Islamic University of Technology

CSE-4308
Database Management Systems Lab
Lab Report - 7

Name: Mukit Mahdin
ID: 200042170
Department: CSE
Program: SWE
Group: B

Task - 1: Drawing the Entity Relationship Diagram

Analysis :

Analyzing the scenario, I found 7 entities to make the entire relationship and cover the given description.

1. Citizen
2. License
3. Division
4. District
5. Hospital
6. Accident
7. Patient

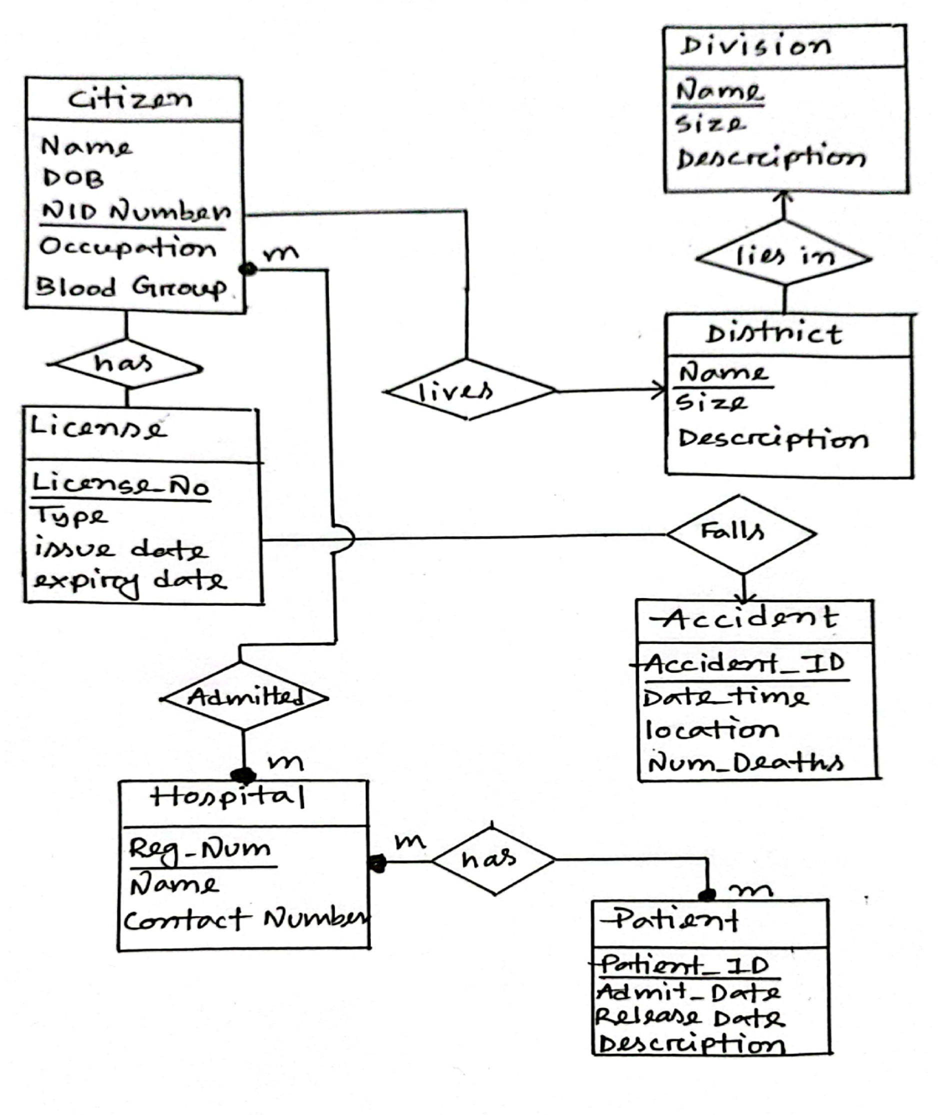

Figure: The Entity Relationship Diagram

Description :

In the scenario, Some entities are directly connected, some are not, and some are associated with junction tables. The diagram above is designed to avoid data redundancy. The description of the relationships is given below sequentially :

1. The Citizen entity has a NID Number as its primary key and a citizen lives in a District and every district is undoubtedly located in a Division.
   And a citizen can have a license. Also, Citizens and Hospitals are connected through a junction table "Admitted".
   Here I'm referring to the Citizens without having Accidents. The Relation between Citizen and District is Many to One and the Relation between Citizen and License and Citizen and Hospital is Many to Many.

