School Management System

- Yazeed Marwan Mwafi (202020186)

1. Scenario:

A school contains many students with a unique ID, name, location, grade, and birth date. each student has a parent that has an ID, name, and phone number.

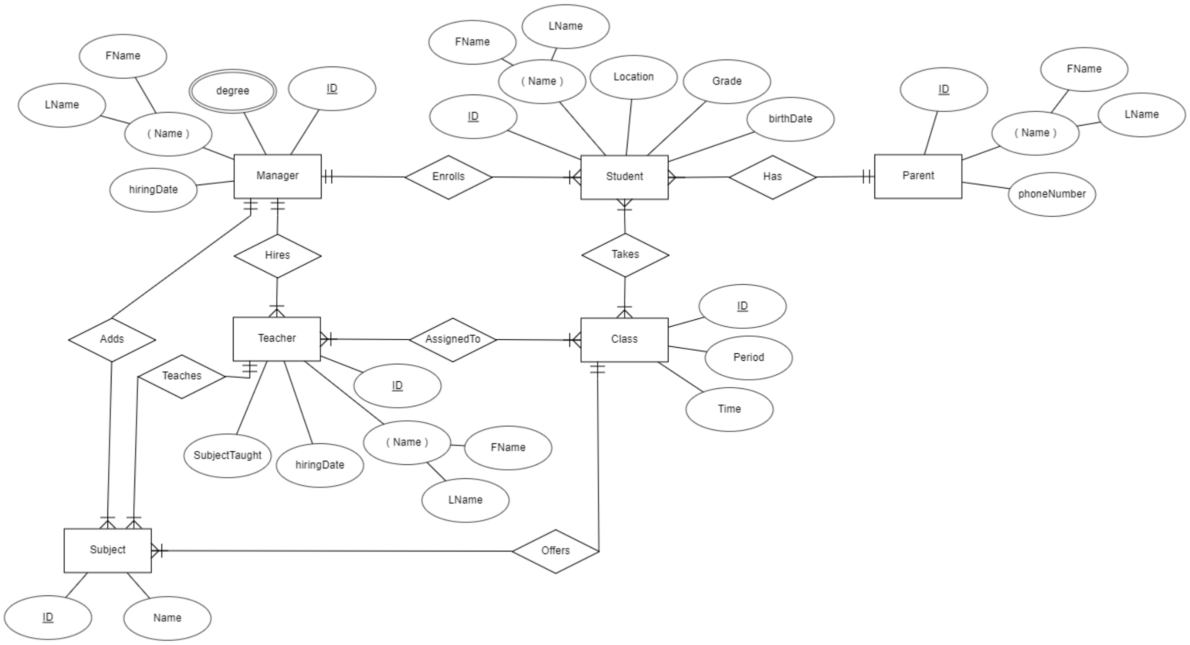
The school also contains a manager that has an ID, name, multiple degrees, and hiring date. the manager has the authority to hire teachers, enroll the students, and add subjects.

The manager hires teachers, each teacher has an ID, name, hiring date, and the subjects that he taught and teachers teach students subjects; each subject has an ID and name.

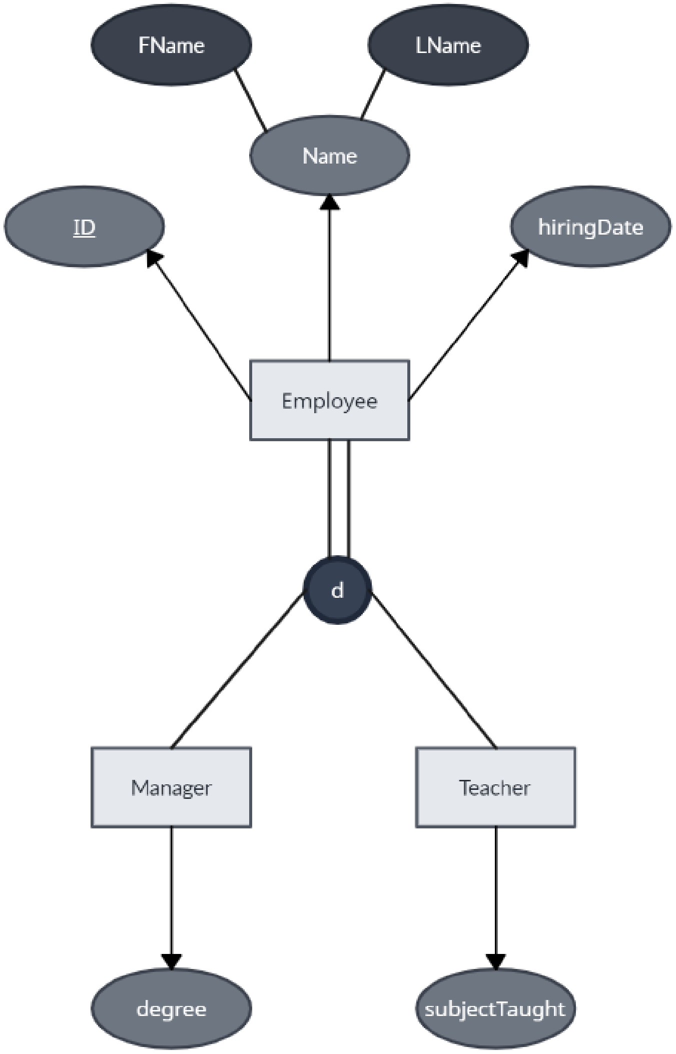
Each student takes classes and many students take each class, each class has an ID, Period, and time, and teachers are assigned to many classes.

