



Assignment #: Lab-07A

Mirza Mohammad Azwad

ID: 200042121

Date of Submission: 20/10/2022

Write a report for the following task. [Click Here](#)

Contents	1
1) Introduction	2
2) Detailed Lab Report	3
Task-1	3
Task-2	5
Task-3	7
3) Findings	11
4) Summary	11

Programming Language Used For Solution: **Oracle SQL**

Introduction

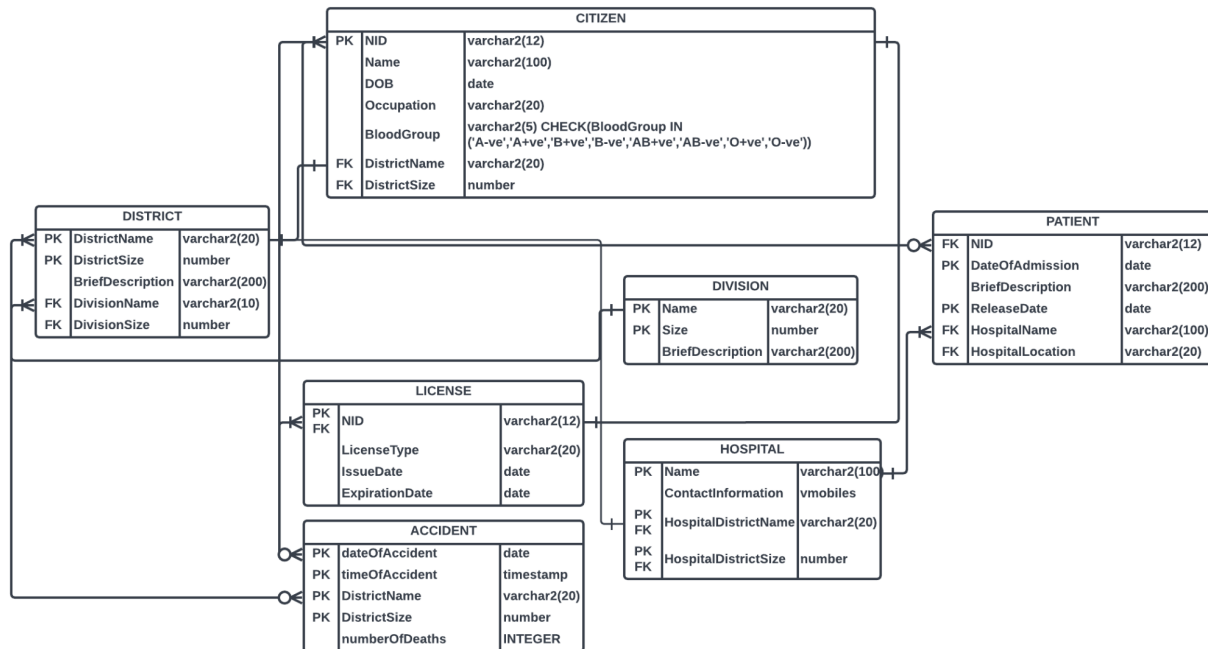
In this lab we were asked to prepare an ERD using a collection of given information. The ERD was then to be translated into DDL statements. And then we had a collection of DMLs to perform on the constructed database.

I used the lucidchart drawing tool for the ERD diagram construction. I wrote the DDL statements using oracle SQL to create the database as per the constructed ERD as well as the DML as per the given requirements.

The DDL implementation made use of upto 9 tables all of which had various interconnections between them, I did not however show the junction tables in the ER Diagram instead just showed the many to many cardinality as it is.

Detailed Lab Report

Task-1 ERD implementation



The CITIZEN table stores information as per the first paragraph and then it stores the DistrictName and DistrictSize, the reason I store both is that I thought there could be District Names that match, it may be a speculative general approach, but I felt that District Name alone cannot be a good indicator and this question does not mention that this system is applicable only for Bangladesh. Like for instance 2 different countries could have the same district names.

CITIZEN shares a one to one connection with license since one citizen can have one and only one license. The LICENSE table stores the NID as its primary key as well as foreign key to allow this connection to take place.

CITIZEN shares a connection with DISTRICT with a one to many cardinality as there can be many citizens from the same district.

The DISTRICT table shares a many to one cardinality with DIVISION as there can be many districts within one division.