2. Every District lies in a Division. So, the relationship between District and Division is Many to One.

3. In every accident, there is an associated License No and a license can fall into multiple accidents. So the Relation between Accident and License is Many to One.

4. Hospital has multiple numbers of patients admitted. And a patient can be registered in multiple hospitals. So the relationship between Hospital and Patient is many to Many. Also, Citizens can be admitted to Hospitals. This Relationship is also many to many.

Findings:

After completing the task, I realized this particular scenario can be covered by multiple kinds of solutions. The junctions table can be removed if I followed other solutions.

Problems:

I faced some confusion while connecting Hospitals, Citizens, and Patients. That's why I took junction tables to make the solution more reliable. I was not confident enough about this step.

Task - 2: The DDL Statements

Description:

To convert the above scenario, the below DDL statements were used. Suitable Primary Keys and Foreign Keys were used based on the context to make the design more solid.

```
CREATE TABLE CITIZEN(
    NID_NUMBER INT NOT NULL,
    NAME VARCHAR2(20) NOT NULL,
    DATE_OF_BIRTH DATE NOT NULL,
    OCCUPATION VARCHAR2(20),
    BLOOD_GROUP VARCHAR2(5),

    CONSTRAINT PK_NID PRIMARY KEY(NID_NUMBER),
    CONSTRAINT FK_DISTRICT FOREIGN KEY(DISTRICT_NAME) REFERENCES
DISTRICT(DISTRICT_NAME)
);

CREATE TABLE DISTRICT(
    DISTRICT_NAME VARCHAR2(20) NOT NULL,
    SIZE INT NOT NULL,
    DESCRIPTION VARCHAR2(150) NOT NULL,

    CONSTRAINT PK_DISTRICT PRIMARY KEY(DISTRICT_NAME),
    CONSTRAINT FK_DIVISION FOREIGN KEY(DIVISION_NAME) REFERENCES
DIVISION(DIVISION_NAME)
);

CREATE TABLE DIVISION(
    DIVISION_NAME VARCHAR2(20) NOT NULL,
    SIZE INT NOT NULL,
    DESCRIPTION VARCHAR2(150) NOT NULL,

    CONSTRAINT PK_DIVISION PRIMARY KEY(DIVISION_NAME)
);

CREATE TABLE LICENSE(
    LICENSE_NO VARCHAR2(20) NOT NULL,
    LICENSE_TYPE VARCHAR2(20) NOT NULL,
    ISSUE DATE NOT NULL,
    EXPIRY DATE NOT NULL,

    CONSTRAINT PK_LICENSE PRIMARY KEY(LICENSE_NO),
    CONSTRAINT FK_NID FOREIGN KEY(NID_NUMBER) REFERENCES
CITIZEN(NID_NUMBER)
);

CREATE TABLE ACCIDENT(
    ACCIDENT_ID INT NOT NULL,
    ACCIDENT_DTIME DATETIME NOT NULL,
    LOCATION VARCHAR2(20) NOT NULL,
    NUM_DEATHS INT,

    CONSTRAINT PK_ACCIDENT PRIMARY KEY(ACCIDENT_ID),
    CONSTRAINT FK_LICENSE FOREIGN KEY(LICENSE_NO) REFERENCES
LICENSE(LICENSE_NO)
);

CREATE OR REPLACE TYPE VMOBILES AS VARRAY(10) OF VARCHAR2(20);

CREATE TABLE HOSPITAL(
    REG_NUM VARCHAR2(20) NOT NULL,
    NAME VARCHAR2(50) NOT NULL,
    CONTACT NUMBER VMOBILES,

    CONSTRAINT PK_HOSPITAL PRIMARY KEY(REG_NUM),
);

CREATE TABLE ADMITTEDHOSPITAL(
    CONSTRAINT FK_ADMITTED FOREIGN KEY(NID_NUMBER) REFERENCES
CITIZEN(NID_NUMBER),
    CONSTRAINT FK_HOSPITAL FOREIGN KEY(REG_NUM) REFERENCES
HOSPITAL(REG_NUM)
);
```

```
CREATE TABLE PATIENT(
    PATIENT_ID VARCHAR2(20) NOT NULL,
    ADMIT_DATE DATE NOT NULL,
    RELEASE_DATE DATE NOT NULL,
    DESCRIPTION VARCHAR2(100) NOT NULL,

    CONSTRAINT PK_PATIENT PRIMARY KEY(PATIENT_ID)
);

CREATE TABLE PATIENTHOSPITAL(
    CONSTRAINT FK_PATIENT FOREIGN KEY(PATIENT_ID) REFERENCES
PATIENT(PATIENT_ID),
    CONSTRAINT FK_HOSPITAL FOREIGN KEY(REG_NUM) REFERENCES
HOSPITAL(REG_NUM)
);
```

