HARAMAYA UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

PROJECT TITLE: RIFT VALLEY UNIVERSITY REGISTRAR DATABASE MANAGEMENT SYSTEM, HARAR CAMPUS

BY

1. **Lemi Wakuma Jifara ----------379/14**
2. **Henok Geto Hussien ---------- 370/14**
3. **Ebrahim Beker Esmo ----------356/14**
4. **Elias Cherenet weldemariam ----------357/14**

Submitted to: Mr. Asedo Sh.

Haramay, Ethiopia
NOV.2023

Table of Contents

Figures list

Tables list

Chapter 1

Introduction

1.1 Background

Rift Valley University was founded in October 2000 in Adama, Ethiopia. It started with 154 evening program students, five part-time faculty staff, and a capital of 1.3 million Ethiopian birr. The Asella branch campus opened three months later, offering Accounting, Law, and Marketing Management programs. By the end of the 2000/2001 academic year, there were about 250 students enrolled in five diploma programs at the two campuses.
The Gotera and Batu branch campuses were established in September 2003 and 2004, respectively. The Bishoftu campus opened two years later, followed by two other branch campuses in Dire Dawa and Chiro in August/September 2005. The Bole and Gulele campuses opened in October 2005, and the Harar campus was established in October 2006. The institution was granted the authority to award bachelor's degrees, Accounting, Business Management, and Law were the first academic programs offered on the campus in Adama.

Rift Valley University Harar campus was established in 2006 and began offering diploma programs in Information Technology, Accounting, Nursing, Pharmacy, Midwifery, and others. Currently, Rift Valley University offers both degree and diploma programs, as well as master's degrees in Project Management and Business Administration (MBA).

1.2 Statement of problem

The manual registrar system at Rift Valley University, established in 2000, has proven to be tedious, time-consuming, and difficult to manage. The existing file-based student information system lacks accuracy, timeliness, flexibility, and control mechanisms. This leads to challenges in database management, registration processes, and service delivery. The system suffers from data collection inaccuracies, duplication of data, difficulties in editing and validating information, and delayed output generation. The storage of data requires excessive physical space, and the system does not accommodate additional requirements efficiently. Moreover, the current system incurs high costs due to the large number of staff required for manual operations and the wastage of resources. Control and security mechanisms are lacking, posing a risk to student information confidentiality. The registration process is time-consuming, resulting in long waiting times for students. Overall, the manual processes and paper-based activities hinder the efficiency and effectiveness of the registrar system.

To address these challenges, there is a clear need for a more streamlined automatic database management system and efficient data handling system for Rift Valley University. This research aims to design and implement a secure, efficient, and flexible student information system that automates registration processes, improves data accuracy and timeliness, and enhances service delivery. The system should include authentication mechanisms to ensure data confidentiality and

minimize the wastage of materials and resources. By automating the system, the university can reduce costs, improve efficiency, and provide faster and more reliable services to students and staff. Implementing a centralized database system would facilitate easier access to information and reduce the time required to retrieve data. Such a system would provide a standardized and centralized approach to data management, ensuring clarity, consistency, and timely availability of information.

1. 3 Objectives

1.3.1 General objective

The general objective this project is to develop automated registrar database management system for RVU that support easily management of students' record such as student, grade, course, instructors, department and schedule.

1.3.2 Specific Objectives

The specific objectives of this project are listed as following:

- ✓ To assess the existing registrar system of RVU Harar Campus student database
- ✓ To create automated students database for the RVU

1.4 Scope

The scope of this project work is on the development of the registrar student database management system of RVU Harar campus. This includes developing of platforms for courses registration, schedule setting, grade submission, and allowing student's enrollment for different courses they have registered for the semester. Furthermore, this project mainly delimited on the development of the main database schema based on the information collected and user requirement, hence it doesn't involve testing and implementing the prototype of the proposed registrar system. .

1.5 Limitations

One limitation of this project is that it is delimited on the development of the registrar database schema or conceptual design for the new proposed registrar system of RVU. However, the workability and functionality of the system have not been tested in a practical setting.

To ensure the success of the new registrar's database system, it should undergo thorough testing to assess its compatibility, efficiency, and user-friendliness with existing software and the capacity of the staff members. This testing would involve evaluating the system against user requirements and industry standards for developing registrar systems

1.6 Constraints

Several challenges and constraints were encountered during the data collection process, making it difficult to obtain the required data in a timely and comprehensive manner. As a result, we had to rely on various secondary sources such as website reviews, reports, and existing databases. However, these sources presented issues related to data accuracy, clarity, and inconsistency. Retrieving information from past records proved to be time-consuming, causing significant delays in project timelines. Complicating matters further was the lack of a centralized data handling process, resulting in scattered information across different offices. For instance, student information was managed by the administration, while staff information was held by human resources. This fragmented approach hindered the efficient and timely access to all the necessary information required by students.

