is changed in a form, it is automatically reflected in the table [9]. The question that arises is as follows: why do we need forms when we already have the table? Particularly when there are many fields in the table, separate forms can be created for each entry in the table, providing an easy way to view the data. Also, with the help of forms, you can change one record at a time and update the existing table automatically. Figure 20, shows a screenshot of a form.

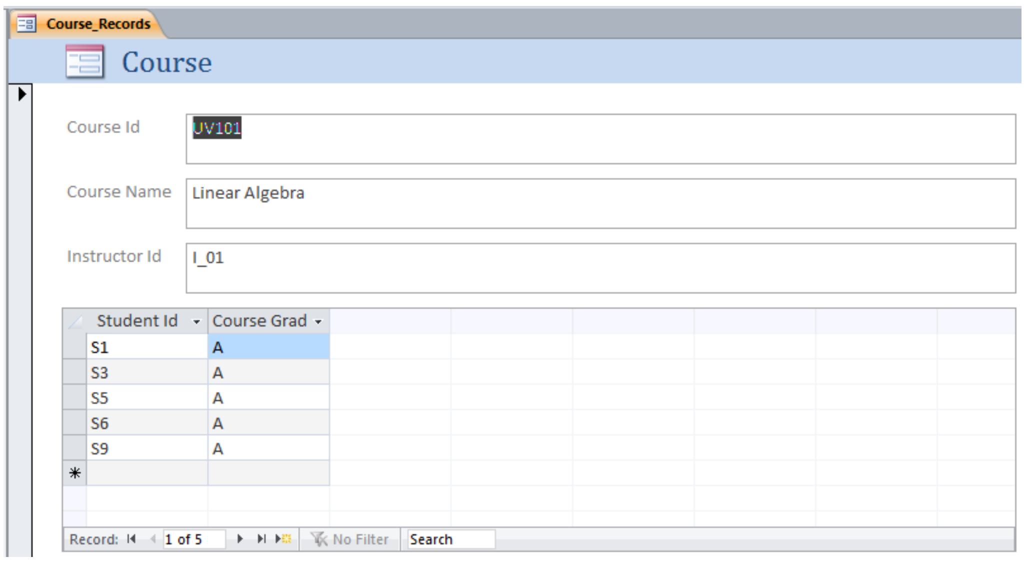


Figure 20. Course Records Form.

Task 7 Questions

i. Open the "Instructor_Records" (form) under the "Instructor" table on the left navigation pane. The small table in each record indicates the record's relationship with the fields in another table, so consider the record number of the main window where it says "1 of 10." For accessing the next record, simply click the arrow until you reach the desired record. Now in record 1, change the Instructor Id from "I_01" to "F_01,"and in record 3, change

InstructorContact from “(171) 789-0123” to “(171) 789-0000”; save the changes. You can now see the InstructorId and the InstructorContact changed in the main table and also for the respective records in the Instructor table. Create screenshots of the main table and the two changed records from the form, and paste them into your answer.

ii. Open “**Course_Records**” (form) under the “Course” table on the left navigation pane. You can see that this form contains 11 records. Now, carefully observe what you see in records 1 and 3. Create screenshots of records 1 and 3, and explain what the “Course_Records” form shows for records 1and 3.

Figure 21, shows the mapping between “Task 7- Creating a Form” and the levels of the Revised Bloom’s Taxonomy. A detailed explanation of the mapping is provided in Table 9.

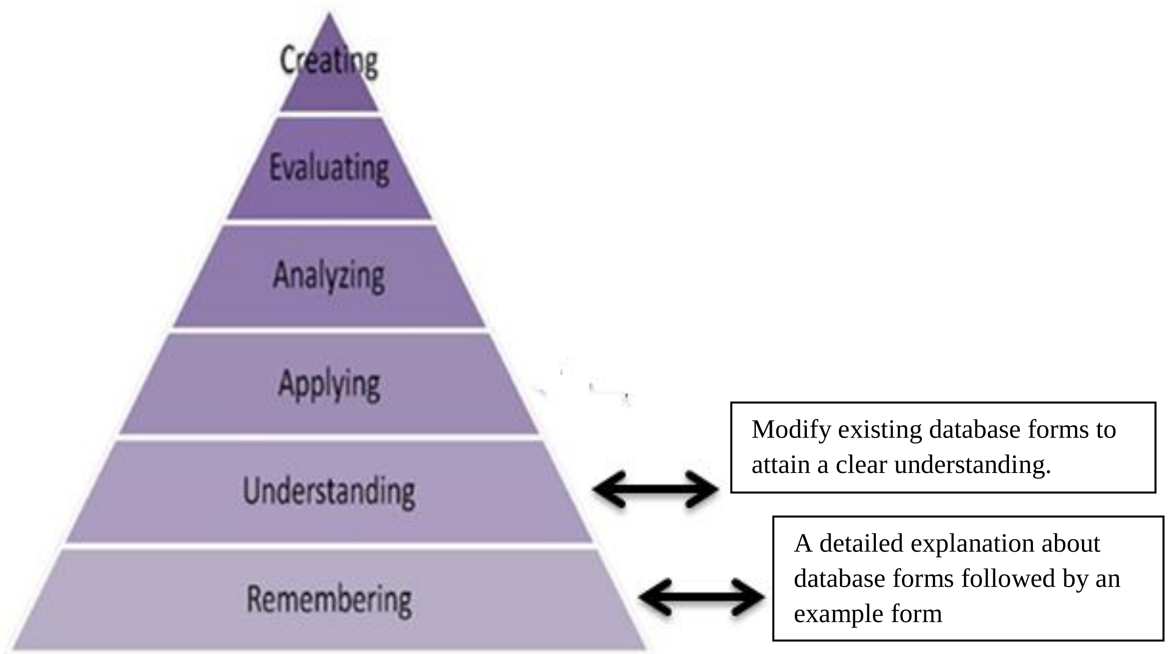


Figure 21. Revised Taxonomy Levels for Creating a Form.

Table 9. Revised Taxonomy Level for Creating a Form.

Taxonomy Level	**Description**
Remembering	In this task, the main emphasis is offering the users fluency to work with the forms in the databases by offering a detailed tutorial presentation followed by an example form.
Understanding	In order to attain a clear understanding of forms, the users are asked to modify existing forms and to create new records in the existing forms.

4.9. Task 8: Creating a Report

The report provides controls of the format for printing a database for use in presentations or other business purposes. Basically, reports provide control of the appearance of the information displayed [9].

Task 8 Questions

i. Create a report for the "Student" Table. To do this task, select the table, and use the Create tab to select the Report option. Under Auto format, select the first style. You can view the table in a printable order with date and time information. Create a screenshot of the report, and paste it into your answer.

ii. Create a report for the "Instructor" Table. To do this task, follow the same procedure you performed in the previous question. Also, create a screenshot of the report, and paste it into your answer.

Figure 22, shows the mapping between "Task 8- Creating a Report" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 10.

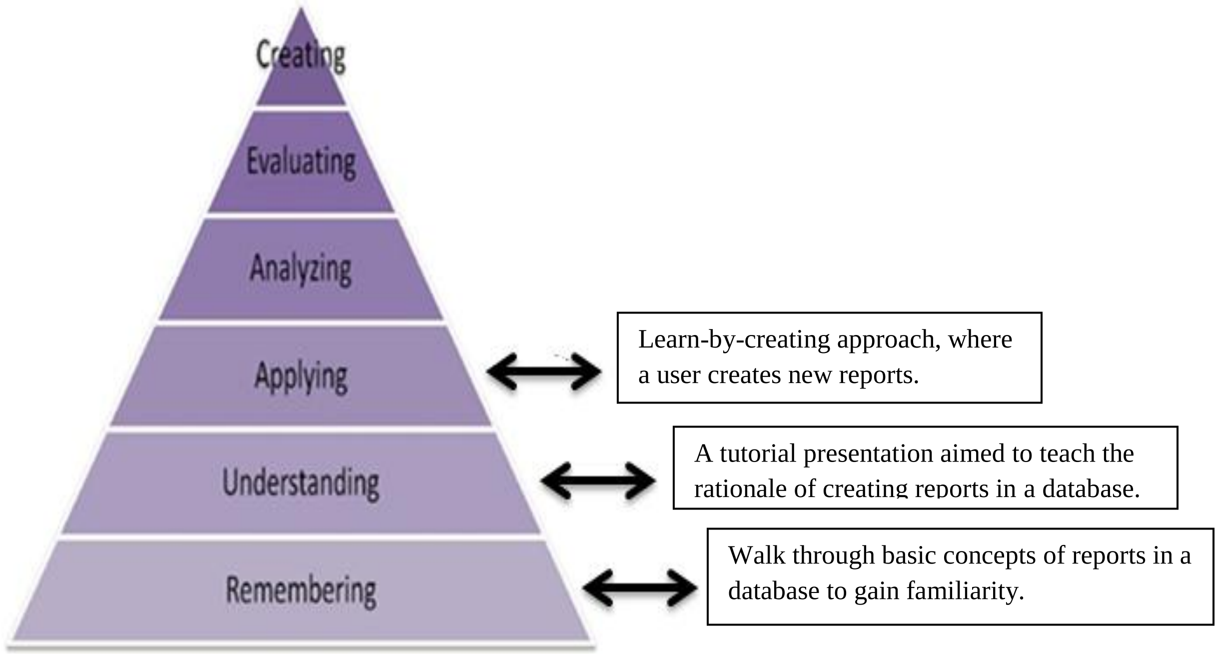


Figure 22. Revised Taxonomy Levels for Creating a Report.

Table 10. Revised Taxonomy Levels for Creating a Report.

Taxonomy Level	**Description**
Remembering	Walkthrough basic concepts of reports in a database to gain knowledge in generating reports and to remember the use of creating reports.
Understanding	This task offers a tutorial presentation to teach the rationale of creating reports in a database.
Applying	A “learn-by-creating” approach is followed in this task, where the user is asked to create a new report following the instructions provided.

4.10. Summary

So far, we have illustrated each of the eight tasks with different levels of the Revised Bloom's Taxonomy. This illustration is the basis of the mapping between the tasks and the levels of the Revised Bloom's Taxonomy (Cognitive Domain). You shall notice that, for certain tasks, a few of the taxonomy levels are missing. The main reasons are as follows:

i. For a particular task assigned to the user, the exercises performed by the user, map well, only until the "Applying level" in the Revised Blooms Taxonomy.

ii. The Revised Bloom's Taxonomy mapped here in the eight tasks are not restrained to the tasks; rather, they are illustrated, keeping in mind a much broader perspective that covers the user's knowledge and understanding of hands-on work with databases.