Here, PATIENTHOSPITAL is the junction table that connects Patient and Hospital and ADMITTEDHOSPITAL is the junction table to connect Citizen and Hospital.

Problem:

The above DDLs only cover the process that I mentioned in Task 1. Defining other scenarios or relations isn't possible by them. For that, the keys should be redefined.

Task - 3: Performing Queries.

```
-- a
select divisionname,count(districtname) as numberofdistricts
from division natural join district
group by division.divisionname;
```

```sql
-- b
select districtname
from district natural join citizen
where count(nid)>=20000;



-- c
select count(accidentid) as NumberofAccidents
from citizen natural join accident
where citizen.nid=210;



-- d
select rownum as rank,hospitalid
from (select hospitalid, count(patientid) as numberofpatients
from hospitalpatient
group by hospitalid)
order by numberofpatients
where rownum<=5;

-- e
select patientid,bloodgroup
from patient natural join citizen;



-- f
select divisionname,tot/(divisionsize)
from
(select divisionname,count(citizen.nid) as tot,divisionsize
from division natural join citizen
group by divisionname);
```

```
-- g
select rownum as rank,divisionname,tot/(divisionsize) as density
from
(select divisionname,count(citizen.nid) as tot,divisionsize
from division natural join citizen
group by divisionname)
order by tot.(divisionsize) asc
where rownum<=3;


-- h
select district.districtname,count(accidentid) as numofaccidents
from district natural join accident
group by districtname;


-- i
select    rownum    as    rank,divisionname,count(accidentid)    as
numofaccidents
from division natural join accident
group by divisionname
order by count(accidentid) asc
where rownum<=1;


-- j
select          drivinglicense.type,count(accidentid)          as
numberofaccidentsoccured
from drivinglicense natural join accident
group by drivinglicense.type
where drivinglicense.type='professional' or
drivinglicense.type='non=professional' ;
```

```
-- k
select rownum as rank,name
from
(select
citizen.citizenid,(patient.releasedate-patient.admissionedate)
as period
from citizen natural join patient
order by (patient.releasedate-patient.admissionedate) desc)
where rownum<=1;


-- l
select rownum as rank,name
from
(select divisionname,count(nid) as Tot
from division natural join citizen
where (CAST(GetDate() AS Date)) - DOB>= 15 and (CAST(GetDate()
AS Date)) - DOB<=
30
order by count(nid) desc)
where rownum<=1;


-- m
select citizen.name;
from citizen natural join drivinglicense
(CAST(GetDate() AS Date)) > expirationdate;




-- n
select licensenumber,count(accidentid) as numofaccidentsoccured
```

```sql
from
(select drivinglicense.licensenumber;
from citizen natural join drivinglicense
(CAST(GetDate()   AS   Date))   >   expirationdate)   natural   join
accident;


-- o
select licensenumber,count(accidentid) as accidents
from drivinglicense natural join accident
group by licensenumber
where count(accidentid)=0;


-- p
select divisionname,count(accidentid) as accidentnumber
from division natural join accident
group by divisionname;


-- q
select citizen.name
from citizen natural join drivinglicense
where   ((CAST(GetDate()   AS   Date))   -   DOB)   <=22   OR   (
(CAST(GetDate() AS Date)) -
DOB) >=40;

-- r
select citizen.name
from accident natural join patient
where accident.date=patient.admissionedate;


-- s
```

```sql
select hospital.id
from
(select hospital.id,count(citizen.nid) as tot
from citizen natural join division natural join hospital
where division.name='dhaka'
group by hospital.id
order by count(citizen.nid) desc
)
where rownum<=1;


-- t
select citizen.name
from citizen natural join accident natural join drivinglicense
where accident.location
not in
(select citizen.name,division.divisionname
from citizen natural join division)
```