A Database for an

Electronic Medical Record System

Student Clinic Module

Group Members

Abhishek Deshpande

Harika Katragadda

Suyash Nande

**Requirement Specification**

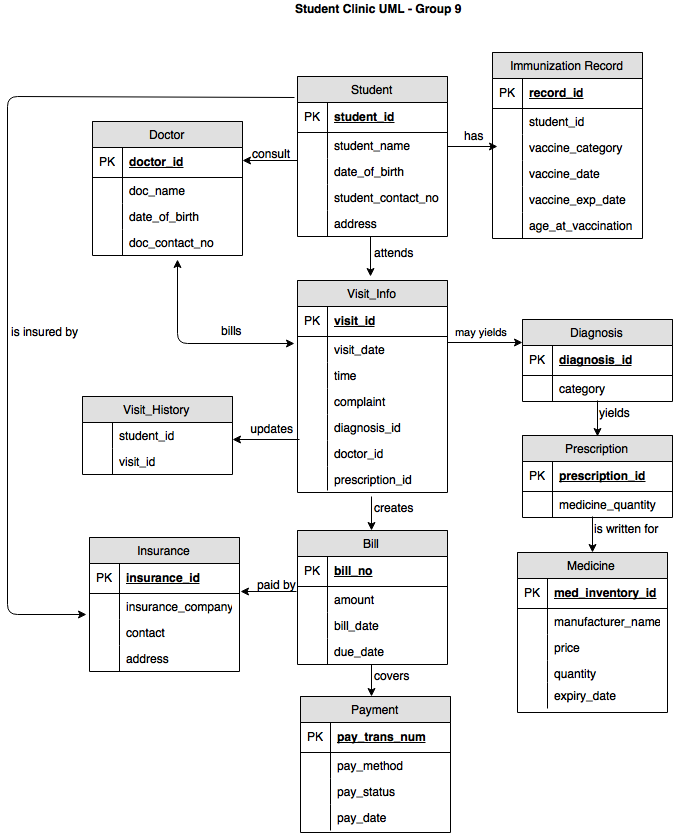
Student Clinic is the database project designed to facilitate student health record in a university. This system will be used by the doctor and clinic staff in the student health center of the university. It is developed to improve the clinic management and automates the workflow that happens in a student clinic. It will make it easier for the retrieval of history information of the student.

When the student will visit on clinic, they will ask to provide their name, date of birth, contact number, address and type of problems. Student will make registration first. If the student never registered before, student information is collected and stored in the database. Now, student is assign to the doctor, which may yield to diagnosis of the disease based on the problem category.