CHAPTER 5. SQL DATABASE: CONCEPTS AND COMMANDS

5.1. Introduction

As we near the end of the first tutorial, users may be thinking "Microsoft Access seems to have some great features, why would I want to use any other database package?". If you have heard of the SQL Server, in particular, you may be wondering what the difference is between Microsoft Access and an SQL Server, and what one should prefer to use. Now that you have learned how to use MS Access, there are some very good reasons why you would not use it in some cases. Here is a quick run through the use of Microsoft Access versus an SQL Server [14].

- Access is more suited for desktop use with a small number of users accessing it simultaneously.
- An SQL Server is a more robust database management system. It is designed to have many hundreds, or even thousands, of users accessing it at any point in time. Microsoft Access, on the other hand, does not handle this type of load very well.
- On a broader prospective of database development, an SQL Server is perfectly suited for database-driven websites. MS Access is limited to stand-alone systems and smaller-scale projects. Using Access for a database-driven website would, absolutely, be a bad idea.
- An SQL Server also contains some advanced database administration tools that enable organizations to schedule tasks, receive alerts, optimize databases, configure security accounts/roles, transfer data between other disparate sources, and much more, unlike in MS Access [14].

Now that you have some hands-on experience with MS Access and you are quite aware of what SQL offers in contrast with MS Access, it is time to make a decision. In fact, the decision might have already been made for you. If you have just finished the entire MS Access tutorial, then you would already have MS Access on your systems, and you would probably prefer to use Access because you have experience working on it and because you might not have come across a stage where you need to work on large-scale development, such as creating a website. Based on what we have already discussed in regards to SQL, if you think you need to upgrade to an SQL Server, do not panic and worry thinking that you just wasted your time learning Access. In fact, you will definitely find that an SQL Server is similar in many ways to Access. Now that you know your way around Access, you should be able to relate it to an SQL Server.

Download the "Project 2" zip folder. Extract all the files from the folder. Once you extract the files, you can see four files in the folder. The first file is the "SQL Server Set Up" document; this document provides a detailed pictorial description of how to set up the SQL Server 2008 R2 Express edition on your personal computers. Once you have SQL Server 2008 R2 Express installed, you should open the second file, "University Database Set Up" document, which provides a detailed description with images about how to set up the University database on the SQL Server. While following the steps in the "University Database Set Up" document, you would need to use the Excel file "SQL_University," the third file in the extracted folder. The fourth and last file in the extracted folder is the assignment file. Once you open this file, you will find a lot of information to read through. You can attempt the questions at the end of each task.

The main objective of this chapter includes an introduction of the SQL Server to the users. So far, the users have learned the basics of working with databases using flat files and then working with an Access database. In this step, users step forward and extend their knowledge of databases from small-scale, stand-alone projects performed using Access to a larger-scale scenario like a database used for website development. The tutorial for this phase of database development is designed keeping in mind both that the users understand the basics of database operations using an SQL Server. By the end of the tutorial, the user should be able to analyze and evaluate whether and when to use Access for database management or whether to use an SQL Server.

5.2. Task 1: Examining an SQL Database

5.2.1. What is an SQL Server?

SQL Server is a software product with a primary function to store data and retrieve data as requested by other software applications. This data retrieval can be on the same computer, or it can also run on another computer across a network [15].

Referring to SQL as a software product would not be a complete definition of it. An SQL Server is also defined as any Database Management System (DBMS) that can respond to queries from client machines [16]. The queries are written on the SQL Server using the computer language known as SQL.

5.2.2. What is a Query Language?

A query language is nothing but a computer language designed specifically in order to make a request to the database for retrieving a particular piece of information [10]. One such language that we are going to use for this assignment is Structured Query Language (SQL). .

5.2.3. What is a Relational Model?

The Relational Model is a simple model that uses the concept of relations between tables. The information is put into a database that consists of columns running up and down and rows that run from left to right. This database model is designed with emphasis to eliminate the duplication of data while ensuring that the data are accurate, valid, and consistent, e.g., RDBMS.

Task 1 Questions

i. The following items are mismatched. Match terms with the correct descriptions. The first one is done for your reference:

- Primary Key ⟷ ___Uniqueness of rows, disallows nulls____
- Query ⟷ Desktop use
- SQL Server ⟷ Based on Relational Model
- MS Access ⟷ Information request from a database
- RDBMS ⟷ Used in database-driven websites

Figure 23, shows the mapping between “Task 1- Examining an SQL Database” and the levels of the Revised Bloom’s Taxonomy. A detailed explanation of the mapping is provided in Table 11.

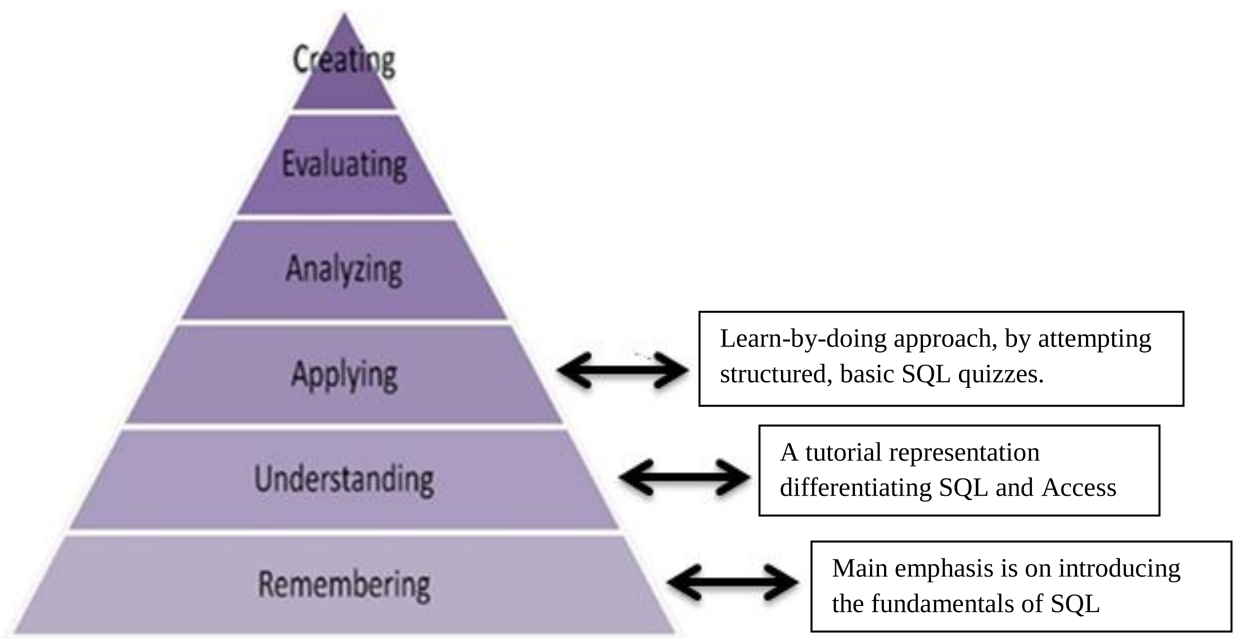


Figure 23. Revised Taxonomy Levels for Examining an SQL Database.

Table 11. Revised Taxonomy Levels for Examining an SQL Database.

Taxonomy Level	Description
Remembering	In this task, the main emphasis is introducing the fundamentals of Structured Query Language to the user.
Understanding	This task offers a tutorial presentation to provide the user with a clear understanding of using SQL in contrast to an Access database.
Applying	A "learn-by-doing" approach is followed in this task, where the user is provided with SQL fundamental-based exercises and structural quizzes to attempt and employ.

5.3. Task 2: Beginning with SQL

5.3.1. Example Database Tables

This tutorial requires using the Excel file "University_new" database that contains the following tables:

- Student Table
- Student Details Table
- Course Table
- Instructor Table

5.3.2. Creating a Database Query

The first query that you create when you start working with SQL is the query to create a database [11]. The syntax of the query is given below. Also Figure 24 shows a screenshot of the result of the query.

```
CREATE DATABASE [database_name];
```

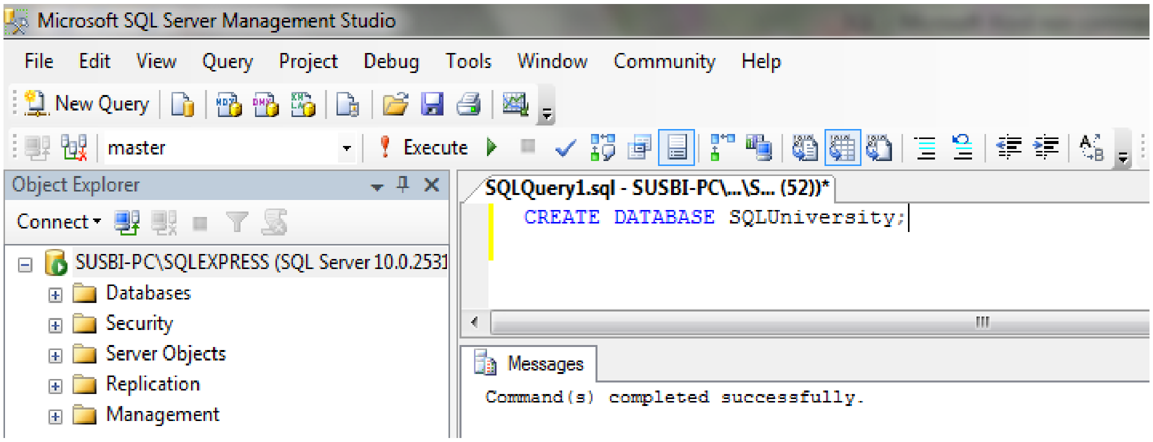


Figure 24. Screenshot of Creating a Database Query.

5.3.3. Creating a Table Query

The following query creates tables in an empty database to feed the data. The syntax of the query is given below [11]. Also Figure 25 shows the screenshot of the result of the query.

