



SEEK WISDOM, ELEVATE YOUR INTELLECT AND SERVE HUMANITY !



DATABASE PROJECT

FOR ASCO PROGRESS ACADEMY

AAU CNCS – COMPUTER SCIENCE DEPARTMENT

GROUP MEMBERS

<i>No</i>	Name	ID No
1	Bersabeh Degafu	UGR/2049/15
2	Eden Awoke	UGR/2222/15
3	Eden Ephrem	UGR/6388/15
4	Hana Hilekiros	UGR/5655/15
5	Hanan Abdulshikur	UGR/6257/15
6	Hanan Fatih	UGR/6009/15

Submitted to: Ins. Fikerte

Submission date: 5/14/2024

CONTENTS

Group members	1
1. Introduction	3
Background.....	3
Purpose	3
Statement of problem.....	3
Scope and objective	3
Objective of the Project	3
General objective	4
Specific Objective	4
Database Development Methodology.....	4
Data Sources & Collection Methods	4
DB Analysis and Design Methods.....	4
Deliverable of the Project	5
Development Tools, Platforms, and Technologies	Error! Bookmark not defined.
Project Time Plan.....	5
2. Requirement Specification	5
Data requirments.....	5
Transaction Requirements	6
Data Entry Requirements	6
Data Retrieval Requirements	7
Data Updating Requirements	7
Data Removal Requirements	7
3. Database design.....	7
Conceptual database design	7
Entity and attributes	7
Relationships between the entities	10
E-R diagram	10
Logical Database Design	11
ER-Relation Mapping.....	11
4. Implementation and Testing	12
5. References and websites used.....	20

1. INTRODUCTION

BACKGROUND

Through our choice of organization, we were able to see that the management of the school lacks up to date efficiency hence we decided to solve this problem by designing an Active school management system. The Active School Management System (ASMS) goes beyond basic tools, focusing on learning, instruction and empowers teachers, administrators, students, and parents. Over all it is:

A digital platform replaces paperwork, saving time and minimizing errors. Automated functions streamline tasks like enrollment and attendance.

Where Teachers gain a platform for planning, managing resources, and communicating.

An administrative tool to facilitate attendance tracking, performance analysis, and parent communication. Integration with other platforms builds a holistic student support system.

ASMS isn't just management; it's an investment in a more efficient, accurate, and empowering learning experience for everyone.

PURPOSE

Even though the system we designed tackle the school data management problems from different perspective; the two main purposes are as follows:

1. **Reduce Errors and Improve Efficiency:** The system aims to minimize errors in student record management and streamline administrative tasks. This likely involves automating data entry and calculations, reducing reliance on manual processes prone to human error.
2. **Enhance User Experience:** By optimizing data handling and minimizing manual work, the system seeks to improve the overall experience for administrators, teachers, students, and potentially parents. This could mean faster access to accurate information, reduced time spent on paperwork, and easier communication within the school system.

STATEMENT OF PROBLEM

On our research of the school's background, we found out how ineffective and tiresome the manual data management system they were using was. As confirmed from the school's administrator and officials:

- ❖ The search of an individual's record might take hours, often going missing.
- ❖ Handling fee payment checking and updates might take a labor and is prone to error.
- ❖ These are one of the few problems the school faces from the manual management system.

SCOPE AND OBJECTIVE

The scope of the system ranges from managing information on an individual student to teachers to determine the most crucial functionalities for the school. We can see the objective of the system as specific and genera objectives.

OBJECTIVE OF THE PROJECT

GENERAL OBJECTIVE

- ❖ **Enhance efficiency and accuracy in school operations:** This broad objective encompasses all the ways the SMS will streamline processes and reduce errors.
- ❖ **Improve the user experience:** This objective focuses on making the system easy to use and beneficial for different user groups.
- ❖ **Reduce time and cost:** making the management system cost effective in low duration.

SPECIFIC OBJECTIVE

- ❖ **Reduce administrative burden:** by automating tasks like enrollment, attendance tracking, and report generation.
- ❖ **Decrease data entry errors:** through data validation and automated calculations.
- ❖ **Improve data accessibility for all authorized users** by creating a centralized and secure data storage system.
- ❖ **Empower administrators with data-driven decision-making:** by providing real-time reports on student performance, attendance, and enrollment trends.
- ❖ **Increase communication and collaboration** by providing secure communication channels between teachers, administrators, students, and parents.