LICENSE and ACCIDENT share a many to many cardinality as one individual can commit many accidents and accidents can involve more than one person which means that we must introduce a junction table(ACCIDENT_LICENSE) here as shown in the DDL.

ACCIDENT and CITIZEN also share a many to many cardinality as accidents may not just take place for license holders but it can also take place for non-license holders. A junction table(ACCIDENT_CITIZEN) is also required here to represent the many to many cardinalities.

ACCIDENTS also share a one to many cardinality with DISTRICT as one accident takes place in a specific DISTRICT but the same accident cannot take place in multiple places.

HOSPITAL table also shares a one to one relation with DISTRICT and one to many with PATIENT as HOSPITAL would have a specific location and one patient can be admitted to one hospital at once but one hospital can have many patients.

PATIENT table has a one to one relation with CITIZEN as one patient corresponds to one and only one CITIZEN and it also has a many to one relationship with HOSPITAL as one HOSPITAL can contain multiple patients.

Extra attributes added include HospitalName as an identifier for HOSPITAL

Task-2 DDL of the table as per the ERD

```
CREATE REPLACE TYPE vmobiles as varray(10) of varchar2(20);
```

Run this in the sqlplus command line with ‘\’ to prevent errors when executing the remainder of the DDL.

```
CREATE TABLE CITIZEN(
    NID varchar2(12),
    Name varchar2(100),
    DOB date,
    Occupation varchar2(20),
    BloodGroup varchar2(5),
    DistrictName varchar2(20),
    DistrictSize number,
    constraint PK_CITIZEN PRIMARY KEY(NID),
    constraint BloodGroup_CONSTRAINT CHECK( BloodGroup
in('A-ve','A+ve','B+ve','B-ve','AB+ve','AB-ve','O+ve','O-ve'))
);

CREATE TABLE DIVISION(
    DivisionName varchar2(20),
    DivisionSize number,
    BriefDescription varchar2(200),
    constraint PK_DIVISION PRIMARY KEY(DivisionName,DivisionSize)
);

CREATE TABLE DISTRICT(
    DistrictName varchar2(20),
    DistrictSize number,
    BriefDescription varchar2(200),
    DivisionName varchar2(10),
    DivisionSize number,
    CONSTRAINT PK_DISTRICT PRIMARY KEY(DistrictName, DistrictSize),
    CONSTRAINT FK_DISTRICT_Division FOREIGN KEY(DivisionName,DivisionSize)
REFERENCES DIVISION(DivisionName,DivisionSize) ON DELETE CASCADE
);
```

```

CREATE TABLE LICENSE(
    NID varchar2(12),
    LicenseType varchar2(20),
    IssueDate date,
    ExpirationDate date,
    constraint PK_LICENSE PRIMARY KEY(NID),
    constraint FK_LICENSE_CITIZEN FOREIGN KEY(NID) REFERENCES CITIZEN(NID)
);

CREATE TABLE ACCIDENT(
    dateOfAccident date,
    timeOfAccident timestamp,
    DistrictName varchar2(20),
    DistrictSize number,
    numberOfDeaths INTEGER,
    constraint PK_ACCIDENT PRIMARY KEY(dateOfAccident, timeOfAccident,
DistrictName,DistrictSize)
);

CREATE TABLE ACCIDENT_LICENSE(
    dateOfAccident date,
    timeOfAccident timestamp,
    DistrictName varchar2(20),
    DistrictSize number,
    NID varchar2(12),
    constraint PK_ACCIDENT_LICENSE PRIMARY KEY(dateOfAccident,
timeOfAccident,DistrictName,DistrictSize,NID),
    constraint FK_ACCIDENT_LICENSE_LICENSE FOREIGN KEY(NID) REFERENCES
LICENSE(NID),
    constraint FK_ACCIDENT_LICENSE_ACCIDENT FOREIGN KEY(dateOfAccident,
timeOfAccident,DistrictName,DistrictSize) REFERENCES
ACCIDENT(dateOfAccident, timeOfAccident,DistrictName,DistrictSize)
);

CREATE TABLE HOSPITAL(
    HospitalName varchar2(100),