CREATE TABLE [table_name]

(column_name_1 INT NOT NULL,

column_name_2 VARCHAR (30) NOT NULL,

column_name_3 DATE NOT NULL

);

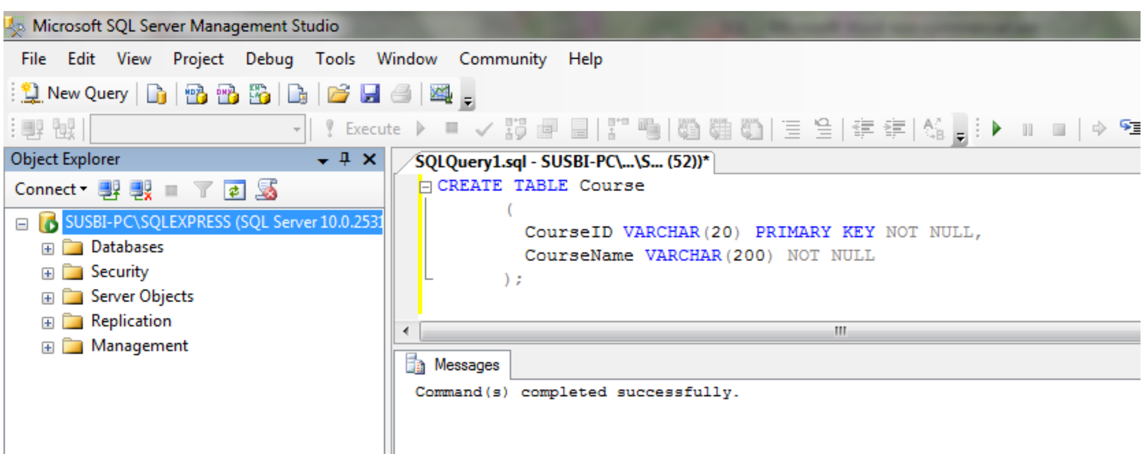


Figure 25. Screenshot of Creating a Table Query.

5.3.4. Select Statement

The purpose of the Select Statement is to query tables, to apply some logical manipulation, and to return a result [10]. The Select Statement is used to select specific data from the database. The syntax of the query is given below. Also Figure 26 shows the screenshot of the result of the query.

SELECT [column_name (s)]

FROM [table_name];

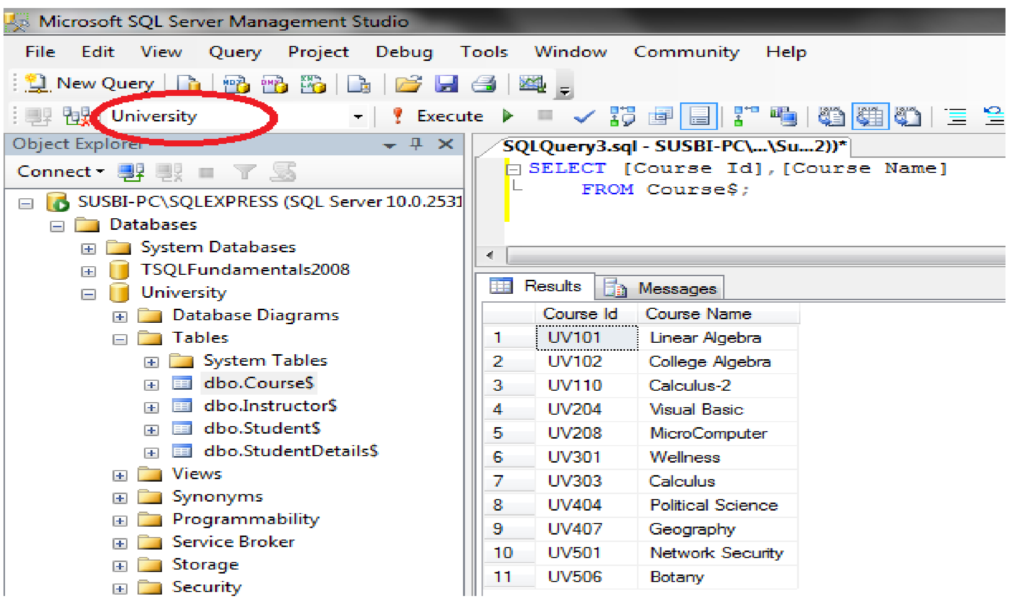


Figure 26. Screenshot of Select Statement.

5.3.5. Select Statement with a Where Clause

In a Select Statement, we use a Where Clause which performs the function of extracting only those records that fulfill a specified criterion. Here in the Where Clause, the operator value is nothing but the logical expression which is used to filter the rows returned by the From phase [11]. The syntax of the query is given below. Also Figure 27 shows the screenshot of the result of the query.

SELECT [column_name(s)]

FROM [table_name]

WHERE [column_name] = ‘operator value’;

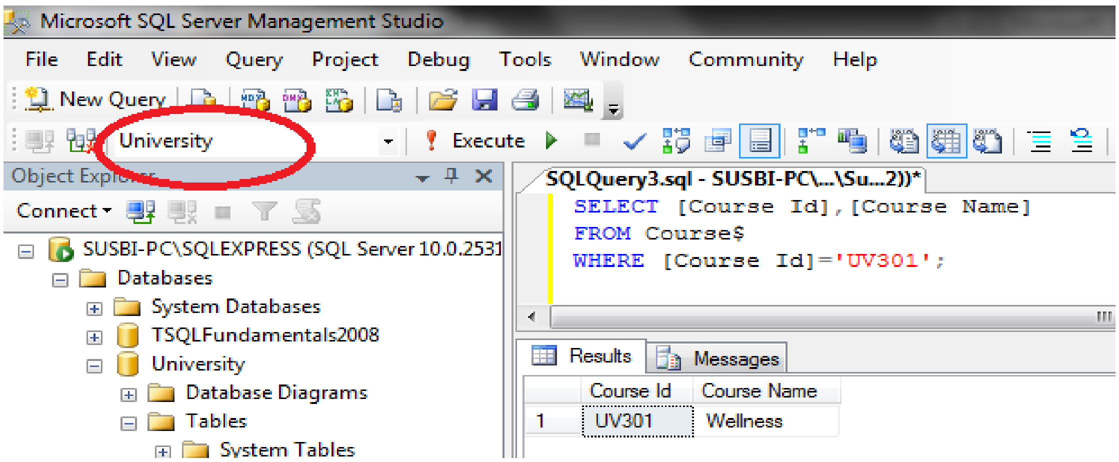


Figure 27. Screenshot of Select Statement with a Where Clause.

Task 2 Questions

i. Write a Select Statement from the Course Table to show the entire courses available (Course id, Course name) at the university. Create a screenshot of the result set (table generated from the query), and paste it in your solution document.

ii. Write a Select Statement to show all student Course grades from the Student Details table for students who are registered in Course id = “UV 10.” Create a screenshot of the result set, and paste it in your solution document.

Note: Include column names (Student id, Course id, and Grades). In the WHERE Clause, mention the operator values as Course id = “UV101.”

Figure 28, shows the mapping between “Task 2- Beginning with SQL” and the levels of the Revised Bloom’s Taxonomy. A detailed explanation of the mapping is provided in Table 12.

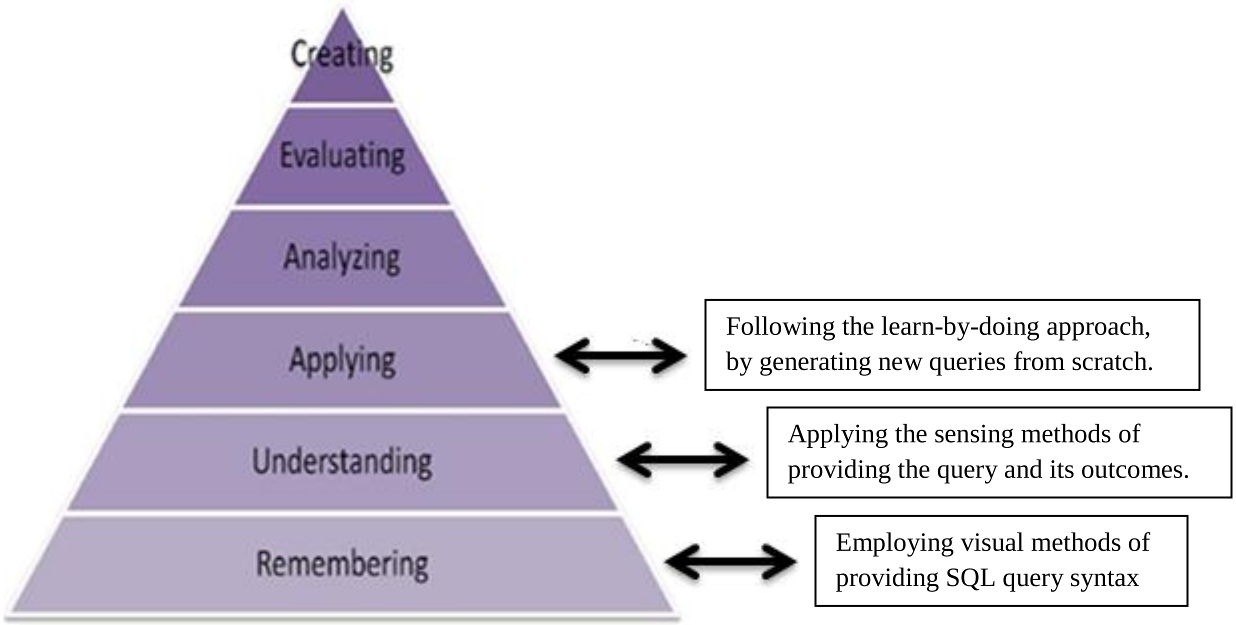


Figure 28. Revised Taxonomy Levels for Beginning with SQL.

Table 12. Revised Taxonomy Levels for Beginning with SQL.

Taxonomy Level	**Description**
Remembering	This task includes employing visual methods of providing SQL query syntax to help and guide the user to understand how to write a SQL query.
Understanding	Many users learn and understand concepts by following facts or established methods. Therefore, the users are given the query syntax to follow and examples to test their own understanding.
Applying	This task implements the "learn-by-doing" approach by generating new queries from scratch.

5.4. Task 3: SQL Query Statements with Clauses

5.4.1. SQL Query Statement with a Group By Clause

The Group By Clause is used in an SQL query to collect data across multiple records and to group the results by one or more columns. This Clause allows the user to arrange the rows returned by the logical query-processing phase in groups. The syntax of the query is given below [11]. Also Figure 29 shows the screenshot of the result of the query.