1.6 Significance of the Project

The database project on the RVU registrar management system is significant because it can help the university address the challenges of its current manual data handling process and scattered information. The project can provide insights into how to modernize the system, develop a centralized database, and make data collection more efficient and reduce delays.

Furthermore, the project aligns with the school's general and specific objectives of producing competent and well-versed graduates who can manage the RVU registrar database. By comprehending the fundamental problems and exploring potential solutions through research, the project can help students become active civic-minded agents who are passionate about their university and aware of modern education and learning systems.

Overall, the database project on the RVU registrar is significant because it can help the university achieve its goals of providing quality education, becoming a research center, and producing quality education for the community.

In addition to the above, the project can also have the following benefits:

- Improved accuracy and consistency of data
- Reduced risk of human error
- Increased accessibility to data for students, staff, and administrators
- Improved ability to generate reports and analyze data
- Enhanced decision-making capabilities

1.7 Methodology of the data collection

Methodology for acquiring data for a database project on RVU registrar management system. Our team followed a comprehensive methodology to acquire data for our database project on the RVU registrar management system.

I. Conduct face-to-face interviews with the different departments.

Purpose: To gather information on their programs, faculty, and research activities, as well as their needs and expectations for a new database system.

Procedure:

1. Schedule interviews with the heads of each department.
2. Prepare a list of questions to ask during the interviews.
3. Conduct the interviews and take detailed notes.
4. Transcribe the notes and review them carefully.

II. **Conduct a thorough *analysis of the data collected.***

Purpose: To identify the key data elements that need to be included in the database, and to develop a database design that will be efficient and scalable.

Procedure:

1. Review the interview transcripts and other data sources to identify the key data elements.
2. Group the data elements into logical categories.
3. Develop a database schema that defines the relationships between the different data tables.
4. Normalize the database schema to eliminate redundancy and improve data integrity.

III. Check the feasibility of the project.

Purpose: To assess the resources and expertise available, and to develop a realistic plan for implementing the database system.

Procedure:

1. Identify the resources that will be needed to implement the database system, such as hardware, software, and personnel.
2. Assess the expertise of the team that will be developing and implementing the database system.
3. Develop a project plan that outlines the tasks that need to be completed, the resources required, and the estimated timeline.

IV. Review RVU's website and other publicly available information.

Purpose: To gain a better understanding of the university's overall mission, goals, and objectives, and to ensure that the database system will be aligned with these priorities.

Procedure:

1. Review RVU's website and other publicly available information, such as its strategic plan, annual reports, and course catalogs.
2. Identify the university's key stakeholders and their needs.
3. Ensure that the database system will be designed to meet the needs of the university's key stakeholders.

V. Conduct data quality assurance.

Purpose: To ensure that the data in the database is accurate, complete, and consistent.

Procedure:

1. Develop a data quality assurance plan that outlines the procedures that will be followed to identify and correct any errors or inconsistencies in the data.
2. Implement the data quality assurance plan.
3. Review the data quality assurance results and make any necessary adjustments.
4. Provide user training and support.

1.7 System Development Tools

While developing the project started from the documentation to the implementation we used the following case tools:

Hardware tools used to develop this project activities

Table 1 Hardware Tool

tools	specification	purpose
Desktop computer	Core i7	Server
Flash	32Gb	Store and transfer
paper	A4 size	To collect information

Software used during the data analysis and preparation

Table 2 Software Tool

Tools	Activities
MySQL , Edraw Max	For editing , and drawing (Data base)
Mozilla Firefox ,Chrome	Browser
Ms. Office 2010	For Documentation and presentation

1.8 Time schedule and work breakdown

All the activity done in the project with their respective times has been shown as follows chart

Table 3 Work breakdown and activity seclude

Task	*Duration (days)*	*Start Date*	*End Date*
Phase 1: Requirements Gathering and Analysis	10	2023-11-09	2023-11-18
- Identify and document user requirements	5	2023-11-09	2023-11-13
- Analyze user requirements and develop system specifications	5	2023-11-14	2023-11-18
Phase 2: Database Design	15	2023-11-21	2023-12-05
- Develop the conceptual database model	5	2023-11-21	2023-11-25
- Translate the conceptual model into the physical database model	5	2023-11-28	2023-12-02
- Normalize the database model	5	2023-12-03	2023-12-05
Phase 3: Database Implementation	20	2023-12-06	2023-12-29
- Create the database schema in the chosen database management system	5	2023-12-06	2023-12-10
- Develop database access code	10	2023-12-11	2023-12-22
- Implement data integrity constraints	5	2023-12-23	2023-12-27

1.9.1. Budget plan

The project has the following budgetary requirement as outlined as the table below

Table 4 Budget plan

No	Budget Item	Amount	Price for each (Birr)	Total Price
1	Paper	10 pt	5	50
2	Pen	5	20	100
3	Removal Flash	1	300	300
4	Mobile card	2	25	50
Total				500

Chapter 2

2.1 Database modeling: The ER model (diagram)

The following ER diagram shows the six entities that represent our project and the relationship that exists between these entities.