DATABASE DEVELOPMENT METHODOLOGY

For this project, a structured database development methodology will be followed. This methodology will include the following phases:

1. **Requirements gathering:** Identify the specific needs and requirements of the school regarding students, teachers and resources.
2. **Conceptual design:** Create a high-level conceptual model that captures the essential entities, relationships, and attributes relevant to school.
3. **Logical design:** Translate the conceptual model into a detailed logical model, including tables, keys, and relationships.
4. **Physical design:** Implement the logical model in a database management system, considering performance, scalability, and security requirements.
5. **Implementation:** Populate the database with school data and develop the necessary algorithms and functionalities for data analysis.
6. **Testing and validation:** Verify the system's accuracy, reliability, and performance through rigorous testing and validation procedures.

DATA SOURCES & COLLECTION METHODS

The school data will be collected from various sources, including:

1. Our data sources are teachers, students and other school communities
2. Surveys: Feedback and preferences collected directly from students and school communities through surveys and feedback forms.
3. By observing the previous file-based data management system

DB ANALYSIS AND DESIGN METHODS

To analyze and design the database system, a combination of entity-relationship modeling and normalization techniques will be utilized. Entity-relationship diagrams will help capture the

relationships between entities, while normalization will ensure the database's efficiency and integrity.

DELIVERABLE OF THE PROJECT

The deliverable of this project will include:

1. school database system: A fully functional and scalable database system specifically designed for school.
2. Data collection methods: Mechanisms to collect school data from various sources and integrate it into the database system.
3. Data analysis algorithms: Algorithms and techniques to extract meaningful insights from the school data.

DEVELOPMENT TOOLS, PLATFORMS AND TECHNOLOGY

The development of the school database system will utilize the following tools, platforms and technologies.

1. Relational database management system for data storage and management.
2. Programming language for Implementing data collection, analysis and visualization functionalities.

PROJECT TIME PLAN

The project will be executed according to the following time plan:

1. Requirements gathering: 1 Week
2. Conceptual design: 2 Days
3. Logical design: 2 Days
4. Physical design: 1 Day
5. Implementation: 2 Days
6. Testing and validation: 1 Day

2. REQUIREMENT SPECIFICATION

ENTERPRISE CONSTRAINT

When we were gathering information from the school, we have observed the following constraints:

- One teacher must teach only one Subject.
- And one classroom is specified to one grade level
- For a student to pass from one grade level to other grade level he/she must get more than 50%.
- ✚ Asco progress school have three branches. Each branch has an address, location, involved grades and branch id as a unique identifier.
- ✚ Each branch has one director and one director is assigned to only one branch. Director has a name , address, salary and director id as a unique identifier.

- ✚ One branch constitutes many students. Student has a name, date of birth, date of enrollment, age , gender, branch, classroom, grade level and student id as unique identifier. For a student to be promoted to the next grade level an average of above 50% must be scored.
- ✚ Many students with the same grade level attends in a one fixed classroom. Each classroom has a capacity of students it can hold, type and class number as a unique identifier.
- ✚ Multiple teachers teach many students. Teacher has a name, address, department, salary, level of education and teacher id as a unique identifier. One teacher can teach only one subject.
- ✚ One subject can be taught by multiple teachers. Subject has department it belongs to , name and subject code as a unique identifier. Many students take many subjects.
- ✚ Students pay payments for different purposes. Payment includes amount of birr, grade level and student id to identify who made the payments.
- ✚ Students have multiple guardians. Guardians have name,phone number, location, realtion to student and guardian id as a unique identifier.

DATA REQUIRMENTS

The database system for the school must be capable of managing and storing the following data:

- Student: name, grade level, date of birth, age, gender...
- Teacher: name, department, salary, education level...
- School: name, location, school id, capacity...
- Director: name, address, salary...
- Branch: location, branch id, grades involved
- Classroom: class number, type, equipment, capacity...

TRANSACTION REQUIREMENTS

The database system should support the following transaction operations:

- Recording newly enrolled students' information: Capture details of student's name, educational background, guardian information and payment information etc.
- Updating student, teacher and staff information: Allow modifications to payments required by students and adjustment of salary such as adjusting quantities, applying discounts, or processing returns.
- Terminate records- in case a student or a teacher resides.

DATA ENTRY REQUIREMENTS

The system should facilitate efficient and accurate data entry through the following means:

- Integration of data across every branch at the point of entry of data.
- Depending on the data type staff members working in different offices facilitate data entry. For example registrar captures information such as student info and the cashier enters payment related info into the database.