SELECT [column_name(s)]

FROM [table_name]

WHERE [column_name] = 'operator value'

GROUP BY [column 1], [column 2];

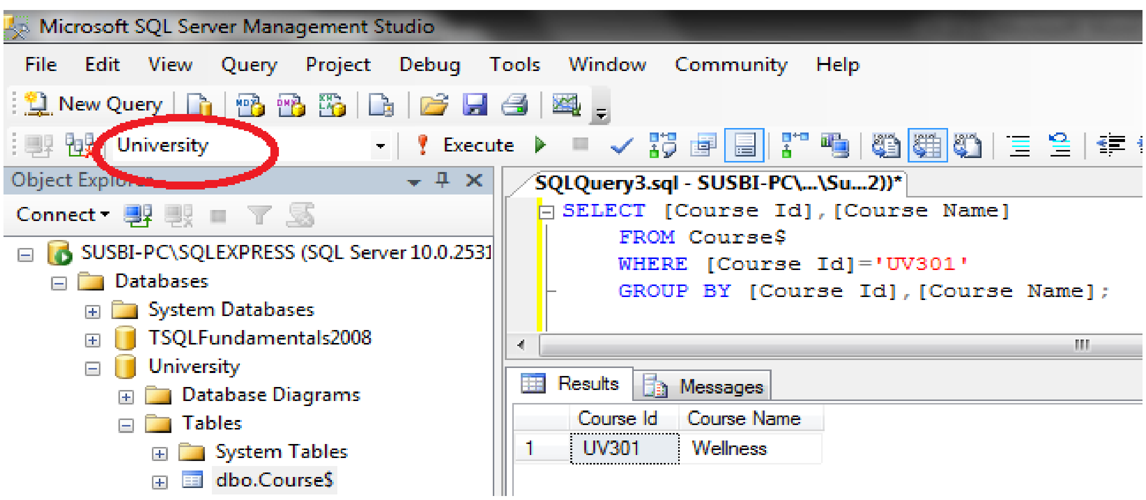


Figure 29. Screenshot of SQL Query Statement with a Group By Clause.

5.4.2. SQL Query Statement with the Order By Clause

The Order By Clause allows the user to sort the records in the generated result set. The Order By Clause can only be used with Select Statements. In terms of logical query processing, the Order By Clause is the very last Clause to be processed. The syntax of the query is given below [11]. Also Figure 30 shows the screenshot of the result of the query.

SELECT [column_name(s)]

FROM [table_name]

ORDER BY [column_name] ASC/DESC;

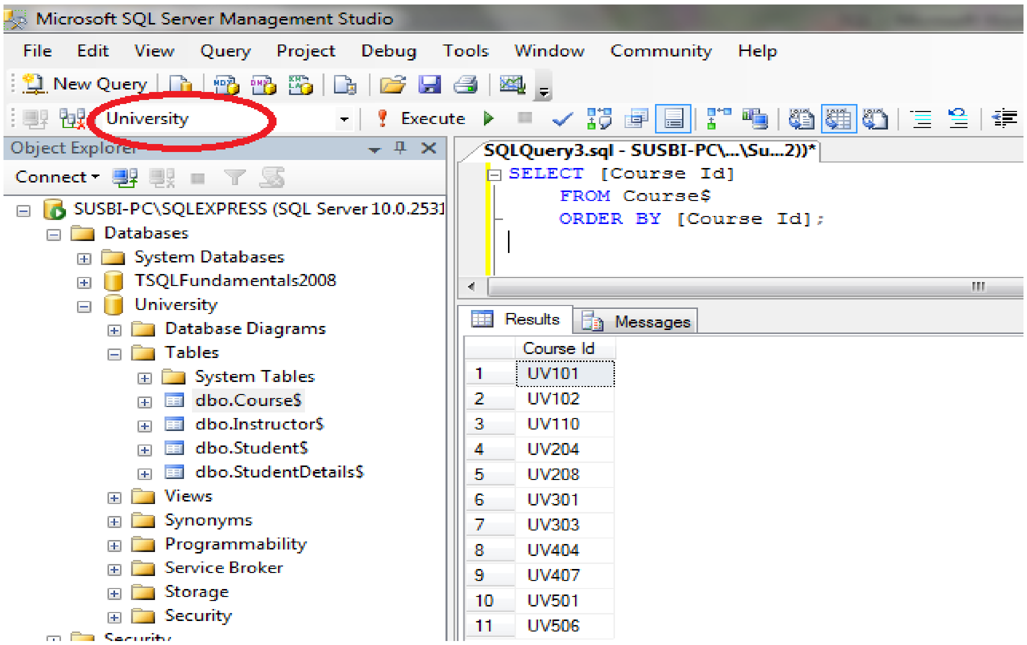


Figure 30. Screenshot of SQL Query Statement with Order By Clause.

5.4.3. SQL Query Statement with a Top Clause

The Top Clause is the feature in SQL that allows the user to limit the number or percentage of rows that the query returns as a result set. When an Order By Clause is specified in the query, the Top Clause relies on it to define the logical precedence among rows [11]. The syntax of the query is given below [11]. Also Figure 31 shows the screenshot of the result of the query.

SELECT TOP (3) [column 1], [column 2]

FROM [table_name]

ORDER BY [column_name] DESC;

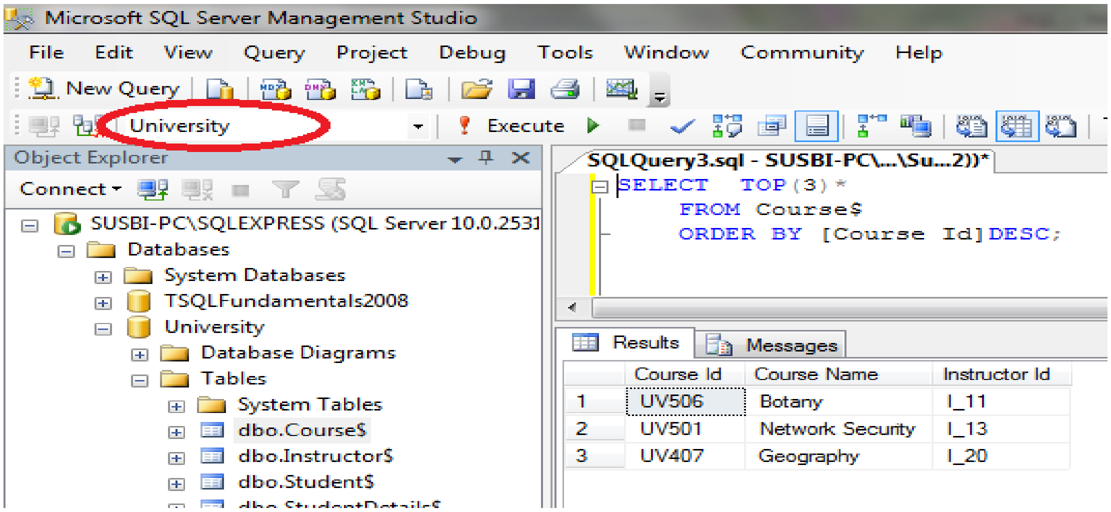


Figure 31. Screenshot of SQL Query Statement with a Top Clause.

Task 3 Questions

i. Write a Select Statement to print the Student id and Course id from the Student Details table, where Course id = "UV404" Group By Student id and Course id. Create a screenshot of the result set, and paste it into your solution document.

ii. Write a Select Statement to print all details from the Instructor table Order By "Joining year" in ascending order. Create a screenshot of the result set, and paste it into your solution document.

iii. Write a Select Statement to print the Top seven values from the Course id and Course name columns from the Course table. Create a screenshot of the result set, and paste it into your solution document.

iv. Write a Select Statement to print the Top five values of all columns from the Student table Order By "Admission year" in descending order. Create a screenshot of the result set, and paste it into your solution document.

Figure 32, shows the mapping between "Task 3- SQL Query Statements with Clauses" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 13.

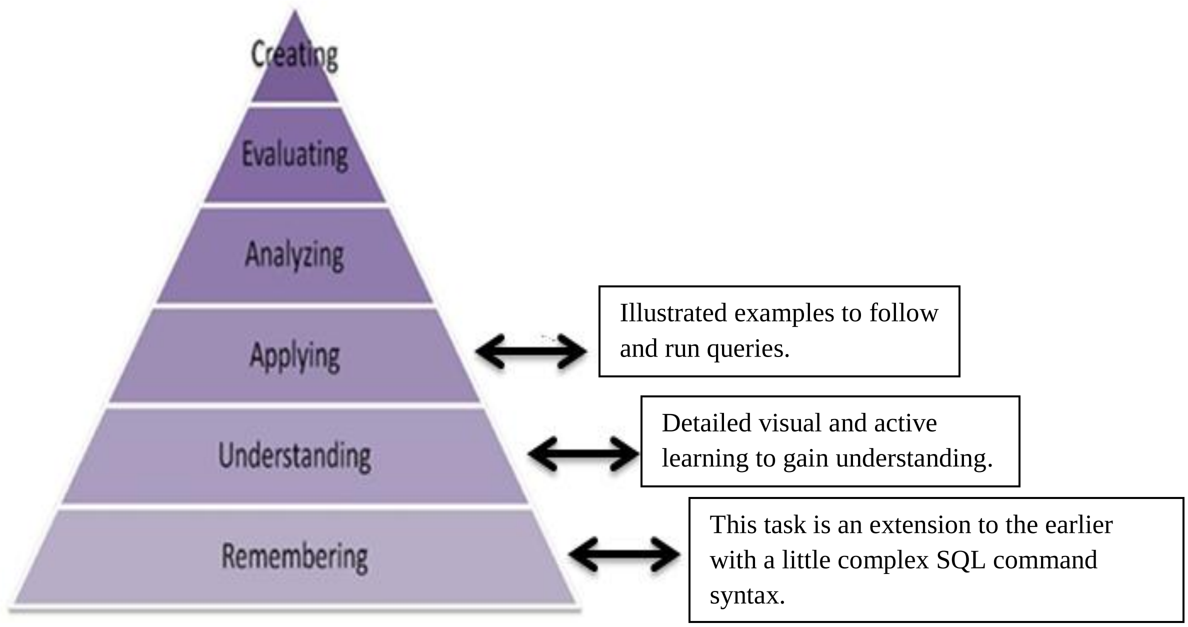


Figure 32. Revised Taxonomy Levels for SQL Query Statements with Clauses.

Table 13. Revised Taxonomy Levels for SQL Query Statements with Clauses.

Taxonomy Level	**Description**
Remembering	This task is designed as an extension to the earlier task. It provides SQL syntaxes with further details and Clauses.
Understanding	As seen from the previous tasks, the tutorial employs a detailed visual and active learning style for the users by providing some working examples for the users to obtain a thorough understanding.
Applying	As the users gain information about how to run a query, practical implication is a must. The users are given illustrations under each task section to follow and implement by running those SQL commands.

5.5. Task 4: SQL Query and Charater Data

5.5.1. Working with Character Data

In this task, the user is expected to grasp knowledge of query manipulation for character data, including data types, operators and functions, and pattern matching.