The entities in this project are the following:

1, Student

2, Instructor

3, Department

4, Course

5, Program

6. Schedule

And each of these entities have a relationship between them and a set of attributes that represent them.

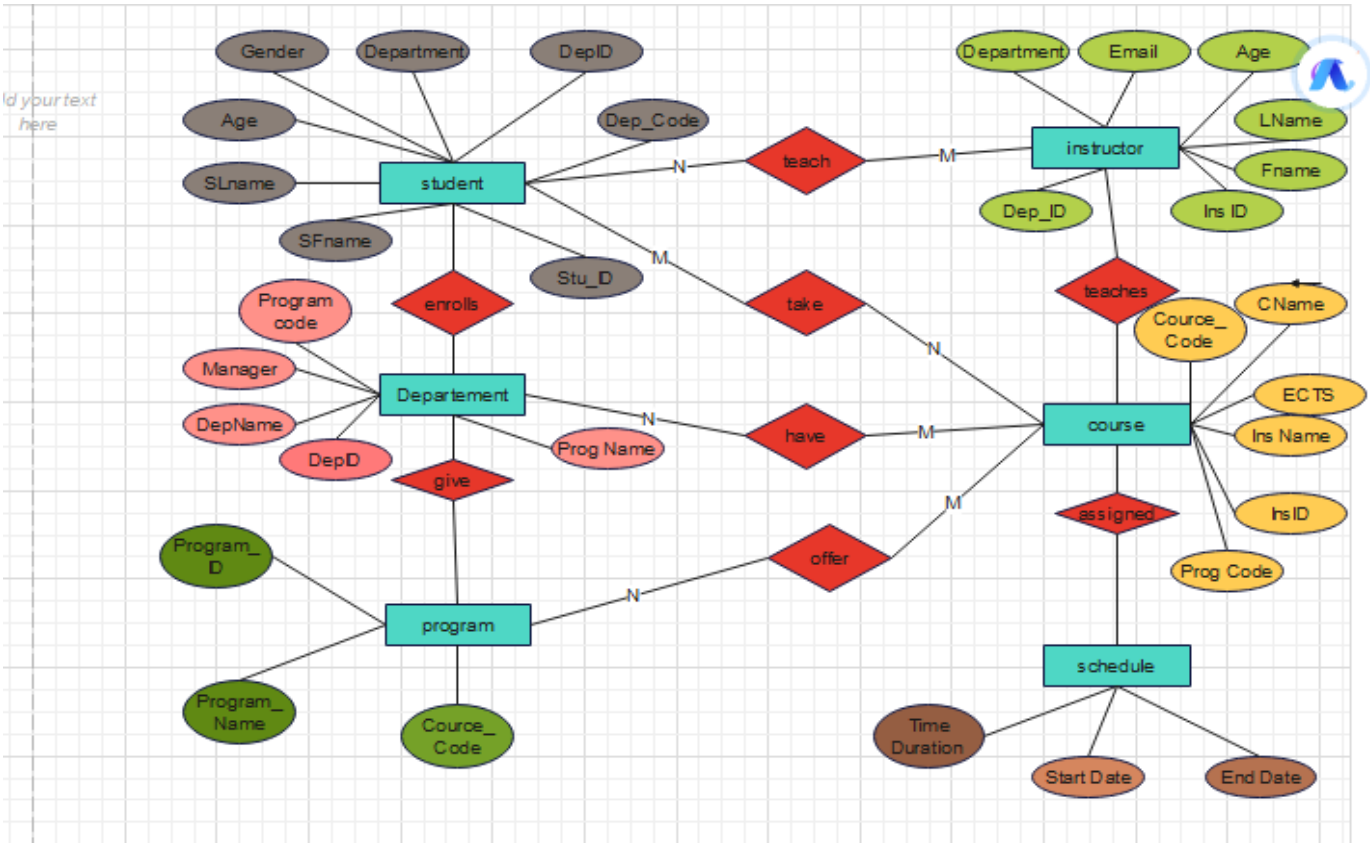


Figure 1 The ER model (diagram)

2.2. Database schema

As part of a database project for the RVU database management system we utilized the Entity-Relationship (ER) diagram to design the database schema and create tables that represent the relationships between the various entities in SQL format. The ER diagram helped us to visualize the relationships between the entities, which allowed us to create a well-structured database that meets the specific needs of the school.

By using SQL format to create the tables, we were able to implement the relationships between the entities in a way that is consistent and efficient. For example, we could define foreign keys to enforce referential integrity, which ensures that the relationships between the entities are maintained and data is not lost or corrupted.