DATA RETRIEVAL REQUIREMENTS

The database system should enable the retrieval of data to support decision-making and reporting. The following retrieval requirements should be met:

- Student analysis: Retrieve information of student name, grade level, date of birth, age, gender, branch and class room.
- Teacher analysis: Obtain teachers name, address, department and salary.
- Guardian information: name, id, phone number, location.
- Classroom information: class number, class type, capacity.

DATA UPDATING REQUIREMENTS

The system should allow for efficient and accurate data updating, including:

- Modifying teachers and directors' salary.
- Updating student payment.
- Updating student grade level.

DATA REMOVAL REQUIREMENTS

The system should provide the capability to remove or delete data when necessary, including:

- Terminate student and teacher records when they reside.

3. DATABASE DESIGN

CONCEPTUAL DATABASE DESIGN

ENTITY AND ATTRIBUTES

The conceptual database design for the new system will involve identifying the key entities and their attributes, as well as understanding the relationships between these entities.

❖ Entities and Attributes with Descriptions:

a. **Student:** the person that gets to school to get education

- Student_id(primary key) - each student identifier
- name
 - First_name - student first name
 - Last_name - student last name
- Grade_level - which grade level the student is enrolled to

- DOE - on which year doe the student enrolled
- DOB - which date is the student birth
- Gender - student gender
- Age- age of the student drived from date of birth

b. **Teacher:** teaches the students

- Teacher_id(primary key) - identifier for teacher
- name
 - First_name - teacher first name
 - Last_name -teacher
- address
 - Phone_no - teacher Phone_no
 - Email - teacher email address
- Salary - teacher salary per month
- Department - to whihch department does the teacher belongs
- degree - what degrees does the teacher have(multi-valued attribute)

c. **Director:** leads every branch of the school

- Director_id(primary key) - Director identifier
- name
 - First_name - Director first name
 - Last_name - Director last name
- Salary - Director salary
- address
 - Phone_no - Director Phone_no
 - Email - Director email address

d. **Payment:** is the school fee and the transportation payment that students pay

- Payment_reason - for what thing the student paid (eg, school fee, transportation fee, after class fee)
- Amount - amount paid

- Grade_level - to which grade level the student belong
- Penalty – amount of money paid for delay of regular payments
- e. **Guardian:** are people who teach their students in that school
 - Guardian_id(primary key) - Guardian identifier
 - First_name - first name of Guardian
 - Last_name - last name of Guardian
 - Phone_no - Guardian Phone_no
 - Location - location of the Guardian
- f. **Subject:** is the thing that students learn
 - Subject_code(primary key) - Subject identifier
 - Name - subject name
 - Department - to which department does it belong to
- g. **Classroom:** is where the students learn
 - class_number(primary key) - Classroom identifier
 - type - for what purpose does the Classroom is used
 - Capacity - how many student can the Classroom hold
- i. **Branch:** it is the branches that the school have at different place
 - Branch_id(primary key) - Branch identifier
 - Location
 - woreda
 - kebele
 - grades_involved - what what grades are in that branch
 - address
 - Phone_no - Phone no of the branch
 - Email - branch email address
 - Teacher_no – The number of teachers that teaches in that branch
 - Student_no – Number of students that learn in that branch

- Classroom_no – Number of classes in that branch

RELATIONSHIPS BETWEEN THE ENTITIES

- Student and branch=> N :1
- students and teachers => N: N
- students and Classroom => N:1
- students and Guardian => N: N
- students and payments => 1: N
- teachers and subjects => N: 1
- students and subjects => N: N
- students and Classroom => N:1
- students and Guardian => N: N
- director and Branch => 1:1

E-R DIAGRAM

The ER diagram presented below illustrates the relational structure of the database, depicting the entities and their relationships in a clear and concise manner. This diagram offers a comprehensive overview of the database schema, facilitating a deeper understanding of the data model and its components. Through intuitive visual representation, stakeholders can easily grasp the interconnections between various entities and attributes, aiding in the development and maintenance of the database system. We have tried to show the E-R diagram both by *chen style* and *UML style*. We have used Draw.io and Lucid chart to draw the diagrams.