5.5.1.1. Upper and Lower Functions

The Upper and Lower functions return the input string with all uppercase or lowercase characters [11]. The syntaxes of both the Upper and Lower query are given below [11]. Also Figure 33 shows the screenshot of the result of the query.

SELECT UPPER ([column_name])

FROM [table_name];

SELECT LOWER ([column_name])

FROM [table_name];

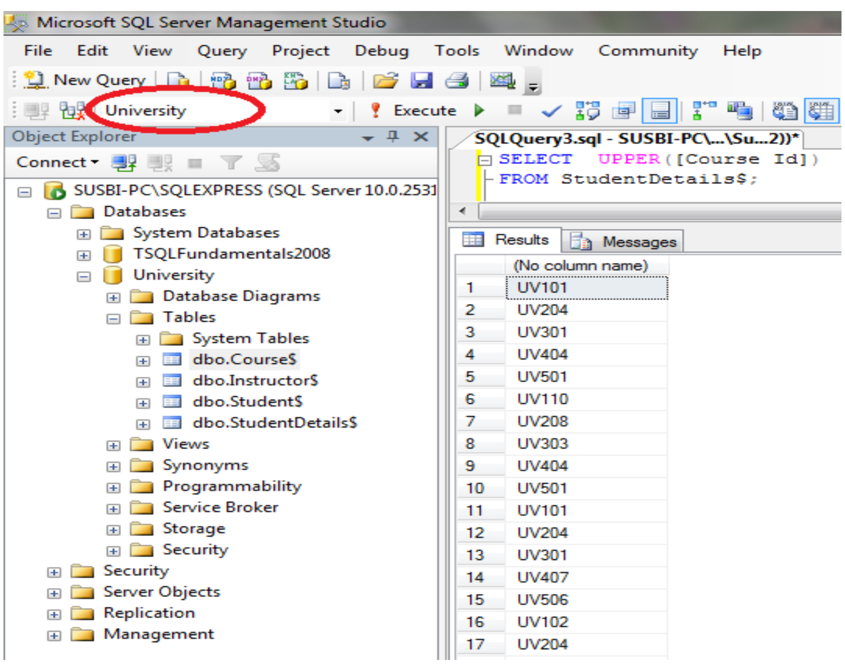


Figure 33. Screenshot of SQL Query with Upper Function.

5.5.1.2. Like Predicate

SQL provides a predicate called Like Predicate that allows the user to check whether a character string matches a specified pattern [11].

5.5.1.3. % (Percent) Wildcard

The Percent Sign represents a string of any size, including an empty string [11]. This query returns a result set where the column 2 starts with the character "D." The syntax of SQL query with both the Like Predicate and Percent Wildcard is given below. Also Figure 34 shows the screenshot of the result of the Like Predicate and the Percent Wildcard query.

SELECT [column 1], [column 2]

FROM [table_name]

WHERE [column 2] LIKE N 'D%';

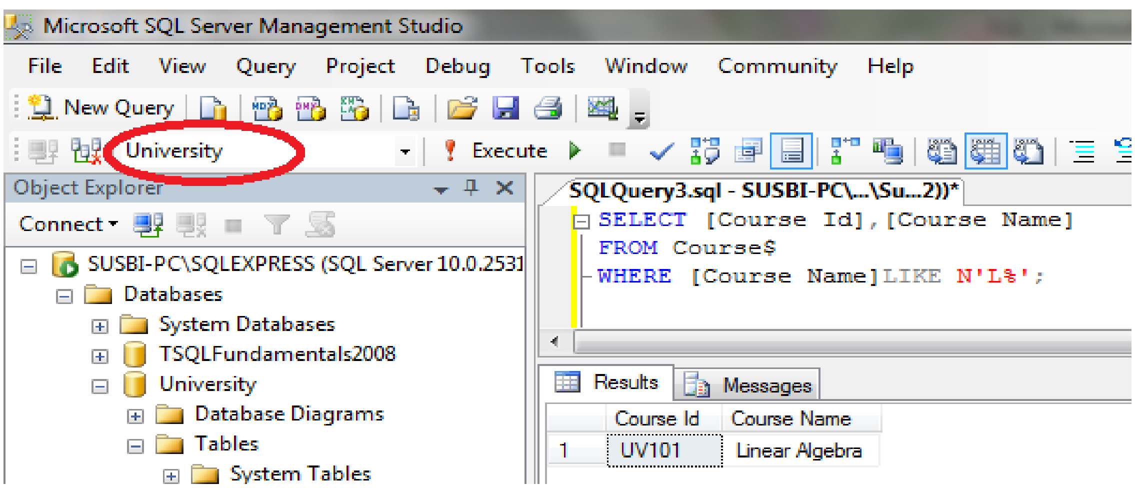


Figure 34. Screenshot of SQL Query with Like Predicate and Percent Wildcard.

5.5.1.4. _ (Underscore) Wildcard

An underscore represents a single character. The following query returns a result set where the second character in column 2 is "e". Also Figure 35 shows the screenshot of the result of the query.

SELECT [column 1], [column 2]

FROM [table_name]

WHERE [column 2] LIKE N **'_e%'**;

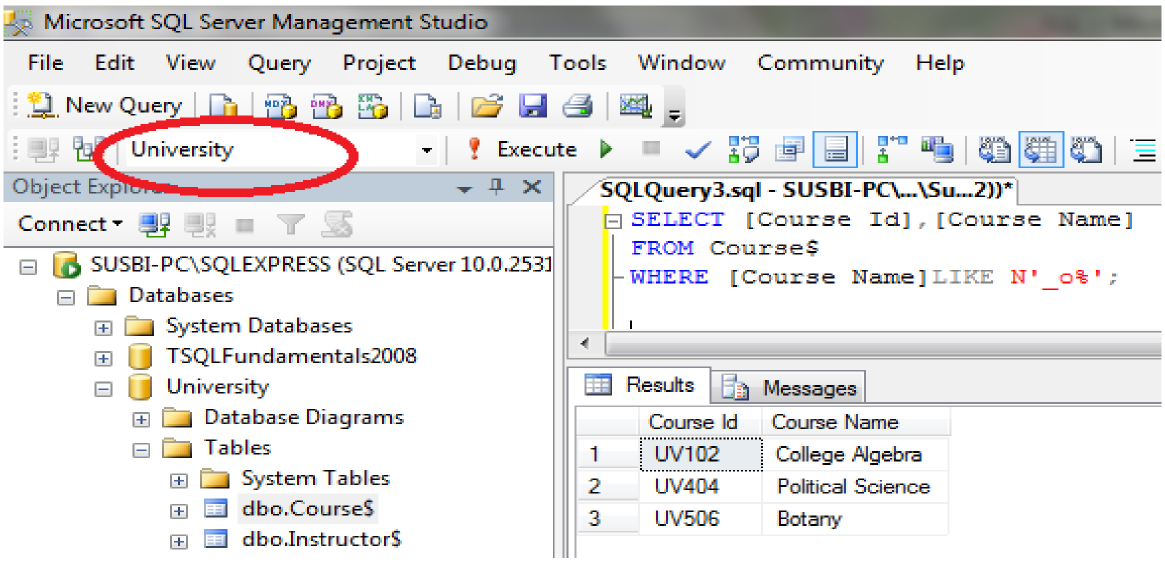


Figure 35. Screenshot of SQL Query with an Underscore Wildcard.

5.5.1.5. [<List of Characters>] Wildcard

Square brackets with a list of characters (such as "[ABC]") represent a single character that must be one of the characters specified in the list. For example, the following query returns a result set where the first character in column 2 is "A," "B," or "C" [11]. Also Figure 36 shows the screenshot of the result of the query.

SELECT [column 1], [column 2]

FROM [table_name]

WHERE [column 2] LIKE N **'[ABC]%'**;

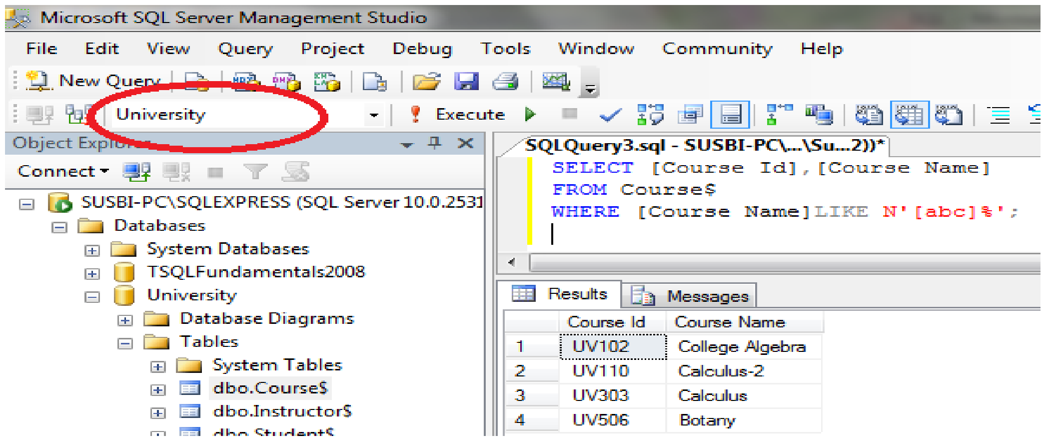


Figure 36. Screenshot of SQL Query with a List of Characters Wildcard.

5.5.1.6. [<Character> - <Character>] Wildcard

Square brackets with a character range (such as “[A-E]”) represent a single character that must be within the specified range. For example, the query given below returns a result set where the first character in column 2 is “A” through “E” [11]. Also Figure 37 shows the screenshot of the result of the query.

