# AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### PROJECT

**COURSE NAME: Distributed Database Systems Lab**

**COURSE NO: CSE 4126**

**Project Name: Hospital Management System**

**Submitted by:**

Tanvir Iqbal (12-02-04-020)

Al Masum Fahim (12-02-04-027)

Shaimon Rahman(12-02-04-055)

Group: A2

Hospital Management System

Description:

In this project Patient can take serial under a particular doctor. Also we can see the whole information about the patient. If the patient needs to admit into the hospital then we can assign a room for the patient. Also patient can pay the hospital bills in this system.

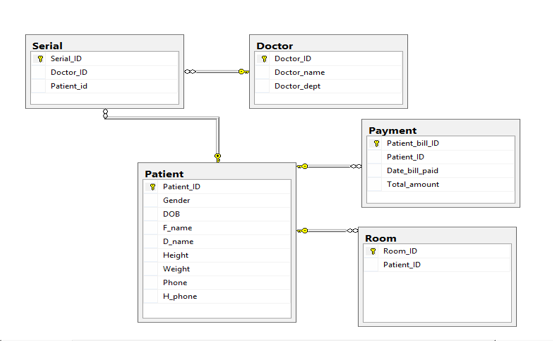
Development Tool:

* C#
* Oracle 10g
* Navicat Premium

Features:

* Patient Records
* Doctor Information
* Patient Bill Records
* Serial Record

**ERD diagram :**

****

**Table creation :**

**PATIENT TABLE**

CREATE TABLE PATIENT

(

Patient\_Id int not NULL CHECK (regexp\_like(Patient\_Id,'[0-3]{7}')),

Gender VARCHAR(6) CHECK (Gender in ('male','female')),

Dob VARCHAR(20),

P\_name VARCHAR(15) CHECK (regexp\_like(P\_name,'[a-zA-Z]')),

Weight NUMBER(5,2),

Phone NUMBER(11) CHECK (regexp\_like(Phone,'[0-9]')),

primary key (Patient\_Id)

);

**DOCTOR TABLE**

CREATE TABLE Doctor

(

Doctor\_Id int not NULL CHECK (regexp\_like(Doctor\_Id,'[4-5]{4}')),

Doctor\_name varchar(30) CHECK (regexp\_like(Doctor\_name,'[a-zA-Z]')),

Doctor\_dept VARCHAR(30) CHECK (regexp\_like(Doctor\_dept,'[a-zA-Z]')),

primary key (Doctor\_Id)

);

**SERIAL TABLE**

CREATE TABLE Serial

(

Serial\_Id int not NULL CHECK (regexp\_like(Serial\_Id,'[a-z][0-9][0-9][0-9]')),

Doctor\_Id int ,

Patient\_Id int,

foreign key (Doctor\_Id)

references Doctor (Doctor\_ID)

on delete cascade ,

foreign key (Patient\_Id)

references PATIENT (Patient\_Id)

on delete cascade

);

**ROOM TABLE**

CREATE TABLE Room

(

Room\_Id int not null CHECK (regexp\_like(Room\_Id,'[0-9][0-9][0-9][0-9]')),

Room\_Type varchar(2),

Patient\_Id int,

foreign key (Patient\_Id)

references PATIENT (Patient\_Id)

on delete cascade

);

**PAYMENT TABLE**

CREATE TABLE Payment(

Patient\_Bill\_Id int not null CHECK (regexp\_like(Patient\_Bill\_Id,'[0-9][0-9][0-9][0-9]')),

Patient\_Id int,

Date\_Bill\_Paid VARCHAR(15),

Total\_amount number(9,2),

Primary key (Patient\_Bill\_Id),

foreign key (Patient\_Id)

references PATIENT (Patient\_Id)

on delete cascade

);

Triggers:

* Trigger1 – This trigger assigns “C (Cabin)” if room id is above 5000 otherwise it assigns “W (Ward)”.

create or replace trigger room\_generate

before insert on room

for each row

declare

cnt INTEGER;

typ varchar(2);

begin

SELECT ROOM\_ID into cnt from ROOM ;

if cnt>5000 then

SELECT ROOM\_TYPE into typ FROM ROOM;

typ := 'c';

:new.room\_type := typ ;

else

typ := 'w';

:new.room\_type := typ ;

end if;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

typ := 'w';

end;

* Trigger2 – If patient’s bill is above 5000 then patient gets 5% discount.

create or replace trigger balance\_generate

before insert on PAYMENT

for each row

declare

cnt number;

begin

SELECT Total\_Amount into cnt from PAYMENT;

cnt := :new.Total\_Amount - :new.Total\_Amount\*0.05;

if :new.Total\_Amount > 5000 then

:new.Total\_Amount := cnt;

else

:new.Total\_Amount := cnt;

end if ;

end;

Functions:

* Function1 – This function returns patient name for given patient id.

CREATE FUNCTION GET\_patient\_name(paitent\_id\_param IN PATIENT.PATIENT\_ID%TYPE)

RETURN varchar

IS get\_name PATIENT.P\_NAME%TYPE;

BEGIN

SELECT P\_NAME

INTO get\_name

FROM patient

WHERE Patient\_Id = paitent\_id\_param ;

RETURN get\_name;

END;