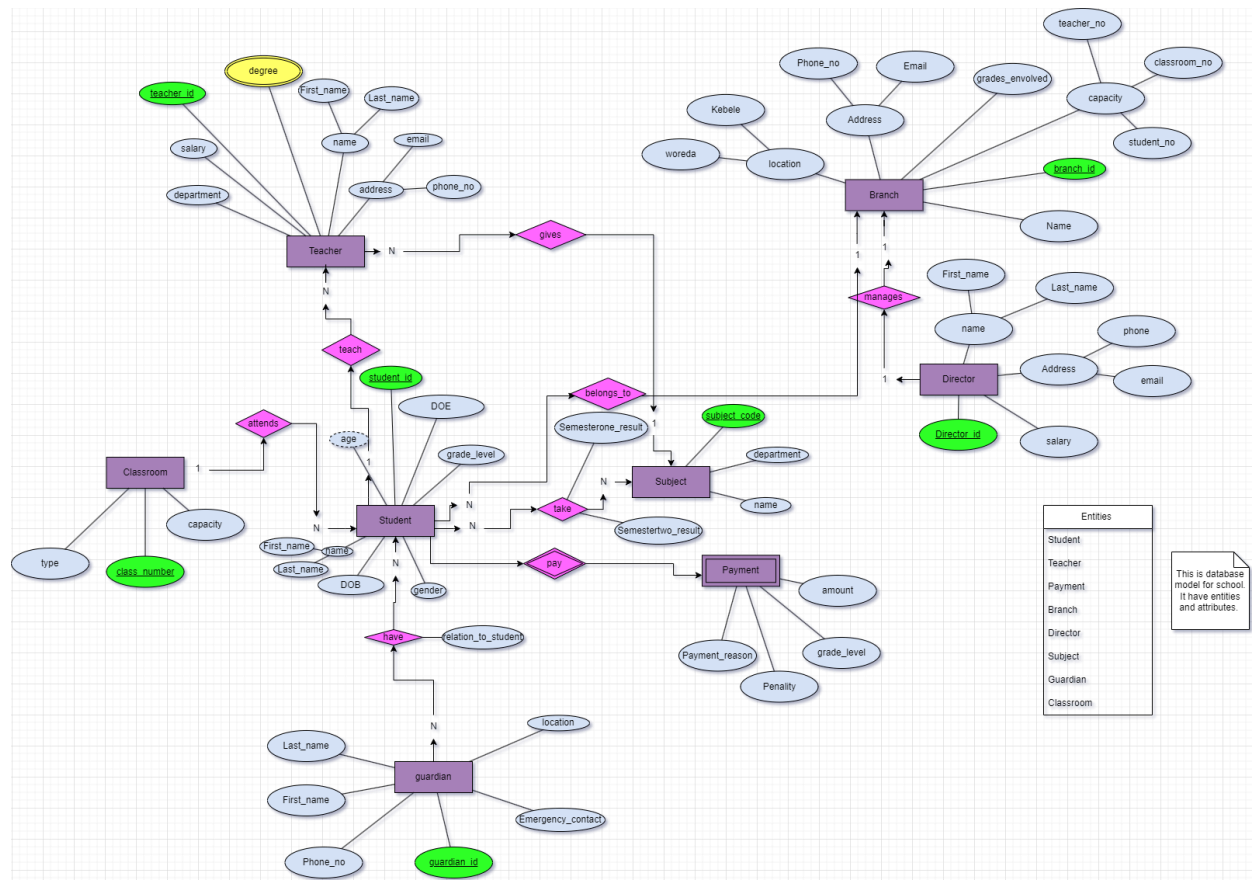


Fig 1: ER diagram for the school database. using chen style

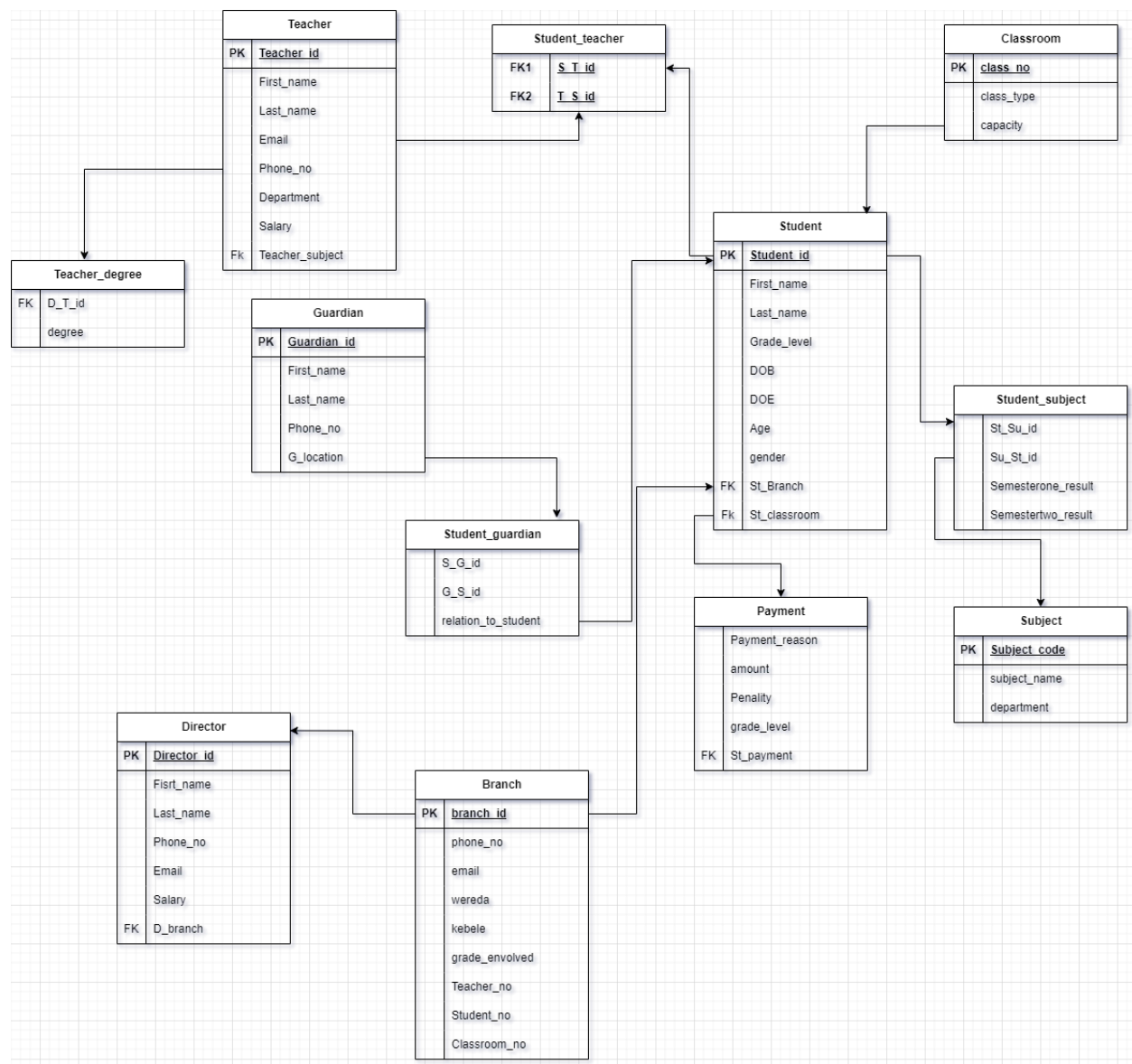


FIG 2: CLASS DIAGRAM OF THE SCHOOL DATABASE. USING THE UML STYLE

LOGICAL DATABASE DESIGN

ER-RELATION MAPPING

At this stage, the ER diagram transitions into the logical design of the database, culminating in the presentation of the refined database schema depicted below. This transformation involves converting the conceptual representation of entities and relationships into a more concrete and structured format. The logical design provides a blueprint for organizing and implementing the database,

capturing the essential elements and their interdependencies. The database schema serves as a fundamental framework, outlining tables, attributes, and their relationships, setting the stage for the subsequent steps in database development and implementation.

Branch

<u>Branch_id</u>	Name	Phone_no	Email	Wereda	Kebele	Grade_involved	Teacher_no	Class_no	Student_no
------------------	------	----------	-------	--------	--------	----------------	------------	----------	------------

Director

<u>Director_id</u>	First_name	Last_name	Phone_no	Email	Salary	Branch_id
--------------------	------------	-----------	----------	-------	--------	-----------

Student