SELECT [column 1], [column 2]

FROM [table_name]

WHERE [column 2] LIKE N **‘[A-E]%’**;

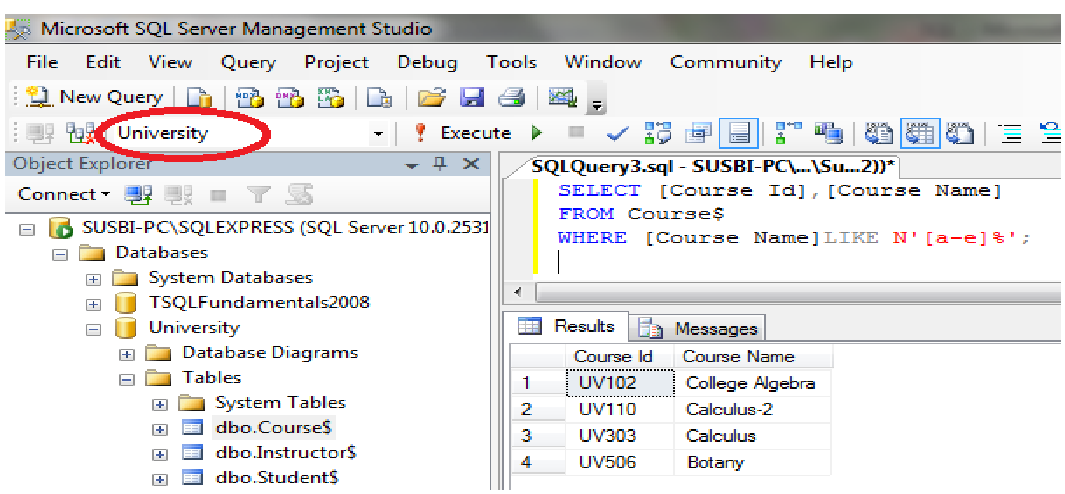


Figure 37. Screenshot of SQL Query with [<Character> - <Character>] Wildcard.

Task 4 Questions

i. Write a Select Statement to print the Course name in uppercase, From the Course table, using the Upper function query. Create a screenshot of the result set, and paste it into your solution document.

ii. Write a Select Statement to print the names of all the instructors, From the Instructor table, whose name starts with "T." Create a screenshot of the result set, and paste it into your solutions document. (Note: Refer to the % (Percent) Wildcard query syntax shown above.)

iii. Write a Select Statement to print the last name of all the students, From the Student table, that fall in the range "R" through "Z." Create a screenshot of the result set, and paste it into your solution document. (Note: Refer to the [<Character> - <Character>] Wildcard query syntax shown above.)

Figure 38, shows the mapping between "Task 4- SQL Query and Character Data" and the levels of the Revised Bloom's Taxonomy. A detailed explanation of the mapping is provided in Table 14.

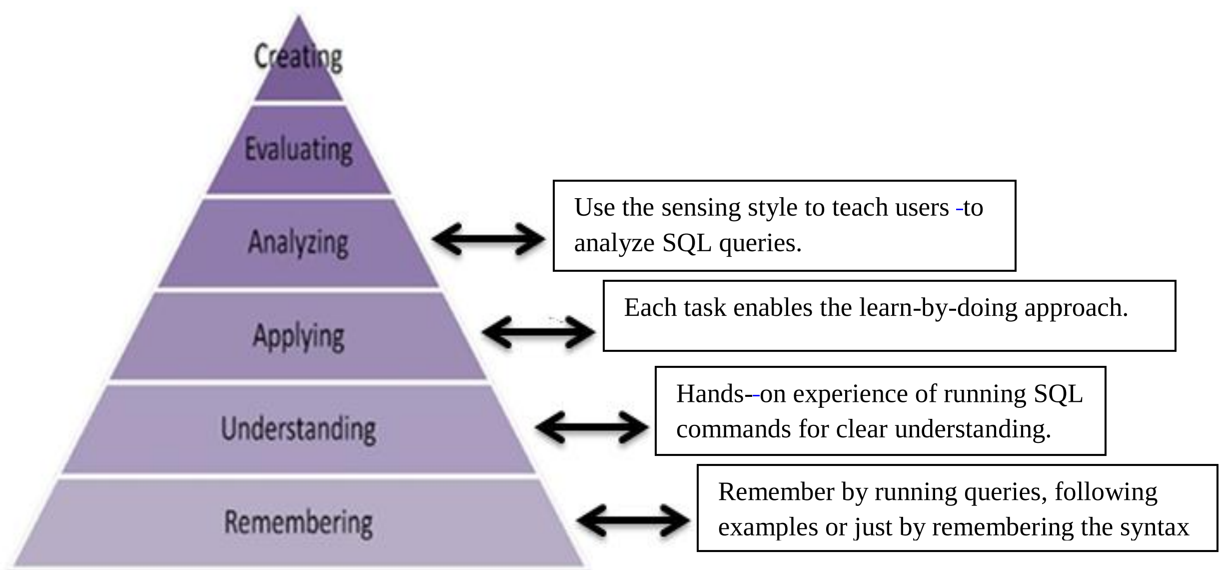


Figure 38. Revised Taxonomy Levels for an SQL Query and Character Data.

Table 14. Revised Taxonomy Levels for an SQL Query and Character Data.

Taxonomy Level	**Description**
Remembering	Similar to what the users have been doing in earlier tasks, the user needs to substantiate the lower-order thinking level of remembering the appropriate SQL command to use. In order to do this, the user is given detailed examples as well as the syntax to run the query.
Understanding	The most appropriate method to understand how to run SQL commands is to gain hands-on experience by running those commands. Thus, the users are provided with examples for all SQL commands illustrated throughout the task.
Applying	Each section in the task enables the users to work on the SQL queries in order to gain concrete knowledge as well as hands-on-experience.
Analyzing	The tutorial employs a sensing style to teach the users to analyze through the delivered syntax and methodology, and to understand which SQL command fits a particular problem scenario.

5.6. Task 5: SQL Query with Date and Time

5.6.1. Working with Date and Time

An SQL date and timestamp is a record containing date/time data, such as the month, day, year, hour, and minutes/seconds. In SQL Server 2008, “Date” and “Time” are introduced as separate data types, whereas in the previous versions of SQL Server, there was no separation between them. Also, SQL Server 2008 introduces separate Date and Time data types, DateTime, etc., which differ in their data range and accuracy as well as, most importantly, their entry format [11].

5.6.2. Date and Time Functions

5.6.2.1. Current Date and Time

To get the current date and time, we can use some of the following SQL Date and Time command functions [17]. The syntax of SQL Date and Time query is illustrated below. Also Figure 34 shows the screenshot of the result of the Date and Time query.

SELECT GETDATE() AS [CURRENT_DATE&TIME];

SELECT CURRENT_TIMESTAMP AS [CURRENT_DATE&TIME];

SELECT SYSDATETIME()AS [CURRENT_DATE&TIME];

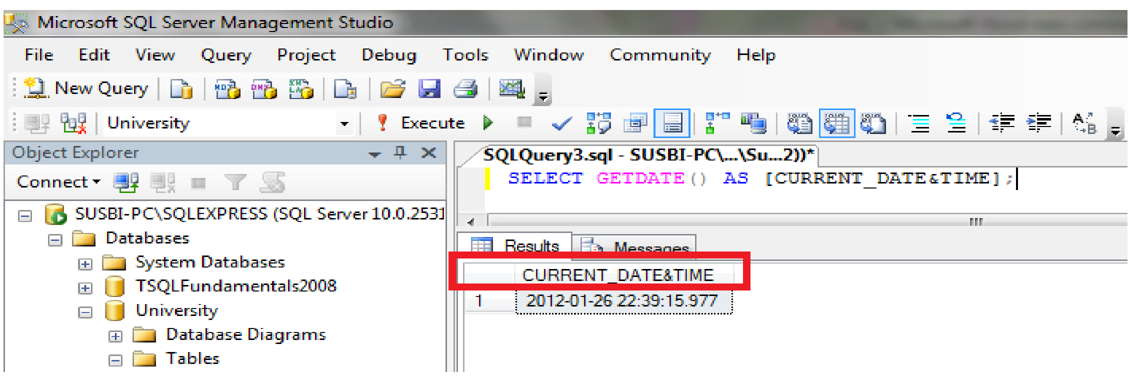


Figure 39. Screenshot of SQL-Date and Time Query.

5.6.2.2. Cast Function

The above-mentioned queries yield both the current date and current time. What if we need the current date without the time? What shall we do in that case? We use the "Cast function" to solve this problem. By using the Cast function on the same queries mentioned above, we can yield only the current Date or only the current Time. Look at the following SQL commands that use the Cast function [17]. Figure 40-41, show the screenshots of the result of the query.

SELECT CAST(SYSDATETIME() AS DATE) AS [CURRENT_DATE];

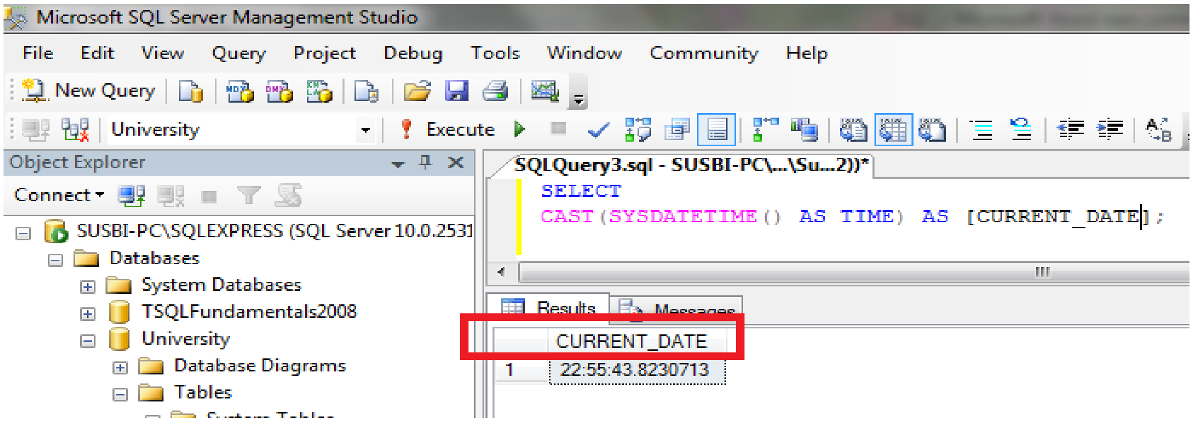


Figure 40. Screenshot of SQL-Cast Query: Current Date.

