Lab 7 Report

Name : Mashrur Ahsan

ID : 200042115

Program : SWE

Department : CSE

Course : CSE 4308

Information Given:

Nationa ID (NID) is an integrated collection of citizens' information such as Name, Date of Birth, Occupation, Blood Group. Each citizen has his/her own NID. In order to investigate the population density, the country has been divided into divisions. Each divison has its name, size (in square KM), and a brief description. Again, each division has a number of districts with similar attributes. Citizen information must be connected to its corresponding division and district.

Each citizen may have exactly one driving license where information such as type of license, issue date, expiration date are maintained. Whenever any accident occurs, it is logged in the central system. The system stores relevant information such as date and time of accident, location of accident, number of deaths (if any), etc.

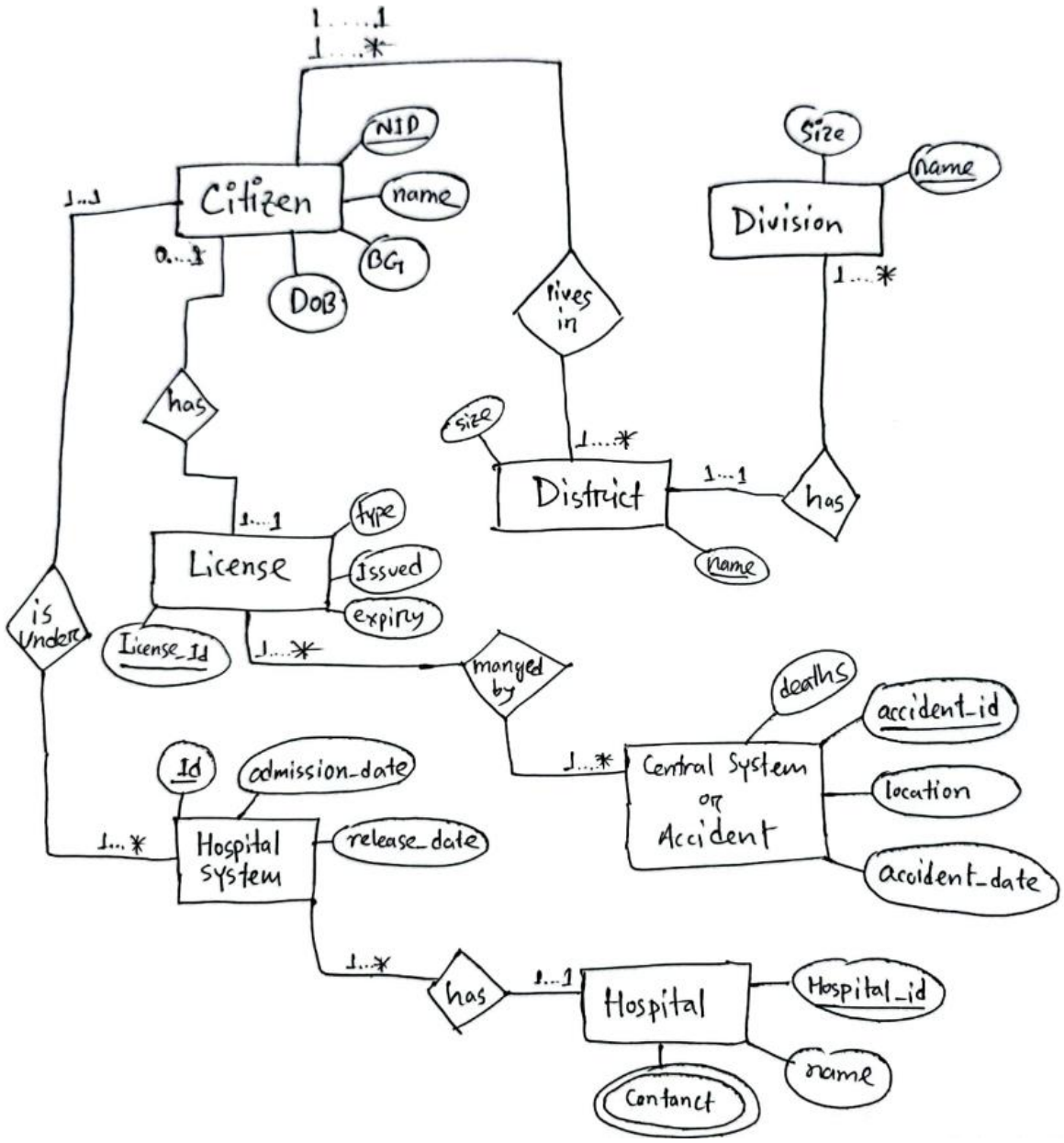
There are a number of hospitals in the country having name and contact information. Each hospital may have more than one contact number. Citizens may avail treatment in any hospitals they prefer. Whenever any patient (i.e., citizen) is admitted, the system keeps the record of his/here date of admission, a brief description, and release date.