Overall, the ER diagram and SQL format played a crucial role in the success of the database project for the RVU registrar database management system. The resulting database provides a solid foundation for managing the University data in a way that is organized, efficient, and easy to maintain.

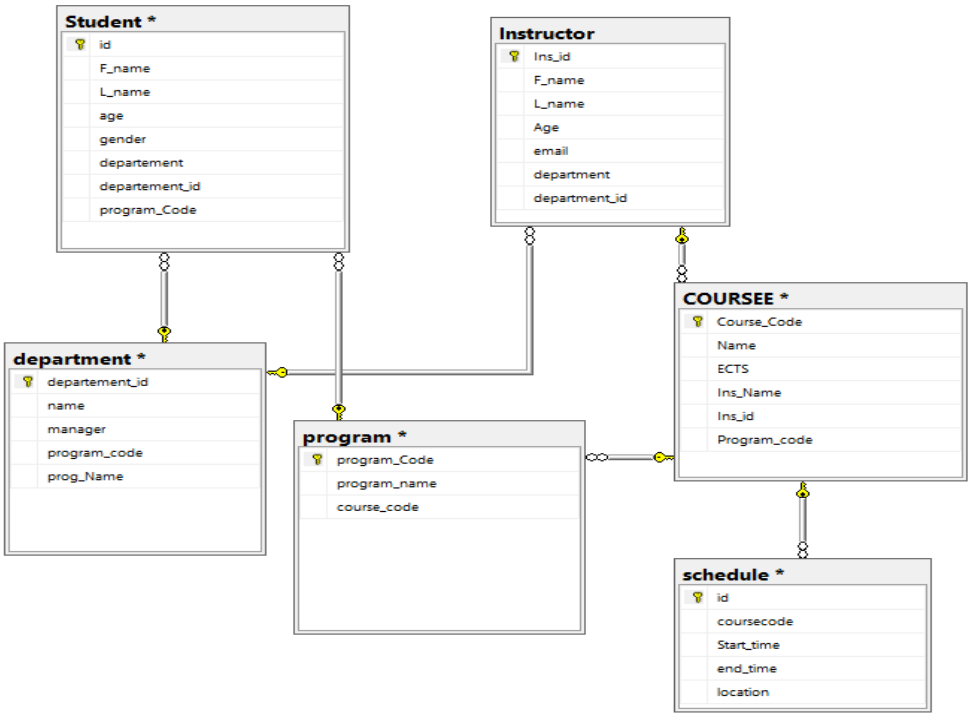


Figure 2 Schema of the new developed registrar debases of RUV

2.3. Database table

And after making both the ER diagram and the database schema we moved on to make the tables base on the information we have collected about each entities and the following list of tables show the entities as tables showing almost all the information we have gathered.

Table 5 Department table

	departement_id	name	manager	program_code	prog_Name
1	1	Information Techology	Jamal Y	2	Diploma
2	2	Accounting	Erandufa	3	Degree
3	3	Managemant	Kenasa	4	Diploma
4	4	Narsing	Dr.Tola	5	Degree

Table 6 course table

	Course_Code	Name	ECTS	Ins_Name	Ins_id	Program_code
1	1011	Emarging Technology	5	Jamal Y	22	2
2	1012	Global Traind	5	Milisha	23	3
3	1015	Mathemathics	5	Ashanafi	25	4
4	1016	English	5	Araba	26	5

Table 7 Instructor table

	Ins_id	F_name	L_name	Age	email	department	department_id
1	22	Jamal	Yusuf	35	jamal@jmail.com	IT	1
2	23	Milisha	Kadir	30	mili100@jmail.com	Accounting	2
3	25	Ashanafi	Tulu	40	ashanafi@jmail.com	Mathemathics	3
4	26	Araba	Aman	38	araba@jmail.com	English	4

Table 8 Program table

	program_Code	program_name	course_code
1	44	diploma	1011
2	55	Degree	1012
3	66	Masters	1013
4	77	level	1014

Table 9 schedule table

	id	coursecode	Start_time	end_time	location
1	1	1011	2023-11-29 08:00:00.000	2023-11-29 09:30:00.000	Room 102
2	2	1012	2023-11-29 08:00:00.000	2023-11-29 09:30:00.000	Room 103
3	3	1013	2023-11-29 08:00:00.000	2023-11-29 09:30:00.000	Room 104
4	4	1014	2023-11-29 08:00:00.000	2023-11-29 09:30:00.000	Room 105

Table 10 student table

	id	F_name	L_name	age	gender	departement	departement_id	program_Code
1	2344	Gadise	Abdi	24	F	Accounting	1	2
2	2345	Meseret	Dereje	25	F	Management	1	2
3	2346	Ayantu	Adam	24	F	Computer Science	1	2
4	2347	Kule	Abrahim	23	F	Management	1	2

2.4. Normalization

Normalization is a process of organizing data in a database to minimize redundancy and dependency. Normalization helps to improve data integrity, reduce data redundancy, and improve database performance. The normalization process involves dividing a database into two or more tables and defining relationships between the tables.