SELECT CAST(SYSDATETIME() AS TIME) AS [CURRENT_TIME];

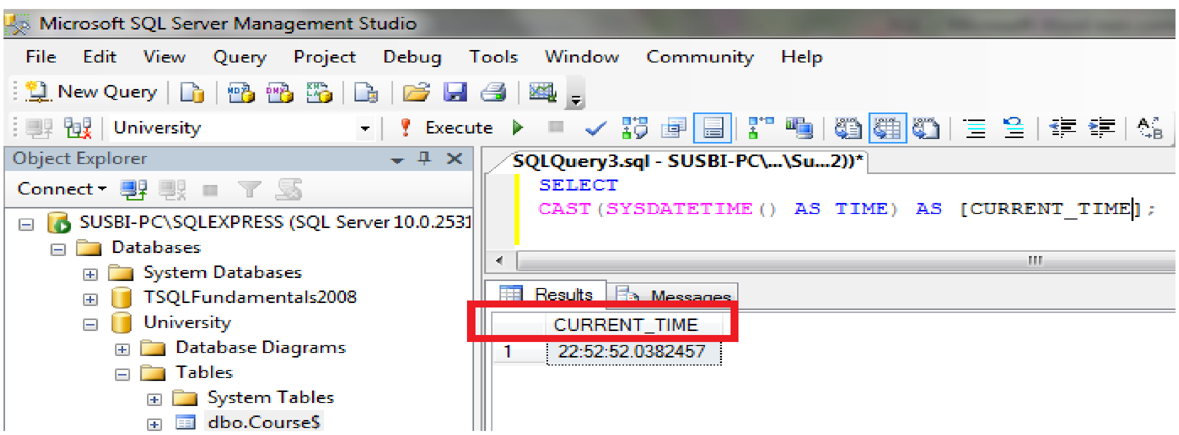


Figure 41. Screenshot of SQL-Cast Query: Current Time.

5.6.2.3. DatePart Function

The DatePart function returns an integer representing the requested part of a given date and time value [17]. The syntax of SQL Query with the DatePart Function is illustrated below. Figure 42-44, show the screenshots of the result of the DatePart Function query.

DATEPART (part, dt_value);

SELECT DATEPART(month,'20120112')AS MONTH;

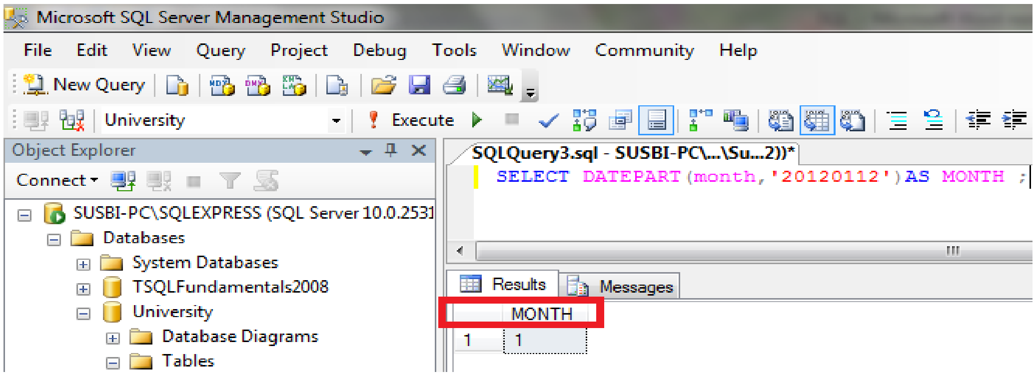


Figure 42. Screenshot of SQL Query with the DatePart Function.

SELECT DATEPART(day,'20120112')AS DAY;

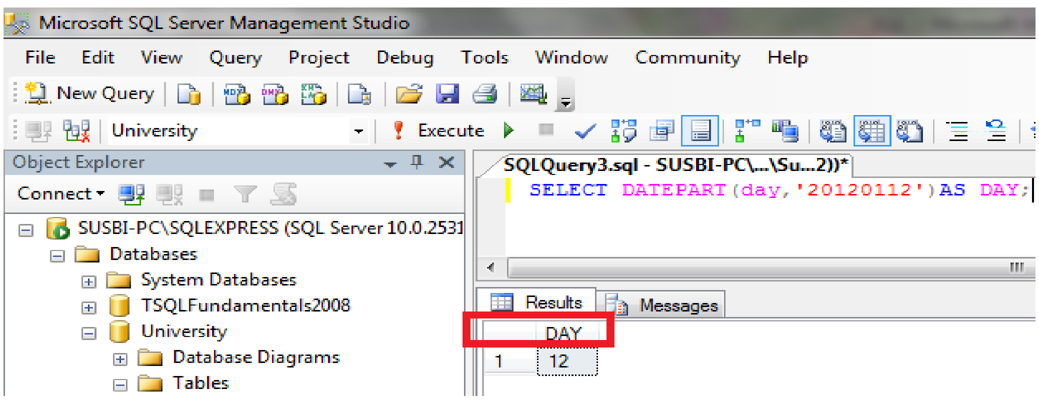


Figure 43. Screenshot of SQL Query with the DatePart Function.

SELECT DATEPART(year,'20120112')AS YEAR ;

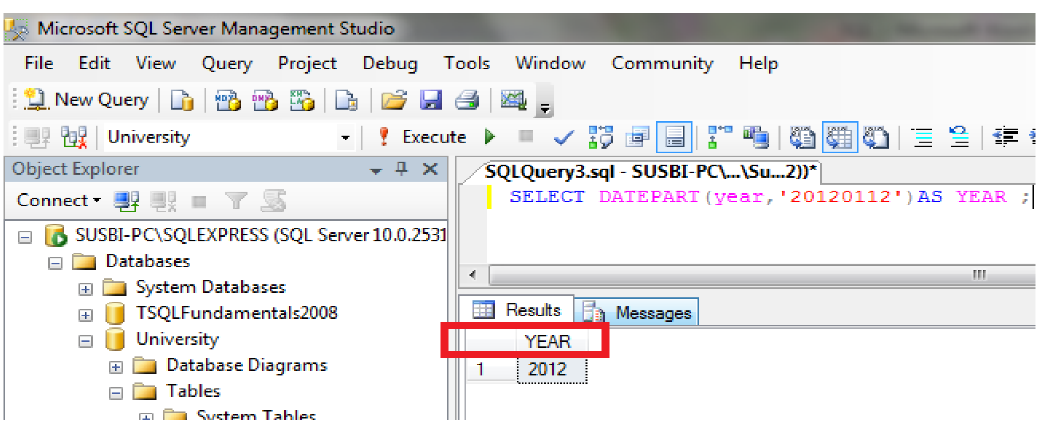


Figure 44. Screenshot of SQL Query with the DatePart Function.

5.6.2.4. DateName Function

The DateName function returns a character string representing part of the given date and time value [11]. The syntax of SQL Query with the DateName Function is illustrated below. Figure 45-47, show the screenshots of the result of the DatePart Function query.

DATENAME (part,dt_val);

SELECT DATENAME(month,'20120112')AS MONTH ;

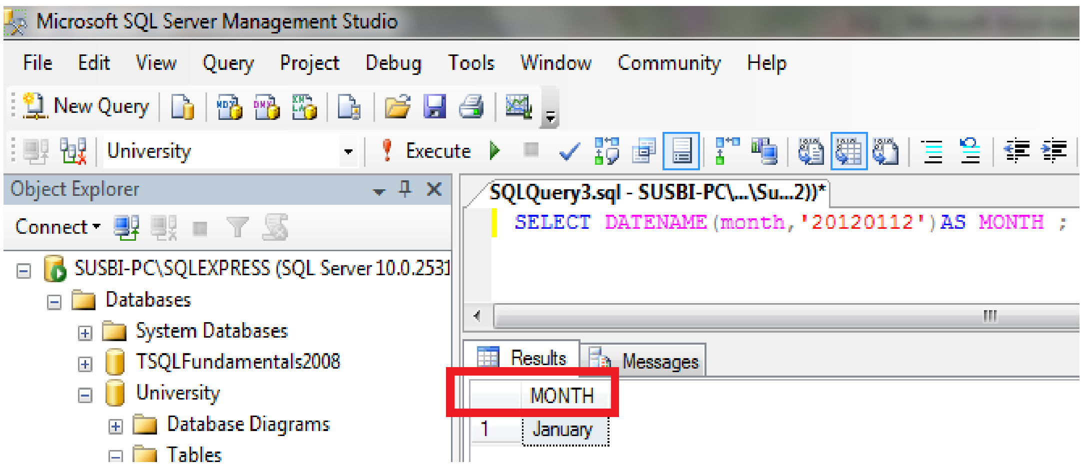


Figure 45. Screenshot of SQL Query with the DateName Function.

SELECT DATENAME(day,'20120112')AS DAY ;

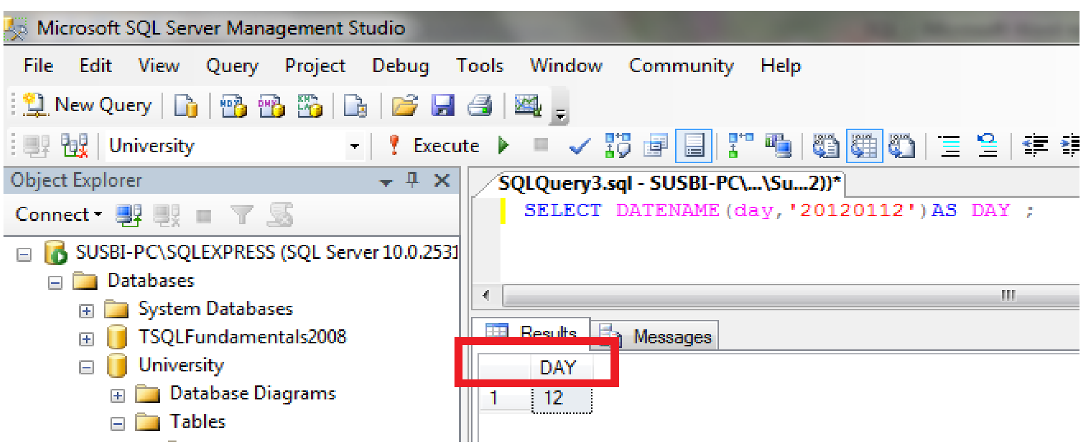


Figure 46. Screenshot of SQL Query with the DateName Function.

SELECT DATENAME(year,'20120112')AS YEAR ;

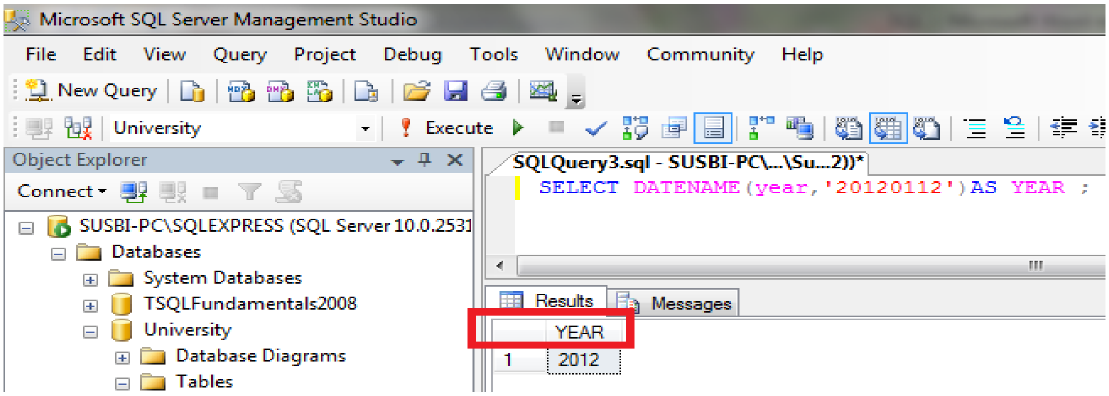


Figure 47. Screenshot of SQL Query with the DateName Function.