Now, your task is to:

Problem statement:

1. Draw an ER Diagram, without any data redundancy, specifying the cardinality explicitly. You may add additional attributes only if it is needed.

Here's the ER Diagram: (It's a scanned picture since I drew it on my copy)



2. Convert the ER Diagram into DDL using standard SQL denoting the appropriate constraints.

The DDL statements are given below:

```
DROP TABLE HospitalSystem;
DROP TABLE Hospital;
DROP TABLE Citizen;
DROP TABLE DISTRICT;
DROP TABLE DIVISION;
DROP TABLE ACCIDENT;
DROP TABLE LICENSE;
```

```
create table DIVISION
(
    name VARCHAR2(55),
    capacity numeric(10,2),
    constraint PK_DIVISION_NAME primary key(name)
);


create table DISTRICT
(
    name VARCHAR2(55),
    capacity numeric(10,2),
    division_name varchar2(55),
    constraint PK_DISTRICT_NAME primary key(name),
    constraint FK_DIVISION_NAME FOREIGN key(division_name) references DIVISION(name)
);
create table HOSPITAL
(
    Hospital_id NUMBER,
    name VARCHAR2(55),
    contact_number NUMBER,
    CONSTRAINT PK_hospital PRIMARY key(Hospital_id)
);


CREATE TABLE License
(
    License_ID NUMBER,
    type VARCHAR(55),
    Issued DATE,
    expiry DATE,
    constraint PK_LICENSE PRIMARY KEY (License_ID)
);
```

```sql
create table Citizen
(
    NID NUMBER,
    name VARCHAR2(55),
    DOB DATE,
    BloodGroup VARCHAR2(3),
    division_name VARCHAR2(55),
    district_name varchar2(55),
    License_ID NUMBER,
    CONSTRAINT PK_CITIZEN_NID PRIMARY KEY (NID),
    CONSTRAINT FK_DISTRICT_CITIZEN FOREIGN KEY (district_name) REFERENCES DISTRICT(name),
    CONSTRAINT FK_DIVISION_CITIZEN FOREIGN KEY (division_name) REFERENCES DIVISION(name),
    CONSTRAINT FK_LICENSE_CITIZEN FOREIGN KEY (license_ID) REFERENCES LICENSE(License_ID)
);


CREATE TABLE HospitalSystem
(
    id NUMBER,
    admission_date DATE,
    release_date DATE,
    NID NUMBER,
    hospital_id NUMBER,
    CONSTRAINT PK_hospital_SYSTEM primary key(id),
    CONSTRAINT FK_CITIZEN_NID_IN_HOSPITAL foreign key(NID) references Citizen(NID),
    CONSTRAINT FK_HOSPITAL FOREIGN KEY (hospital_id) references Hospital(hospital_id)
);

CREATE TABLE ACCIDENT
(
    ACCIDENT_ID NUMBER,
    ACCIDENT_DATE DATE,
    DEATHS NUMBER,
    location varchar2(55),
    License_ID NUMBER,
    constraint PK_ACCIDENT PRIMARY KEY (ACCIDENT_ID),
    constraint FK_LICENSE_ACCIDNT FOREIGN KEY (License_ID) REFERENCES LICENSE(LICENSE_ID) ON DELETE CASCADE
);
```

The last task was to write SQL statements for the following queries:

(a) Find the list of divisions along with its total number of districts.

(b) Find the list of districts having at least 20,000 people living there.

(c) Find the number of accidents that involved a citizen whose NID is 210.

(d) Find the list of top 5 hospitals based on the number of patients admitted so far.

(e) Find the blood group of all the patients admitted to different hospitals.

(f) Find the population density for each division.

(g) Find the top 3 densely populated districts.

(h) Find the number of accidents that occurred in each district.

(i) Find the division where the least amount of accidents occurred.

(j) Find the number of accidents caused by 'non-professional' and 'professional' license holders.

(k) Find the person who was admitted to the hospital for the longest period of time.