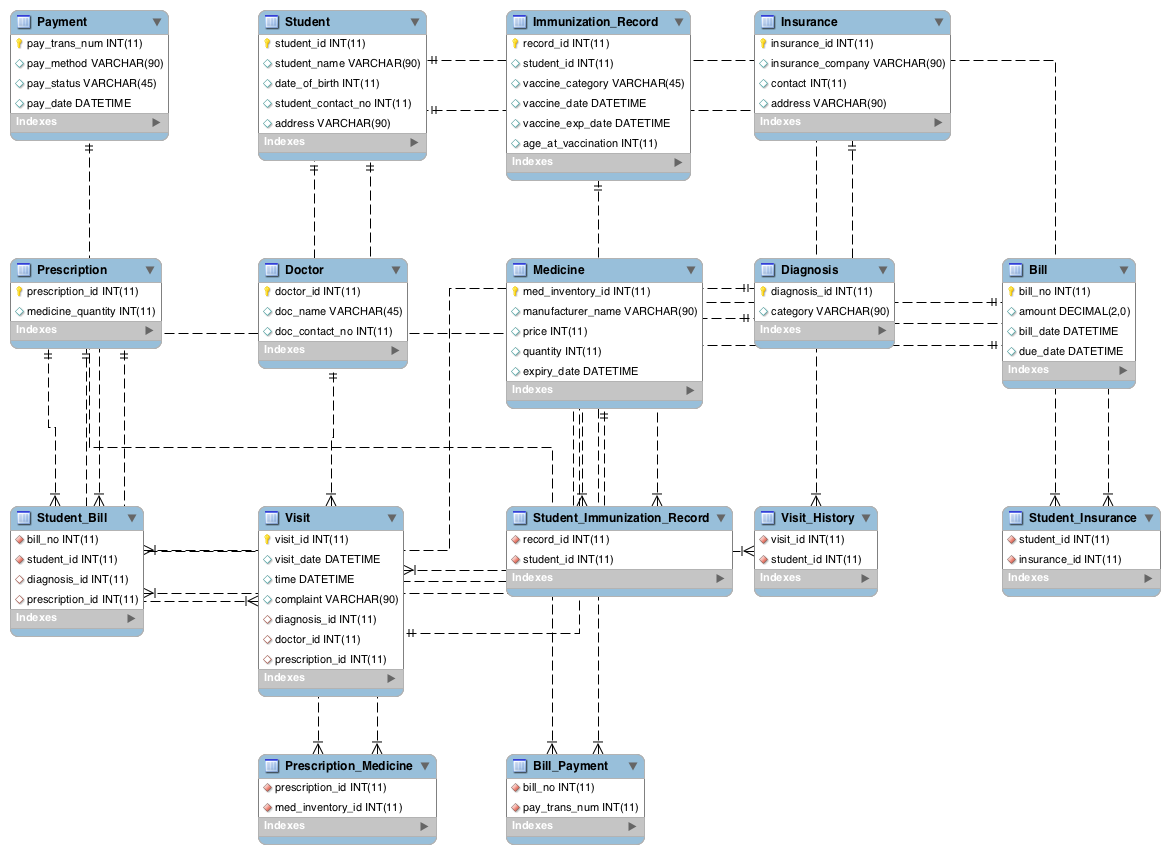
This diagnosis may further yield to treatment and doctor will then prescribe medicines which contains the medicine name, duration and quantity. Once the student gets the treatment, the doctor will send the report to the clinical staff.

The staff will view the report and complete the student record. After that, the staff will prepare the bills for the student. Student will also provide with an option for making payments. They can make their payments either by self or by medical insurance. Student will also have their immunization record maintained by clinical staff. It is easy for the management to maintain record about the student, the time for retrieving the information needed will be less compare to the manual. Then the staff will update the visit information and the student record will be kept in database.

**UML Data Model**



**ER Diagram**



**Database creation scripts**

**SQL Scripts**

1. **Select all the students having insurance**

select s.student\_name, i.insurance\_company

from insurance i

join student\_insurance using(insurance\_id)

join student s using (student\_id);

1. **Get all the visit granted by Dr. Rahul**

select visit.visit\_id, visit.visit\_date, doctor.doctor\_id

from visit

join doctor using(doctor\_id)

join diagnosis using(diagnosis\_id)

join prescription using(prescription\_id)

where doctor.doc\_name="Dr. Rahul";

1. **Get the bill details and payment number of student with student\_id=1**

select student.student\_name, bill.bill\_no, bill.bill\_date, payment.pay\_trans\_num, payment.pay\_date

from bill

join bill\_payment using (bill\_no)

join payment using (pay\_trans\_num)

join student\_bill using (bill\_no)

join student using (student\_id)

where student.student\_id=1;

1. **Get the names and contact of all doctors who have visited at least 1 student**

select doctor.doc\_name, doctor.doc\_contact\_no

from doctor

where doctor\_id in (select doctor\_id

from visit

join visit\_history using (visit\_id)

group by visit.doctor\_id

having max(visit\_history.student\_id)>=1);

1. **Find the names of the students and their address who have visited to more than one doctor.**

select s.student\_name, s.address, count(v.doctor\_id)

from student s, visit v, visit\_history vh, doctor d

where v.visit\_id = vh.visit\_id

and vh.student\_id = s.student\_id

and v.doctor\_id = d.doctor\_id

group by s.student\_name,s.address

having count(v.doctor\_id) > 1;

1. **Find the name of the doctors with their contact numbers whom has been visited by only student.**

select d.doc\_name, d.doc\_contact\_no, count(v.doctor\_id)

from doctor d, visit v

where d.doctor\_id = v.doctor\_id

group by doc\_name,doc\_contact\_no

having count(v.doctor\_id)=1;

**Experience**

* Merging the relations was made easy since foreign keys existed that referenced the various tables that were merged.
* The SQL statement became complex and more error-prone as the number of relations to be merged increased.