The first, second, and third normal forms are three stages of normalization that each table in a database should pass through to ensure that data is organized in a structured and efficient way. Here is an explanation of each of these normal forms:

1. First Normal Form (1NF):

In the first normal form, a table must have a primary key, and all columns in the table must be atomic. This means that each column in the table must contain only one value, and each value must

be unique. Additionally, the table should not contain repeating groups of data. If a table does not meet these requirements, it is not in first normal form.

2. Second Normal Form (2NF):

In the second normal form, a table must be in 1NF and have no partial dependencies. This means that each column in the table must be dependent on the entire primary key, and not just a part of it. If a table has columns that are dependent on only a portion of the primary key, it is not in second normal form.

3. Third Normal Form (3NF):

In the third normal form, a table must be in 2NF and have no transitive dependencies. This means that each column in the table must depend only on the primary key, and not on any other non-key columns. If a table has columns that are dependent on other non-key columns, it is not in third normal form.

Overall, the normalization process is important to ensure that data is well-organized and efficient, which helps to improve database performance and reduce errors. By following the three normal forms, you can ensure that your database is optimized for storing and retrieving data.

And we tried to normalize the following two tables :

Table 11 Student table normalized

PK FK

ID	F_NAME	L_NAME	AGE	GENDER	DEPARTEMENT	DEPARTEMENT_ID	PROGRAM_CODE
2344	Gadise	Abdi	24	F	Accounting	1	2
2345	Meseret	Dereje	25	F	Management	1	2
2346	Ayantu	Adam	24	F	Computer scince	1	2
2347	Kule	Abrahim	24	F	Management	1	2

This table is not normalized so the first normalized form is going to be :

To normalize the given table into first normal form (1NF), we need to identify repeating groups of data and remove them by creating separate tables. In this case, we can see that the DEPARTMENT and DEPARTMENT_ID columns repeat for each row, indicating a repeating group of data.

To remove the repeating group, we can create a separate table for DEPARTMENT and DEPARTMENT_ID, and use a foreign key in the original table to link the two tables. This results in two tables, one for STUDENT and another for DEPARTMENT:

Table 12 FN1 for the table student

ID	F_NAME	L_NAME	AGE	GENDER	DEPARTEMENT_ID
2344	Gadise	Abdi	24	F	1
2345	Meseret	Dereje	25	F	1

Table 13 FN1 for department

PK FK

DEPARTEMENT_ID	DEPARRTEMENT	PROGRAM_CODE
1	Computer science	2

In Table 1, we have removed the DEPARTMENT and DEPARTMENT_ID columns and added a DEPARTMENT_ID column as a foreign key to link the two tables. In Table 2, we have created a separate table for DEPARTMENT and DEPARTMENT_ID, and added the PROGRAM_CODE column.

With this normalization, we have eliminated the repeating groups of data, and each column in the tables contains atomic values. The resulting tables are now in first normal form (1NF).

To normalize the given tables to the second normal form (2NF), we need to ensure that each non-key column in the tables is fully dependent on the primary key.

In Table 1, we have a composite primary key consisting of the ID and DEPARTMENT_ID columns. However, the DEPARTMENT column is dependent only on the DEPARTMENT_ID column, and not on the entire composite key. Therefore, we need to split Table 1 into two tables to remove this partial dependency.

Table 14 FN2 for student

pk

FK

ID	F_NAME	L_NAME	AGE	GENDER	DEPARTEMENT_ID
2344	Gedisee	Abdii	24	F	1
2345	Meseret	Dereje	25	F	1

Table 15 FN2 for department

pk

FK

DEPARTEMENT_ID	DEPARRTEMENT	PROGRAM_CODE
1	Management	2

In Table 1, we have removed the DEPARTMENT column and created a new table for it. The DEPARTMENT_ID column remains in Table 1 as a foreign key to link to Table 2.

Now, both tables satisfy the requirements of the second normal form (2NF), with each non-key column fully dependent on the primary key. The PERSON table has a composite primary key consisting of ID and DEPARTMENT_ID, and all the columns in it depend on this composite key. Similarly, in the DEPARTMENT table, the DEPARTMENT_ID column is the primary key, and the DEPARTMENT and PROGRAM_CODE columns depend on it.

To normalize the given tables to the third normal form (3NF), we need to ensure that there are no transitive dependencies between non-key columns.

In Table 2, we can see that there is a transitive dependency between the DEPARTMENT and PROGRAM_CODE columns, as the PROGRAM_CODE column depends only on the DEPARTMENT column, and not on the primary key DEPARTMENT_ID. To remove this transitive dependency, we need to split Table 2 into two tables.