<u>Student_id</u>	First_name	Last_name	G_level	DOB	DOE	Age	Gender	St_branch	St_class
-------------------	------------	-----------	---------	-----	-----	-----	--------	-----------	----------

Teacher

<u>Teacher_id</u>	First_name	Last_name	Email	Phone_no	Department	Salary	Teacher_subject
-------------------	------------	-----------	-------	----------	------------	--------	-----------------

Student_Teacher

<u>Student_id</u>	Teacher_id
-------------------	------------

Degree

<u>Teacher_id</u>	degree
-------------------	--------

Subject

<u>Subject_code</u>	Department	Subject_name
---------------------	------------	--------------

Guardian

<u>Guardian_id</u>	First_name	Last_name	Phone_no	G_location
--------------------	------------	-----------	----------	------------

Student_Guardian

<u>Student_id</u>	<u>Guardian_id</u>	Relation_to_student
-------------------	--------------------	---------------------

Classroom

<u>Class_id</u>	Class_type	Capacity
-----------------	------------	----------

Payment

<u>Student_id</u>	<u>Payment_reason</u>	Amount	G_level
-------------------	-----------------------	--------	---------

Student_subject

<u>Student_id</u>	<u>Subject_code</u>	Student_result
-------------------	---------------------	----------------

4. IMPLEMENTATION AND TESTING

SQL SCRIPT FOR CREATING THE DATABASE, TABLES AND VIEWS.