(l) Find the division where the number of young people ($15 \leq \text{age} \leq 30$) is the lowest.

(m) Find the people whose licenses expired.

(n) Find the number of accidents caused by people whose licenses expired.

(o) Find the license holders who were not involved in any accident so far.

(p) Find the number of deaths due to any accident for each division.

(q) Find the name of the people who got their license before the age of 22 or after the age of 40.

(r) Find the list of citizens who were admitted to the hospital on the same day they got into an accident.

(s) Find the hospital where people from Dhaka division were admitted the most.

(t) Find the list of people who caused an accident outside their own district.

The SQL statements are given below:

```sql
-- a --
SELECT DIVISION_NAME, COUNT(*) Districts FROM DISTRICT
GROUP BY division_name;


-- b --
SELECT CITIZEN.district_name, COUNT(Citizen.NID) AS People
FROM CITIZEN , DISTRICT
WHERE CITIZEN.district_name = DISTRICT.NAME
HAVING COUNT(CITIZEN.NID) >= 20000
GROUP BY CITIZEN.district_name;


-- c --
SELECT COUNT(ACCIDENT_ID) Accidents
FROM ACCIDENT, License, CITIZEN
where Citizen.license_ID = License.license_ID
    AND ACCIDENT.license_ID = License.license_ID
        AND Citizen.NID = 210;


-- d --
SELECT ROWNUM AS RANK, name
FROM(SELECT Hospital.name, COUNT(HospitalSystem.id)
    FROM HOSPITAL, HospitalSystem
    WHERE hospital.hospital_id = HospitalSystem.hospital_id
    GROUP BY Hospital.name)
WHERE ROWNUM <= 5;
```

```sql
-- e --
SELECT CITIZEN.NAME, BloodGroup FROM CITIZEN, HospitalSystem
WHERE CITIZEN.NID = HospitalSystem.NID;


-- f --
SELECT division.name, Total/(capacity)
FROM(SELECT DIVISION.NAME, COUNT(CITIZEN.NID) AS TOTAL
    FROM division, CITIZEN
    where DIVISION.name = division_name
    GROUP BY DIVISION.name);


-- g --
SELECT ROWNUM AS RANK, DIVISIN.name
FROM(SELECT Division.NAME, TOTAL/(capacity)
    FROM(SELECT Division.NAME, COUNT(NID) AS TOTAL
    FROM DIVISION , CITIZEN
    WHERE DIVISION.name = division_name
    GROUP BY Division.name))
WHERE ROWNUM <= 3;
```

```sql
-- h --
SELECT DISTRICT.NAME, COUNT(ACCIDENT_ID) as accidents
FROM DISTRICT , CITIZEN , License, ACCIDENT
WHERE DISTRICT.name = CITIZEN.district_name
    AND CITIZEN.LICENSE_ID = License.LICENSE_ID
    AND License.LICENSE_ID = ACCIDENT.LICENSE_ID
GROUP BY DISTRICT.NAME;


-- i --
SELECT NAME FROM
( SELECT DIVISION.NAME, COUNT(ACCIDENT_ID) AS ACCIDENTS
  FROM DIVISION , CITIZEN, ACCIDENT, LICENSE
  WHERE CITIZEN.LICENSE_ID = License.LICENSE_ID
  AND License.LICENSE_ID = ACCIDENT.LICENSE_ID
  AND DIVISION.name = CITIZEN.division_name
  GROUP BY DIVISION.NAME
  ORDER BY ACCIDENTS
) WHERE ROWNUM <= 1;
```

```sql
-- j --
SELECT COUNT(ACCIDENT_ID) AS ACCIDENTS
FROM ACCIDENT, LICENSE
WHERE ACCIDENT.LICENSE_ID = LICENSE.LICENSE_ID
    AND (LICENSE.TYPE = 'Professional'
      OR LICENSE.TYPE = 'Non-professional');


-- k --
SELECT ROWNUM AS RANK, NAME
FROM(SELECT Citizen.NAME, (HS.release_date - HS.admission_date) AS TIME
    FROM CITIZEN, HospitalSystem HS
    WHERE Citizen.NID = HS.NID
    ORDER BY TIME DESC)
WHERE ROWNUM <= 1;


-- l --
SELECT ROWNUM AS RANK, NAME
FROM(SELECT DIVISION.NAME, COUNT(NID) AS TOTAL
    FROM DIVISION , CITIZEN
    WHERE DIVISION.NAME = division_name
        AND  (CAST(GetDate() AS Date)) - DOB>= 15
        AND  (CAST(GetDate() AS Date))-DOB <= 30
    ORDER BY TOTAL DESC)
WHERE ROWNUM <= 1;

-- m --
SELECT NAME
FROM CITIZEN, License L
WHERE Citizen.LICENSE_ID = L.LICENSE_ID AND (CAST(GetDate() AS Date)) > L.expiry;


-- n --
SELECT COUNT(ACCIDENT_ID) AS ACCIDENTS
FROM ACCIDENT, LICENSE
WHERE ACCIDENT.LICENSE_ID = LICENSE.LICENSE_ID
    AND (CAST(GetDate() AS Date)) > expiry;


-- o --
SELECT L.License_ID
FROM LICENSE L
WHERE L.License_ID NOT IN
(SELECT ACCIDENT.License_ID FROM ACCIDENT);
```

```sql
-- p --
SELECT D.NAME, DEATHS
FROM DIVISION D, CITIZEN C, License L, ACCIDENT A
WHERE D.NAME = C.division_name
    AND C.LICENSE_ID = L.LICENSE_ID
    AND L.LICENSE_ID = A.LICENSE_ID;


-- q --
SELECT C.NAME FROM CITIZEN C, LICENSE L
WHERE C.LICENSE_ID = L.LICENSE_ID
    AND ( ( (CAST(GetDate() AS Date)) - DOB) <=22 OR ( (CAST(GetDate() AS Date)) - DOB) >=40);


-- r --
SELECT C.NAME
FROM CITIZEN C, LICENSE L, ACCIDENT A, HospitalSystem HS
WHERE C.LICENSE_ID = L.LICENSE_ID
    AND L.LICENSE_ID = A.LICENSE_ID
    AND A.ACCIDENT_DATE = HS.admission_date;
```

```sql
-- s --
SELECT NAME
FROM(SELECT H.NAME, COUNT(C.NID) AS TOTAL
    FROM HOSPITAL H, HospitalSystem HS, CITIZEN C
    WHERE C.division_name = 'Dhaka'
       AND HS.HOSPITAL_ID = H.HOSPITAL_ID
    GROUP BY H.NAME
    ORDER BY TOTAL DESC)
WHERE ROWNUM <= 1;


-- t --
SELECT C.NAME
FROM CITIZEN C, ACCIDENT A, License L
WHERE C.LICENSE_ID = L.LICENSE_ID AND A.LICENSE_ID = L.LICENSE_ID
    AND A.LOCATION NOT IN (SELECT C.district_name FROM CITIZEN);
```