The class can offer many subjects but the subject can be offered by one class.

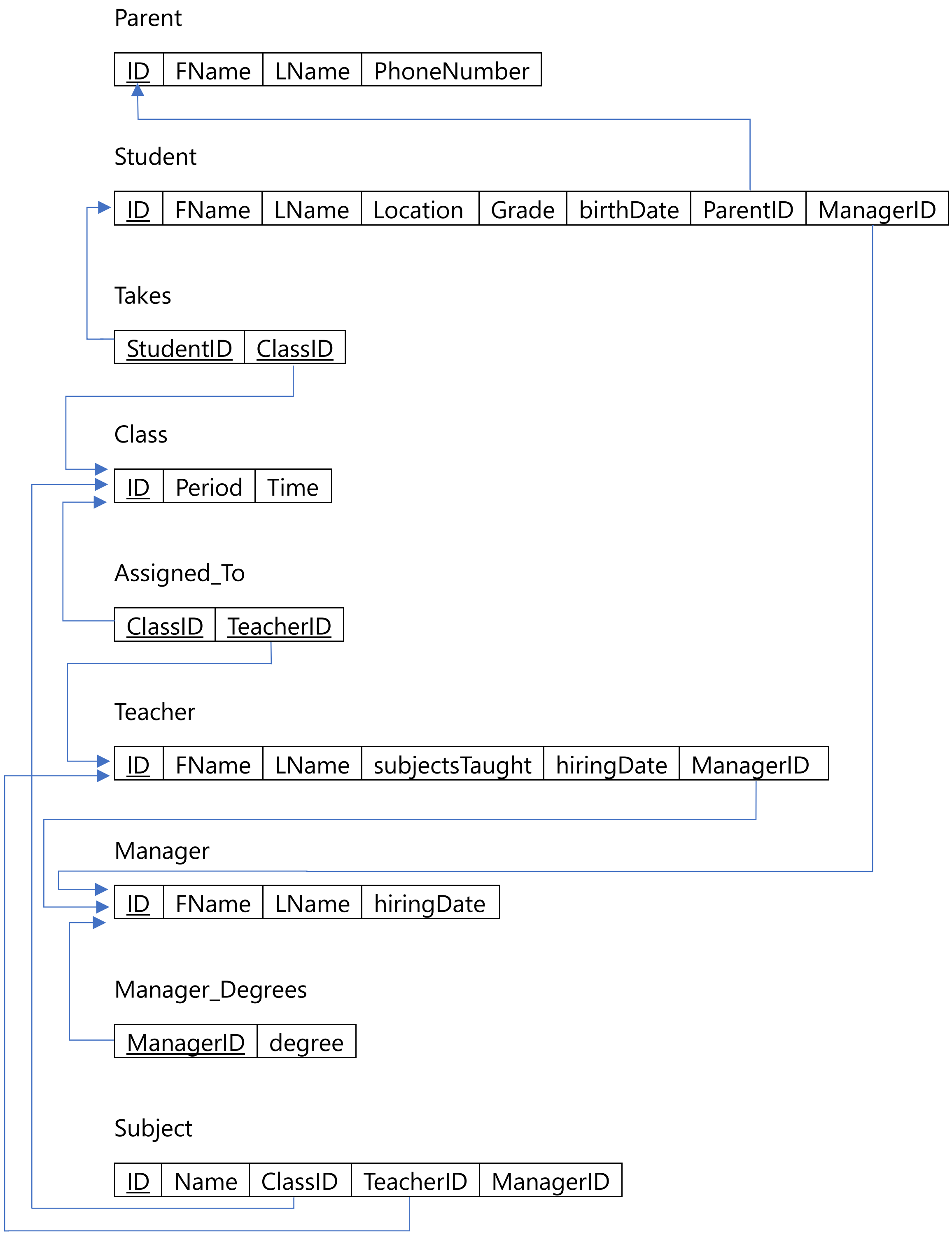
2. ER Diagram:



3. EER Modeling:



4. Mapping:



Parent

ID	FName	LName	PhoneNumber

Student

ID	FName	LName	Location	Grade	birthDate	ParentID	ManagerID

Takes

StudentID	ClassID

Class

ID	Period	Time

Assigned_To

ClassID	TeacherID

Teacher

ID	FName	LName	subjectsTaught	hiringDate	ManagerID

Manager

ID	FName	LName	hiringDate

Manager_Degrees

ManagerID	degree

Subject

ID	Name	ClassID	TeacherID	ManagerID

Phase 2

1. Relational Database Constraints:

- Primary Key Constraints:

Parent: Primary key constraint on the ID.