Student_id	First_name	Last_name	Grade_level	DOB	DOE	Age	Gender	St_Branch	St_classroom	
Branch_id	Name	Phone_no	Email	Wereda	Kebele	Grade_involved	Teacher_no	Student_no	Classroom_no	
Teacher_id	First_name	Last_name	Email	Phone_no	Department	Salary	Teacher_subject			
Class_no	Class_type	Capacity								
Guardian_id	First_name	Last_name	Phone_no	G_Location						
St_payment	Payment_reason	Amount	Penalty	Grade_level						
S_G_id	G_S_id	Relation_to_student								
St_Su_id	Su_St_id	Semesterone_result	Semestertwo_result							
S_T_id	T_S_id									
Subject_code	Subject_name	Department								
D_T_id	Degree									
Director_id	First_name	Last_name	Phone_no	Email	Salary	D_branch				

-----Inserting sample values to the tables for testing-----

1. -----Inserting value to Branch table-----

insert into Branch values

('Berchekobbranch', '+251909632541', 'AscoProgressAcademy@gmail.com', 'Kolfe', 39, 9,30,1000,25)

2. -----Inserting value to Director table-----

insert into Director values (1,'Mulugeta','Desalegn','+251960235689', 'MuleD@gmail.com', 18000, 'Bercheko branch')

3. -----Inserting value to Student table-----

insert into Student **values**

```
('ST00001','Nathan','Kiros',11,'11/16/1999','12/12/2015',17,'Male','Berekeko branch','11A'),  
  
('ST00002','Kaleb','Girma',8,'08/21/2003','12/10/2015',13,'Male','Berekeko branch','8A'),  
  
('ST00003','Helen','Girma',12,'06/29/1998','12/10/2015',18,'Female','Berekeko branch','12A')  
  
('ST00004','Umeran','Afework',7,'05/25/2004','12/11/2015',12,'Female','Berekeko branch','7B'),
```

4. -----Inserting value to Teacher table-----

insert into Teacher **values**

```
('TR001','Wasihun','Tessema','WasT@gmail.com','+251965458798','Science',10000, 'MAT012'),  
  
('TR002','Azeb','Desalegn','Azeb-Desalegn56@gmail.com','+251911801512',null,10000, 'ECO125'),  
  
('TR003','Nuredin','Jemal','Nure-Jemal@gmail.com','+251911111619','Mathematics',10000,  
'SPR056'),
```

5. -----Inserting value to Teacher_degree table-----

```
insert into Teacher_degree values ('TR001','Bachelor of science in Chemistry'),  
  
('TR002','Bachelor of science in Psychology'),  
  
('TR003','Bachelor of science in Mathematics'),
```

6. -----Inserting value to Guardian table-----

```
insert into Guardian values ('G000101','Girma','Negasa','+251911266325','Asco'),  
  
('G000102','Afework','Abebe','+251965789532','Asco'),  
  
('G000103','Tsege','Tadesse','+251911101523','Gulele')
```

7. -----Inserting value to Payment table-----

insert into Payment **values**

('ST00002','School fee',1500,0,'8A'),
 ('ST00002','Transportation fee',800,0,'8A'),
 ('ST00002','After class fee',200,0,'8A'),
 ('ST00003','School fee',1850,0,'12A'),

8. -----Inserting value to Classroom table-----

insert into Classroom **values**

('7A','Learning',50),('8A','Learning',50),('11A','Learning',50),('12A','Learning',50)

9. -----Inserting value to Student_Guardian table-----

insert into Student_guardian **values** ('ST00002','G000101','Father'),

('ST00003','G000101','Father'),

('ST00004','G000102','Father'),('ST00006','G000103','Mother'),

10. -----Inserting value to Student_teacher table-----

insert into Student_teacher **values**

('ST00001','TR001'),('ST00001','TR002'),('ST00001','TR003'),('ST00001','TR004'),('ST00001','TR005')