Task 5 Questions

i. Write the SQL query statements for the following date and time fields. Create a screenshot of both the SQL query and the result:

- Current Date
- Current Time
- Current Date and Time

ii. Write a SQL query statement using the Cast function to cast the following SQL-Date and Time functions. Place screenshots of the results:

- Print the Date and Time separately using the GETDATE() function.
- Print the Date and Time separately using the SYSDATETIME() function.

Figure 48, shows the mapping between “Task 5- SQL Query with the Date and Time Functions” and the levels of the Revised Bloom’s Taxonomy. A detailed explanation of the mapping is provided in Table 15.

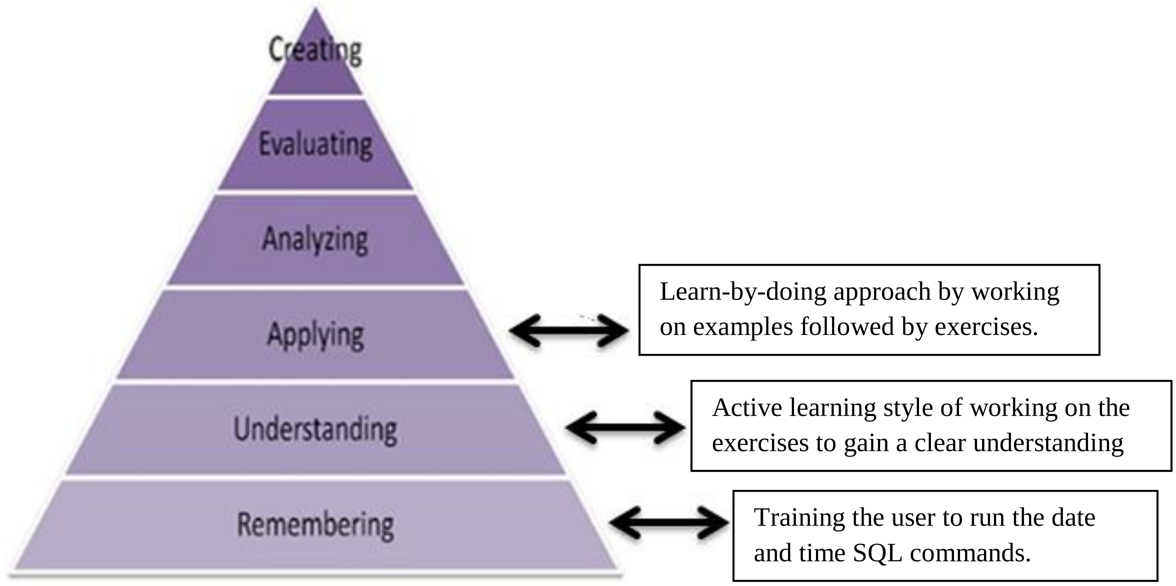


Figure 48. Revised Taxonomy Levels for an SQL Query with the Date and Time.

Table 15. Revised Taxonomy Levels for an SQL Query with the Date and Time.

Taxonomy Level	**Description**
Remembering	The last task is significantly diverse from the former tasks because it trains the user to run a query based on the date and time rather than working on a database table itself. Thus, in this task, the user shall learn how to deal with the date and time SQL commands.
Understanding	Following an active learning style of working on the exercises and examples would help users understand how to get the respective date and time outcome following the illustrated SQL command syntaxes.
Applying	Date and time query syntaxes, comprehensive examples, as well as structured exercises are all set up to provide a hands-on working environment for the user to gain proficiency writing these date and time SQL commands.

5.7. Summary

So far, we have illustrated each of the five tasks with different levels of the Revised Bloom's Taxonomy. This illustration is based on the mapping between the tasks and the levels of the Revised Bloom's Taxonomy (Cognitive Domain). You shall notice that, in certain tasks, a few of the taxonomy levels are missing. The main reasons are as follows:

i. For a particular task assigned to the user, the exercises performed by the user, map well, only until the "Analyzing level" in the Revised Blooms Taxonomy.

ii. The Revised Bloom's Taxonomy mapped here in the five tasks is not restrained to the tasks. Rather, these mappings with the tasks are illustrated keeping a much broader perspective that covers the user's knowledge and understanding of hands-on work with databases in mind.

CHAPTER 6. CONCLUSION, LIMITATIONS, AND FUTURE WORK

The two database tutorials designed in this paper, which are illustrated in Chapters 4 and 5; contain individual tasks in order to provide detailed-guided assignments for the users. This piece of work would be a good start for teachers to use it in database concepts or related classes in order to help the students better understand database fundamentals as well as perform various operations using databases. The two tutorials are inspired by the Revised Bloom's Taxonomy. While designing the tutorials, an effort has been made to accommodate all levels of learning objectives.

6.1. Conclusion

The two tutorials based on the learning theory are task-oriented learning procedures which follow the Revised Bloom's Taxonomy levels very closely. The main emphasis of using the Revised Bloom's Taxonomy is to match the skill levels of different users. The users perform hands-on exercises in each of the tasks illustrated in the two assignments. These tasks are designed keeping the ease of understanding for the user in mind and can easily be followed by any college-level user with little background in computer science.

6.2. Limitations

This task-oriented approach might not work well or prove useful for those users who do not have a need to learn databases in depth as much as to create websites based on databases. In order to work with the two tutorials, the users must acquire software like MS Excel 2010, MS Access 2010, and SQL Server 2008 R2. The software used in the two tutorials, mainly MS Excel and MS Access, might not be available for students taking online courses; therefore, students might need to buy software. Also, these two tutorials might be very basic level tutorials for all the advanced users.

6.3. Future Work

There is always a room for improvement in any accomplished task. A few possible enhancements for this paper are illustrated below.

6.3.1. Improvisation of Taxonomy

Each task provided in the two tutorials map with only the first three levels in the Revised Bloom's Taxonomy. Thus these tasks can be amended to map all the learning objective levels in the Revised Taxonomy. Since these tutorials emphasize more on teaching users the fundamentals of databases, there is room for teaching more advanced database techniques.

6.3.2. Improvisation of Access Database Tutorial

The tutorial based on MS Excel 2010 and MS Access 2010 touches all aspects of working with databases using Access. At the same time, more information based on fundamentals as well as exercises can be added for the users to test their command of the Access database.

6.3.3. Improvisation of an SQL Database Tutorial

The tutorial designed for teaching Structured Query Language to the users also has a scope of improvement as well as more additions. The complexity of this tutorial is set at moderate in order to serve all users, whether they have former SQL experience or not. Therefore, more additions based on SQL queries and other SQL objects, such as joins, stored procedures, views, triggers, etc., can be made, thereby increasing the complexity of the tutorial. These additions would prove to be very substantial for all users who want to gain more in-depth knowledge and hands-on working experience using an SQL Server.

6.3.4. Database-Driven Website Development Training

This tutorial can be extended on a large scale for use in real-time, database-driven web applications to train employees in organizations. Instead of going too much into the basics, one step followed by another, higher-level details could be included to serve a wider range of users.

REFERENCES

[1] M. K. Bhogadi, *Teaching encryption : a learning theory approach*, Fargo: Unpublished master paper, Department of Computer Science, North Dakota State University, 2010.

[2] T. Gram, "Fun with learning taxonomies," 4 Feb 2009. [Online]. Available: http://gramconsulting.com/2009/02/fun-with-learning-taxonomies/. [Accessed Jan 2012].

[3] "Blooms Taxonomy," [Online]. Available: http://projects.coe.uga.edu/epltt/index.php?title=Bloom%27s_Taxonomy. [Accessed Jan 2012].

[4] "COL- Best Practice:Blooms Taxonomy," [Online]. Available: http://academics.georgiasouthern.edu/col/id/bloom.php. [Accessed Jan 2012].

[5] B. S. Bloom, Taxonomy of Educational Objectives:The Classification of Educational Goals, New York: David McKay Co. Inc., 1956, pp. 7-8.

[6] "Teaching Resources- Blooms Taxonomy," [Online]. Available: http://tep.uoregon.edu/resources/assessment/multiplechoicequestions/blooms.html. [Accessed Jan 2012].

[7] M. J. Pickard, "The New Bloom's Taxonomy: An Overview For Family and ConsumerSciences," *Journal of Family and Consumer Sciences Education,* vol. 25, no. 1, pp. 45-50, 2007.

[8] "Blooms Taxonomy of Cognitive Levels- Revised," [Online]. Available: http://faculty.chass.ncsu.edu/slatta/hi216/learning/bloom.htm. [Accessed Jan 2012].

[9] R. Garimedi, *A tutorial for teaching e-commerce applications,* Fargo: Unpublished classroom notes, Department of Computer Science, North Dakota State University, 2009.

[10] R. F. v. d. Lans, Introduction to SQL, Wokingham, England: Addison-Wesley Pub. Co., c1993., 1993.

[11] B.-G. Itzik, Microsoft SQL Server 2008 T-SQL Fundamentals, Redmond: Microsoft Press, 2009.

[12] F. Coulson, "3 Normal Forms Database Tutorial," [Online]. Available: http://phlonx.com/resources/nf3/. [Accessed Jan 2012].

[13] M. Chapple, "Database Normalization Basics," [Online]. Available: http://databases.about.com/od/specificproducts/a/normalization.htm. [Accessed Jan 2012].

[14] "Microsoft Access vs SQL Server," [Online]. Available: http://www.quackit.com/microsoft_access/tutorial/microsoft_access_versus_sql_server.cfm. [Accessed Jan 2012].

[15] "Microsoft SQL Server," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Microsoft_SQL_Server. [Accessed Jan 2012].

[16] "What is SQL Server," Webopedia, [Online]. Available: http://www.webopedia.com/TERM/S/SQL_Server.html. [Accessed Jan 2012].

[17] W. Software, Learning SQL, Englewood Cliffs, NJ: Prentice Hall, 1991.