Table 16 FN3 for student

ID	F_NAME	L_NAME	AGE	GENDER	DEPARTEMENT_ID
2344	Gedise	Abdii	24	F	1
2345	Meseret	Dereje	25	F	1

Table 17 FN3 for department

DEPARTEMENT_ID	DEPARRTEMENT
1	Accounting

Table 18 FN3 for program

PROGRAM_CODE	PROGRAM_NAME
2	Degree

In Table 1, the DEPARTMENT_ID column remains as a foreign key to link to Table 2. In Table 2, we have removed the PROGRAM_CODE column and created a new table for it. The PROGRAM_CODE column is the primary key in Table 3, and the PROGRAM_NAME column is fully dependent on it.

With this normalization, we have eliminated all the transitive dependencies, and each non-key column in the tables depends only on the primary key. The resulting tables are now in third normal form (3NF).

2.5. Implementation

In this stage of our project we will try to show all the commands and syntaxes that we used to create this database schema and we will try to explain them as much as possible . here is an explanation of the syntax used to create a database called RVU Registrar DBMS database, as well as an explanation of the syntax used to create a table called "student":

1. . Syntax to create a database called RVU_REGISTERAL_DBMS_HARA_CAMPUS:

```

CREATE DATABASE RVU_REGISTERAL_DBMS_HARA_CAMPUS;

```

This syntax creates a new database called RVU_REGISTERAL_DBMS_HARA_CAMPUS, which can be used to store and manage data for the RVU Registrar DBMS database.

2. . Syntax to create a table called "student":

```
USE RVU_REGISTERAL_DBMS_HARAR_CAMPUS;
create table Student(
id int primary key,
F_name varchar(255),
L_name varchar(60),
age int,
gender varchar(10),
departement varchar(40),
departement_id int not null,
program_Code int not null)
```

This syntax creates a new table called "student" in RVU Registrar DBMS database. The table has several columns, including "id", "F_name", "L_name", "age", "gender", "department", "department_id", and "program_Code". The "id" column is the primary key for the table, which means it uniquely identifies each row in the table. The "F_name", "L_name", "age", "gender", "department", "department_id", and "program_Code" columns store information about each student, including their first and last names, age, gender, department, and program code. The "department_id" and "program_Code" columns have the "NOT NULL" constraint, which means that these columns cannot have null values. This ensures that each student is associated with a department and a program code.

Overall, this syntax creates a well-structured table that can store and manage data about the students at the RVU Registrar DBMS database.

3. Syntax to create a table called "Program":

```
USE RVU_Registrar_DBMS_HARAR_CAMPUS
create table program(
program_code int primary key,
program_name Varchar(50),
Course_name Varchar(35),
Course_code int not null)
```

This syntax creates a new table called "Program" in the RVU Registrar DBMS database. The table has several columns, including "program_Code", "Prog_name", "Course_name", and "CourseCode". The "program_Code" column is the primary key for the table, which means it uniquely identifies each row in the table. The "Prog_name" column stores the name of the program, while the "Course_name" column stores the name of the course associated with the program. The

"CourseCode" column has the "NOT NULL" constraint, which means that this column cannot have null values. This column stores the course code associated with the program.

Overall, this syntax creates a table that can store information about the programs offered by the RVU Registrar DBMS database.

4. Syntax to create a table called "department":

```
USE RVU_Registrar_DBMS_HARAR_CAMPUS;
create table departement(
departemet_id int primary key,
name varchar(50),
manager varchar(50),
program_name varchar(40),
program_code int NOT NULL)
```

This syntax creates a new table called "department" in the RVU Registrar DBMS database. The table has several columns, including "departement_id", "name", "manager", "prog_Name", and "program_Code". The "departement_id" column is the primary key for the table, which means it uniquely identifies each row in the table.

The "name" column stores the name of the department, while the "manager" column stores the name of the department manager. The "program_Code" column has the "NOT NULL" constraint, which means that this column cannot have null values. This column stores the program code associated with the department.

Overall, this syntax creates a table that can store information about the departments at the RVU Registrar DBMS database.