The data that was inserted while testing the statements:

```
INSERT INTO DIVISION VALUES('Dhaka', 123456.12);
INSERT INTO DIVISION VALUES('Barisal', 23455.12);

INSERT INTO DISTRICT VALUES('Gulshan', 8321.33, 'Dhaka');
INSERT INTO DISTRICT VALUES('Uttara', 5122.22, 'Dhaka');
INSERT INTO DISTRICT VALUES('Taltoli', 5122.22, 'Barisal');
INSERT INTO DISTRICT VALUES('Dhaka', 40000, 'Dhaka');

INSERT INTO License VALUES(101, 'Professional', '01-apr-2009', '01-apr-2014');
INSERT INTO License VALUES(102, 'Non-professional', '03-feb-2017', '03-feb-2022');
INSERT INTO License VALUES(103, 'Non-professional', '23-jun-2015', '23-jun-2020');

INSERT INTO ACCIDENT VALUES(1, '14-mar-2011', 0, 'Dhaka', 101);
INSERT INTO ACCIDENT VALUES(2, '15-mar-2011', 0, 'Barisal', 101);
INSERT INTO ACCIDENT VALUES(3, '25-nov-2012', 0, 'Dhaka', 103);


INSERT INTO CITIZEN VALUES(210, 'Sam', '26-jun-2002', 'A+', 'Dhaka', 'Dhaka', 101);
INSERT INTO CITIZEN VALUES(240, 'Pranto', '02-jul-2002', 'O+', 'Dhaka', 'Dhaka', 103);

INSERT INTO HOSPITAL VALUES(1, 'A MEDICAL COLLEGE HOSPITAL', 01712345987);

INSERT INTO HospitalSystem VALUES(5000, '14-mar-2011', '15-mar-2011', 210 ,1);
INSERT INTO HospitalSystem VALUES(5001, '11-jan-2011', '02-dec-2014', 240 ,1);
```