11. -----Inserting value to Student_subject table-----

insert into Student_subject **values**

('ST00003','MAT012',97.5,96.3),('ST00003','SPR056',99.9,81),('ST00010','MAT012',92.3,92),

12. -----Inserting value to Subject table-----

insert into Subject **values**

('MAT012','Mathematics','Mathematics'),('SPR056','Sport','Sport'),('ECO125','Economics','Social Science')

-----The Out Puts Would Be-----

1) ----- Table Branch -----

	Branch_id	Name	Phone_no	Email	Wereda	Kebele	Grade_involved	Teacher_no	Student_no	Classroom_no
1	Bercheko branch	Asco Progress Academy	+251909632541	AscoProgressAcademy@gmail.com	Kolfe	39	9	30	1000	25

2) -----Table Director -----

	Director_id	First_name	Last_name	Phone_no	Email	Salary	D_branch
1	1	Mulugeta	Desalegn	+251960235689	MuleD@gmail.com	18000	Bercheko branch

3) ----- Table Student -----

	Student_id	First_name	Last_name	Grade_level	DOB	DOE	Age	Gender	St_Branch	St_classroom
1	ST00001	Nathan	Kiros	11	1999-11-16	2015-12-12	17	Male	Bercheko branch	11A
2	ST00002	Kass	Chirs	8	2003-08-21	2015-12-10	13	Male	Bercheko branch	8A
3	ST00003	Helen	Girma	12	1998-06-29	2015-12-10	18	Female	Bercheko branch	12A
4	ST00004	Umeran	Afework	7	2004-05-25	2015-12-11	12	Female	Bercheko branch	7A

4) ----- Table Teacher -----

	Teacher_id	First_name	Last_name	Email	Phone_no	Department	Salary	Teacher_subject
1	TR001	Wasihun	Tessema	WasT@gmail.com	+251965458798	Science	10000	MAT012
2	TR002	Azeb	Desalegn	Azeb-Desalegn56@gmail.com	+251911801512	NULL	10000	ECO125
3	TR003	Nuredin	Jemal	Nure-Jemal@gmail.com	+251911111619	Mathematics	10000	SPR056

5) ----- Table Teacher_degree -----

	D_T_id	Degree
1	TR001	Bachelor of science in Chemistry
2	TR002	Bachelor of science in Psychology
3	TR003	Bachelor of science in Mathematics

6) ----- Table Guardian -----

	Guardian_id	First_name	Last_name	Phone_no	G_Location
1	G000101	Girma	Negasa	+251911266325	Asco
2	G000102	Afework	Abebe	+251965789532	Asco
3	G000103	Tsege	Tadesse	+251911101523	Gulele

7) ----- Table Payment -----

	St_payment	Payment_reason	Amount	Penalty	Grade_level
1	ST00002	After class fee	200	0	8A
2	ST00002	School fee	1500	0	8A
3	ST00002	Transportation fee	800	0	8A
4	ST00003	School fee	1850	0	12A

8) ----- Table Classroom -----

	Class_no	Class_type	Capacity
1	11A	Learning	50
2	12A	Learning	50
3	7A	Learning	50
4	8A	Learning	50

9) ----- Table Student_Guardian -----

	S_G_id	G_S_id	Relation_to_student
1	ST00002	G000101	Father
2	ST00003	G000101	Father
3	ST00004	G000102	Father

10) ----- Table Student_teacher -----

	S_T_id	T_S_id
1	ST00001	TR001
2	ST00001	TR002
3	ST00002	TR002
4	ST00002	TR003
5	ST00003	TR003

11) ----- Table Student_subject -----

	St_Su_id	Su_St_id	Student_result
1	ST00003	ECO125	98.5
2	ST00003	MAT012	97.5
3	ST00003	SPR056	95.7

12) ----- Table Subject -----

	Subject_code	Subject_name	Department
1	ECO125	Economics	Social Science
2	MAT012	Mathematics	Mathematics
3	SPR056	Sport	Sport

-----/*codes to retrieve the data from the database*/-----

```
select First_name, Last_name, Grade_level from Student where gender='Female'
```

	First_name	Last_name	Grade_level
1	Helen	Girma	12
2	Umeran	Afewwork	7
3	Amerti	Tadesse	12
4	Beti	Dawit	6
5	Yohanes	Dawit	6
6	Melat	Ashenafi	11

```
select Max(salary) as maximum_salary from Teacher
```

	maximum_salary
1	10000

```
select First_name, Last_name, Grade_level from Student where age >= 15
```