```

```

HospitalDistrictName varchar2(20),
HospitalDistrictSize number,
ContactInformation vmobiles,
                                constraint          PK_HOSPITAL          PRIMARY
KEY (HospitalName,HospitalDistrictName,HospitalDistrictSize)
);

CREATE TABLE PATIENT(
  NID varchar2(12),
  DateOfAdmission date,
  BriefDescription varchar2(200),
  ReleaseDate date,
  HospitalName varchar2(100),
  HospitalDistrictName varchar2(20),
  HospitalDistrictSize number,
  constraint PK_PATIENT PRIMARY KEY (DateOfAdmission, ReleaseDate),
  CONSTRAINT FK_PATIENT_NID FOREIGN KEY (NID) REFERENCES CITIZEN (NID),
                                constraint          fk_patient_hospital          FOREIGN
KEY (HospitalName,HospitalDistrictName,HospitalDistrictSize) REFERENCES
HOSPITAL (HospitalName,HospitalDistrictName,HospitalDistrictSize)
);

```

Task-3 DML for the given tasks

```

--Task-a--
SELECT          *          FROM          DIVISION, (SELECT
DivisionName,count (DivisionName) DistrictCount FROM DISTRICT GROUP BY
DIVISION) DistrictDivision WHERE
DIVISION.DivisionName=DistrictDivision.DivisionName;

--Task-b--

```



```

SELECT DISTRICT.DistrictName,count(DISTRICT.DistrictName)Population FROM
DISTRICT,CITIZEN WHERE DISTRICT.DistrictName=CITIZEN.DistrictName AND
DISTRICT.DistrictSize=CITIZEN.DistrictSize GROUP BY DISTRICT.DistrictName;

--Task-c--
SELECT count(*)NumberOfAccidents FROM ACCIDENT_LICENSE,CITIZEN WHERE
ACCIDENT_LICENSE.NID='210' AND CITIZEN.NID='210' GROUP BY CITIZEN.NID;

--Task-d--
SELECT * FROM (SELECT HospitalName,count(*)NumberOfPatients FROM
HOSPITAL,PATIENT WHERE HOSPITAL.Name=PATIENT.HospitalName GROUP BY
HospitalName ORDER BY DESC) WHERE ROWNUM<=5;

--Task-e--
SELECT CITIZEN.BloodGroup FROM CITIZEN,PATIENT WHERE
CITIZEN.NID=PATIENT.NID;

--Task-f--
SELECT count(*) FROM DISTRICT,DIVISION,CITIZEN WHERE
DISTRICT.DistrictName=CITIZEN.DistrictName and
DISTRICT.DivisionName=DIVISION.DivisionName;

--Task-g--
SELECT * FROM (SELECT count(*)Population FROM DISTRICT,DIVISION,CITIZEN
WHERE DISTRICT.DistrictName=CITIZEN.DistrictName and
DISTRICT.DivisionName=DIVISION.DivisionName ORDER BY Population DESC)
WHERE ROWNUM<=5;

--Task-h--
SELECT
DISTRICT.DistrictName,DISTRICT.DistrictSize,count(*)AccidentsPerDistrict
FROM ACCIDENT,DISTRICT WHERE ACCIDENT.DistrictName=DISTRICT.DistrictName
and ACCIDENT.DistrictSize=DISTRICT.DistrictSize GROUP BY
DISTRICT.DistrictName,DISTRICT.DistrictSize;

--Task-i--