Student: Primary key constraint on the ID.

Class: Primary key constraint on the ID.

Teacher: Primary key constraint on the ID.

Manager: Primary key constraint on the ID.

Subject: Primary key constraint on the ID.

- Unique Constraints:

Parent: Unique constraint on the phoneNumber.

Class: Unique constraint on the combination of Period and time.

- Foreign Key Constraints:

Student: Foreign key constraints on ParentID and ManagerID referencing Parent(ID) and Manager(ID).

Takes: Foreign key constraints on StudentID and ClassID referencing Student(ID) and Class(ID).

Assigned_To: Foreign key constraints on ClassID and TeacherID referencing Class(ID) and Teacher(ID).

Teacher: Foreign key constraint on ManagerID referencing Manager(ID).

Manager_Degrees: Foreign key constraint on ManagerID referencing Manager(ID).

Subject: Foreign key constraints on ClassID, TeacherID, and ManagerID referencing Class(ID), Teacher(ID), and Manager(ID).

2. Relational Algebra:

//Retrieve all students with a grade of ‘5th grade’.

$\sigma_{\text{grade = '}5^{\text{th}}\text{ grade'}}$ Students

//Retrieve the first name and last name of all teachers.

$\Pi_{\text{FName, LName}}$ Teacher

//Rename student attributes

$\rho_{\text{(stid, stname, staddress, grade)}}$ Student

//Student ID, FName who takes class 201.

$\pi_{\text{\{ID, FName\}}} (\text{Student} \bowtie_{\text{\{ID=StudentID and ClassID='201'\}}} \text{Takes})$

//Retrieve subjects and the teachers who teach them

$\text{Subject} \bowtie_{\text{\{TeacherID=ID\}}} \text{Teacher}$

// Retrieve the names of students and their grades who are enrolled in class 'Math101' taught by 'Professor Smith'

$\pi_{\text{\{Student.FName, Student.LName, Student.grade\}}} ((\text{Student} \bowtie_{\text{\{ID=StudentID\}}} \text{Takes}) \bowtie_{\text{\{ClassID=Class.ID and instructor='Professor Smith'\}}} \text{Class})$

//Retrieve the parents of students who have a grade of '9th grade'.

$$\pi_{\{\text{Parent.FName, Parent.LName, Parent.phoneNumber}\}}((\text{Student} \bowtie_{\{\text{ID=StudentID and grade='9th Grade'}\}} \text{Takes}) \bowtie_{\{\text{ParentID=ID}\}} \text{Parent})$$

//Find the total number of students in each location.

$$\text{location}\ \mathcal{F}_{\text{count(id)}}\ \text{Student}$$

//Count the number of classes taught by each teacher.

$$\text{teacher.FName, teacher.LName}\ \mathcal{F}_{\text{count(class.id)}}((\text{Teacher} \bowtie_{\{\text{ID=TeacherID}\}} \text{Assigned\_To}) \bowtie_{\{\text{ClassID=Class.ID}\}} \text{Class})$$

3. Functional Dependencies:

Student Table:

ID → FName, LName, grade, birthDate, ParentID, ManagerID

ParentID → phoneNumber

ManagerID → FName, LName, hiringDate

Class Table:

ID → Period, time

Teacher Table:

ID → FName, LName, subjectsTaught, hiringDate, ManagerID

Manager Table:

ID → FName, LName, hiringDate

Subject Table:

ID → name

ClassID → name

TeacherID → name

ManagerID → name

Takes Table:

StudentID, ClassID → (other attributes)

Assigned_To Table:

ClassID, TeacherID → (other attributes)

Manager_Degrees Table:

ManagerID → degree

4. Normalization:

Original Tables:

Parent Table (1NF already):

ID (PK)

FName
LName
phoneNumber (Unique)

Student Table (1NF already):

ID (PK)
FName
LName
location
grade
birthDate
ParentID (FK)
ManagerID (FK)

Class Table (1NF already):

ID (PK)
Period
time

Teacher Table (1NF already):

ID (PK)
FName
LName
subjectsTaught
hiringDate
ManagerID (FK)

Manager Table (1NF already):

ID (PK)
FName
LName
hiringDate

Subject Table (1NF already):

ID (PK)
name
ClassID (FK)
TeacherID (FK)
ManagerID (FK)

Takes Table (1NF already):

StudentID (FK)
ClassID (FK)

Assigned_To Table (1NF already):

ClassID (FK)
TeacherID (FK)

Manager_Degrees Table (1NF already):

ManagerID (FK)
degree

Applying 2NF:

Student Table (2NF):

Create a new table for Student information:

Student_Info Table:

ID (PK)
FName
LName
location
birthDate
ParentID (FK)
ManagerID (FK)

Takes Table (2NF):

StudentID (FK)
ClassID (FK)
(other attributes)