	First_name	Last_name	Grade_level
1	Nathan	Kiros	11
2	Helen	Girma	12
3	Amerti	Tadesse	12
4	Abdulhamid	Ibrahim	12
5	Melat	Ashenafi	11

```
select * from Student order by First_name,Last_name asc
```

	Student_id	First_name	Last_name	Grade_level	DOB	DOE	Age	Gender	St_Branch	St_classroom
1	ST00010	Abdulhamid	Ibrahim	12	1997-08-15	2015-12-14	19	Male	Bercheko branch	12B
2	ST00005	Amerti	Tadesse	12	1998-05-02	2015-12-13	18	Female	Bercheko branch	12A
3	ST00009	Ashenafi	Robel	1	2009-06-08	2015-12-14	7	Male	Keta branch	1B
4	ST00006	Beti	Dawit	6	2005-06-09	2015-12-13	11	Female	Eyasu tsebel	6B
5	ST00008	Dawit	Bereket	1	2012-07-06	2015-12-13	4	Male	Keta branch	1A
6	ST00003	Helen	Girma	12	1998-06-29	2015-12-10	18	Female	Bercheko branch	12A
7	ST00002	Kaleb	Girma	8	2003-08-21	2015-12-10	13	Male	Eyasu tsebel	8A
8	ST00011	Melat	Ashenafi	11	1999-06-10	2015-12-13	17	Female	Keta branch	6B
9	ST00001	Nathan	Kiros	11	1999-11-16	2015-12-12	17	Male	Keta branch	11A
10	ST00004	Umeran	Afewwork	7	2004-05-25	2015-12-11	12	Female	Eyasu tsebel	7B
11	ST00007	Yohanes	Dawit	6	2005-06-09	2015-12-13	11	Female	Eyasu tsebel	6B
12	ST00012	Zakir	Nega	9	2002-09-03	2015-12-15	14	Male	Bercheko branch	9A

-----For updating a record from the relation-----

```
update Teacher set salary = 12000 where Teacher_id = 'TR006'
```

```
update Teacher set salary = 12000 where Teacher_id = 'TR003'
```

----After the update the teacher table would be-----

	Teacher_id	First_name	Last_name	Email	Phone_no	Department	Salary
1	TR001	Wasihun	Tessema	WasT@gmail.com	+251965458798	Science	10000
2	TR002	Azeb	Desalegn	Azeb-Desalegn56@gmail.com	+251911801512	NULL	10000
3	TR003	Nuredin	Jemal	Nure-Jemal@gmail.com	+251911111619	Mathematics	12000
4	TR004	Tesfa	Kebede	TesfaKB123@gmail.com	+251912151413	Science	10000
5	TR005	Werqine	Getachew	WerqinaGech@gmail.com	+251960297845	Social science	10000
6	TR006	Yoseph	Tamru	yohanestamru012@gmail.com	+251915963249	Mathematics	12000
7	TR007	Abdissa	Yohanes	AbdiJohn@gmail.com	+251911871119	Civics	10000
8	TR008	Afewerk	Teshome	Afewek_Teshome@gmail.com	+251911975689	Science	10000
9	TR009	Sendek	Abraham	Sendek-AB@gmail.com	+251911425369	Sport	10000

-----For Deleting a record from the relation-----

```
delete from Student where Student_id in ('ST00007','ST00011','ST00006')
```

```
delete from Student where Student_id in ('ST00001','ST00009','ST00008')
```

-----After deletion of some elements the student table would be-----

	Student_id	First_name	Last_name	Grade_level	DOB	DOE	Age	Gender	St_Branch	St_classroom
1	ST00002	Kaleb	Girma	8	2003-08-21	2015-12-10	13	Male	Eyasu tsebel	8A
2	ST00003	Helen	Girma	12	1998-06-29	2015-12-10	18	Female	Bercheko branch	12A
3	ST00004	Umeran	Afewerk	7	2004-05-25	2015-12-11	12	Female	Eyasu tsebel	7B
4	ST00005	Amerti	Tadesse	12	1998-05-02	2015-12-13	18	Female	Bercheko branch	12A
5	ST00010	Abdulhamid	Ibrahim	12	1997-08-15	2015-12-14	19	Male	Bercheko branch	12B
6	ST00012	Zakir	Nega	9	2002-09-03	2015-12-15	14	Male	Bercheko branch	9A

5. REFERENCES AND WEBSITES USED

1. <https://www.lucidchart.com>
2. <https://app.diagrams.net>
3. <https://www.linkedin.com/in/asco-progress-academy>
4. <https://t.me/askoprogress>
5. https://www.w3schools.com/sql/sql_join.asp