```

```

SELECT  DIVISION.DivisionName  FROM  DISTRICT,DIVISION,ACCIDENT  WHERE
ACCIDENT.DistrictName=DISTRICT.DistrictName  and
ACCIDENT.DistrictSize=DISTRICT.DistrictSize  and
DISTRICT.DivisionName=DIVISION.DivisionName  and
DISTRICT.DivisionSize=DIVISION.DivisionSize  GROUP  BY
DIVISION.DivisionName;

--Task-j--
SELECT  LicenseType,count(*)NumberOfAccidents  FROM  ACCIDENT_LICENSE JOIN
LICENSE ON ACCIDENT_LICENSE.NID=LICENSE.NID GROUP BY LICENSE.LicenseType;

--Task-k--
SELECT  CITIZEN.NID,max(PATIENT.ReleaseDate-PATIENT.DateOfAdmission)  FROM
CITIZEN,PATIENT  WHERE  CITIZEN.NID=PATIENT.NID  GROUP  BY
CITIZEN.NID,PATIENT.ReleaseDate,PATIENT.DateOfAdmission;

--Task-l--
SELECT  min(YoungPopulation)  FROM  (select  count(*)YoungPopulation  FROM
(SELECT
CITIZEN.NID,CITIZEN.DistrictName,CITIZEN.DistrictSize,ROUND(((SELECT
SYSDATE FROM DUAL)-DOB)/365) as AGE FROM CITIZEN),DISTRICT,DIVISION WHERE
DISTRICT.DivisionName=DIVISION.DivisionName  AND
DISTRICT.DivisionSize=DIVISION.DivisionSize  WHERE  GROUP  BY
DIVISION.DivisionName,DIVISION.DivisionSize HAVING AGE>=15 AND AGE<=30);

--Task-m--
SELECT  count(*)ExpiredLicenseCount  FROM  LICENSE WHERE (SELECT SYSDATE FROM
DUAL)>LICENSE.ExpirationDate;

--Task-n--
SELECT  count(*)NumberOfAccidentsByExpired  FROM  ACCIDENT_LICENSE,(SELECT  *
FROM  LICENSE  WHERE  (SELECT  SYSDATE  FROM
DUAL)>LICENSE.ExpirationDate)ExpiredLicenses  WHERE
ACCIDENT_LICENSE.NID=ExpiredLicenses.NID GROUP BY ACCIDENT_LICENSE.NID;

```

```

--Task-o--LicenseHolders or Active License Holders?
SELECT count(*) FROM (SELECT NID FROM LICENSE minus SELECT NID FROM
ACCIDENT_LICENSE);

--Task-p--
SELECT      DIVISION.DivisionName,DIVISION.DivisionSize,sum(numberOfDeaths)
FROM          ACCIDENT,DISTRICT,DIVISION              WHERE
ACCIDENT.DistrictName=DISTRICT.DistrictName          AND
ACCIDENT.DistrictSize=DISTRICT.DistrictSize           AND
DISTRICT.DivisionName=DIVISION.DivisionName           AND
DISTRICT.DivisionSize=DIVISION.DivisionSize           GROUP BY
DIVISION.DivisionName,DIVISION.DivisionSize;

--Task-q--
SELECT AGE FROM (SELECT ROUND(((LICENSE.IssueDate-CITIZEN.DOB)/365))AGE
FROM LICENSE,CITIZEN WHERE LICENSE.NID=CITIZEN.NID) WHERE AGE<22 OR
AGE>40;

--Task-r--
SELECT      *          FROM          ACCIDENT_CITIZEN,PATIENT          WHERE
ACCIDENT_CITIZEN.NID=PATIENT.NID                      AND
ACCIDENT_CITIZEN.dateOfAccident=PATIENT.DateOfAdmission;

--Task-s--
SELECT      HOSPITAL.HospitalName,max(PatientCount)      FROM          (SELECT
HOSPITAL.HospitalName,count(*)PatientCount FROM HOSPITAL,DISTRICT,DIVISION
WHERE          HOSPITAL.HospitalDistrictName=DISTRICT.DistrictName          AND
HOSPITAL.HospitalDistrictSize=DISTRICT.DistrictSize          AND
DISTRICT.DivisionName='Dhaka' GROUP BY HOSPITAL.HospitalName);

--Task-t--

```

```
SELECT      *      FROM      ACCIDENT_CITIZEN,CITIZEN      WHERE  
ACCIDENT_CITIZEN.NID=CITIZEN.NID      AND  
ACCIDENT.DistrictName<>CITIZEN.DistrictName      AND  
ACCIDENT.DistrictSize<>CITIZEN.DistrictSize;
```

Findings

Since I did not have a data sample I couldn't test most of the queries what I did test in a limited scope was the age calculation turns out when I subtract dates from Sysdate we get a number corresponding to the number of dates which we can divide by 365 and apply ROUND function to obtain the age. Other than that most of the queries are based on intuition and I also used natural joins via the Cartesian Method rather than the typical JOIN keyword here. What I do have confusion about is whether the aggregate functions, and group by work on the various computed tables. Such as I applied filters via the WHERE clause to AGE which is a derived attribute or computed rather than stored attribute.

Summary

The complete version of the code is available in my GitHub repository. [Click Here](#).