5. Syntax to create a table called "Instructor":

```
USE RVU_Registrar_DBMS_HARAR_CAMPUS;
create table instructor(
ins_id int primary key,
F_name varchar(45),
L_name varchar(50),
Age INT,
email varchar(40),
departement varchar(50),
departement_id int not null,)
```

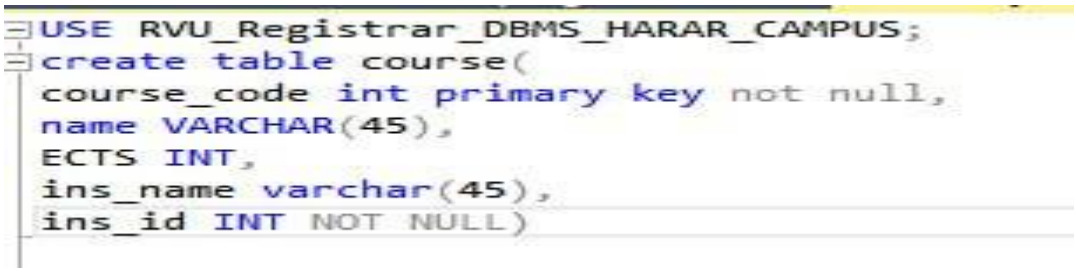
This syntax creates a new table called "Instructor" in the RVU Registrar DBMS database. The table has several columns, including "Ins_id", "F_name", "L_name", "Age", "email",

"department", and "department_id". The "Ins_id" column is the primary key for the table, which means it uniquely identifies each row in the table. The "F_name" column stores the first name of the instructor, while the "L_name" column stores the last name of the instructor. The "Age" column stores the age of the instructor, while the "email" column stores the email address of the instructor.

The "department" column stores the name of the department associated with the instructor, and the "department_id" column has the "NOT NULL" constraint, which means that this column cannot have null values. This column stores the department ID associated with the instructor.

Overall, this syntax creates a table that can store information about the instructors at the RVU Registrar DBMS database

6. Syntax to create a table called "course":



```
USE RVU_Registrar_DBMS_HARAR_CAMPUS;
create table course(
course_code int primary key not null,
name VARCHAR(45),
ECTS INT,
ins_name varchar(45),
ins_id INT NOT NULL)
```

This syntax creates a new table called "course" in the RVU Registrar DBMS database. The table has several columns, including "CourseCode", "Name", "ECTS", "Ins_Name", and "Ins_id". The "CourseCode" column is the primary key for the table, which means it uniquely identifies each row in the table. This column has the "NOT NULL" constraint, which means that it cannot have null values. The "Name" column stores the name of the course, while the "ECTS" column stores the European Credit Transfer and Accumulation System (ECTS) value of the course.

The "Ins_Name" column stores the name of the instructor teaching the course, and the "Ins_id" column has the "NOT NULL" constraint, which means that this column cannot have null values. This column stores the ID of the instructor teaching the course.

Overall, this syntax creates a table that can store information about the courses offered by the RVU Registrar DBMS database

7. Syntax to create a table called "schedule":


```

This syntax creates a new table called "schedule" in the RVU Registrar DBMS database. The table has several columns, including "id", "courseCode", "start_time", "end_time", and "location". The "id" column is the primary key for the table, which means it uniquely identifies each row in the table. The "courseCode" column has the "NOT NULL" constraint, which means that it cannot have null values. This column stores the course code associated with the scheduled class. The "start_time" and "end_time" columns store the start and end times of the class, respectively.

The "location" column stores the location of the class.

Overall, this syntax creates a table that can store information about the class schedules for the courses offered by the RVU Registrar Database management System.

# Chapter 3

## 3.1, Conclusion and Recommendation:

### 3.1.1. Conclusion

In conclusion, the implementation of a new registrar database system for Rift Valley University (RUV) is essential for effective data management and informed decision-making. This project aimed to streamline processes related to student records, course management, faculty information, and resource allocation. By adopting a well-designed database system, RUV can significantly improve data accuracy, accessibility, and operational efficiency.

The new registrar database system enables easy tracking of student progress, providing valuable insights into their academic journey and performance. It facilitates efficient organization of course schedules, preventing conflicts and ensuring smooth class timetabling. Moreover, the system effectively manages faculty workload, ensuring fair distribution of responsibilities and optimal allocation of teaching and research duties.
```

Additionally, the database system plays a vital role in resource management, allowing for effective tracking and allocation of classrooms, laboratory equipment, and other essential materials. This optimization minimizes resource wastage and enhances overall operational efficiency.

By embracing the proposed registrar database system, RUV can achieve improved data management, enhanced accessibility, and increased efficiency in their operations. The system empowers students, faculty, and staff with a seamless experience, facilitating informed decision-making and enabling RUV to fulfill its educational mission effectively.

In summary, the implementation of the new registrar database system is crucial for RUV's success in managing student records, course management, faculty information, and resource allocation. It provides a solid foundation for efficient data management and lays the groundwork for continuous improvement and growth.

3.1.2. Recommendation

Based on this project overall works and achievement we recommends the following key points to be done in the future to improve the existing developed student registrar database system of Harar campus RVU:

I. **Testing the new proposed database system with existing Data:** It is recommended to conduct thorough testing of the proposed student registrar database system using the available data at RVU. This testing will help assess the system's compatibility with the existing data and identify any issues or discrepancies that need to be addressed.

II. **Collecting additional data for a comprehensive database:** Since the proposed registrar database system is based partially on primary data collected from few RVU staff and mostly based on secondary data, it is recommended to gather additional data to create a more comprehensive and accurate database that reflect the reality. This will ensure that the system reflects the reality of RVU's student information and meets the required standards.

III. **Evaluating and enhancing the proposed conceptual schema:** The conceptual schema of the proposed registrar database system should be thoroughly checked and evaluated to ensure it meets the necessary requirements. If any shortcomings are identified, improvements should be made to the existing schema to align it with the desired functionality and data management needs.

IV. **Testing the compatibility and user-friendliness:** Before implementing the proposed new registrar database management system, it is crucial to assess its compatibility with existing software applications and evaluate its user-friendliness based on the capabilities and familiarity of the existing staff. This testing will help identify any potential issues or barriers that may hinder the system's successful adoption and use.

Therefore, these recommendations could improve the newly proposed student registrar database system, hence the RVU can enhance the accuracy, efficiency, and usability of its data management processes. Additionally, addressing these areas will ensure that the system is compatible with existing software, reflects the needs of the staff, and ultimately contributes to an effective and streamlined student registrar system at RVU.

4 References

1. Teaching module for this courses
2. DBMS Tutorial - Database Management System - GeeksforGeeks
3. What is a DBMS? Database Management System Definition (techtarget.com)
4. https://www.techtarget.com/searchdatamanagement/definition/database-management-system
5. Database Management Systems (DBMS): The Beginner's Guide | Splunk
6. What Is a Database Management System? And Why You Need One | Capterra
7. https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system
8. What Is Database Management & DBMS Benefits | Nutanix
9. https://www.techopedia.com/definition/24361/database-management-systems-dbms
10. https://www.appdynamics.com/topics/database-management-systems

5. Appendix

Questionnaires prepared for Developing a RVU Registrar Database Management System

This questioner is prepared by the second year computer science CEP students. The feedback from this will be used for developing a database management system (DBMS) for the registrar's office. Furthermore, this questionaries' is considered as the essential step in understanding the requirements and needs of users for developing a database management system (DBMS) for the registrar's office. Therefore, your response will be kept anonymously and genuine.

1. **General Information:**
   a. Institution Name:
   b. Registrar department head name
   c. Years of experience in registrar office:
   d. Location
   e. Sex of the respondent
   f. Academic Position and Role
2. **Current database system**
   a. How student information currently is managed in your university?
   b. What are the main challenges or limitations of the current system?
   c. Are there any specific features or functionalities lacking in the current system?
   d. Do you currently use a database management system (DBMS) in your office? (Yes/No)
   e. If yes, which DBMS is it? (e.g., MySQL, Oracle, Access)

f. What are the strengths and weaknesses of your current system?
g. What tasks or functions are currently handled by the system?

3. **Data Entry and Retrieval:**
   a. How often is data entered and updated?
   b. What types of information do you find most crucial for day-to-day operations?
   c. What kind of reports or data retrieval functionalities would be most useful to you?
   d. What types of data are currently stored in the system? (e.g., student information, course information, enrollment data, grades)
   e. What are the key data fields for each data type?
   f. What are the reporting requirements for your office? (e.g., transcripts, enrollment reports, student lists)
4. **User Access, Role and Permissions:**
   a. Who are the primary users of the registrar database? (e.g., registrars, faculty, students, advisors)
   b. What level of access does each user role need? (e.g., view, create, edit, delete)
   c. Are there specific permissions or restrictions for certain data?
   d. What level of access permissions do different roles within the registrar's office require?
   e. Are there any specific security measures that need to be implemented?
5. **Integration with Other Systems:**
   a. Are there other systems or databases that the new DBMS should integrate with?
   b. What kind of data sharing or synchronization is required between systems?
6. **Data Validation and Quality:**
   a. How is data currently validated and ensured for accuracy?
   b. What kind of data quality measures do you think should be incorporated into the new system?
7. **Reporting and Analytics:**
   a. What types of reports and analytics would be beneficial for your department?
   b. Are there any specific key performance indicators (KPIs) that you track regularly?
8. **Training and Support:**
   a. What level of training do you think is necessary for users to effectively use the new system?
   b. What kind of ongoing support or documentation would be helpful?
9. **Future Growth and Scalability:**
   a. Are there any anticipated changes or expansions in the registrar's office that the DBMS should accommodate?
   b. How scalable should the system be to handle future growth?
   c. What are your anticipated needs for the database system in the next 3-5 years?
   d. Are there any new functionalities you would like to see implemented?
   e. What are your expectations for the system's performance and scalability?
10. **Preferences and Additional Comments:**
    a. Do you have any specific preferences or expectations for the user interface?
    b. Any additional comments or requirements that haven't been covered?
    c. Please feel free to share any other information or feedback you think would be helpful in developing the new database management system.