Concordia University
Comp 353- Databases

Main Project
A Simple database for the education health facilities

Group: fdc353_1

Andre Hei Wang Law 40175600
Andrei Barbulescu 40208635
Dimitri El-Choueiry 40070184
Ziyi Wang 40167122

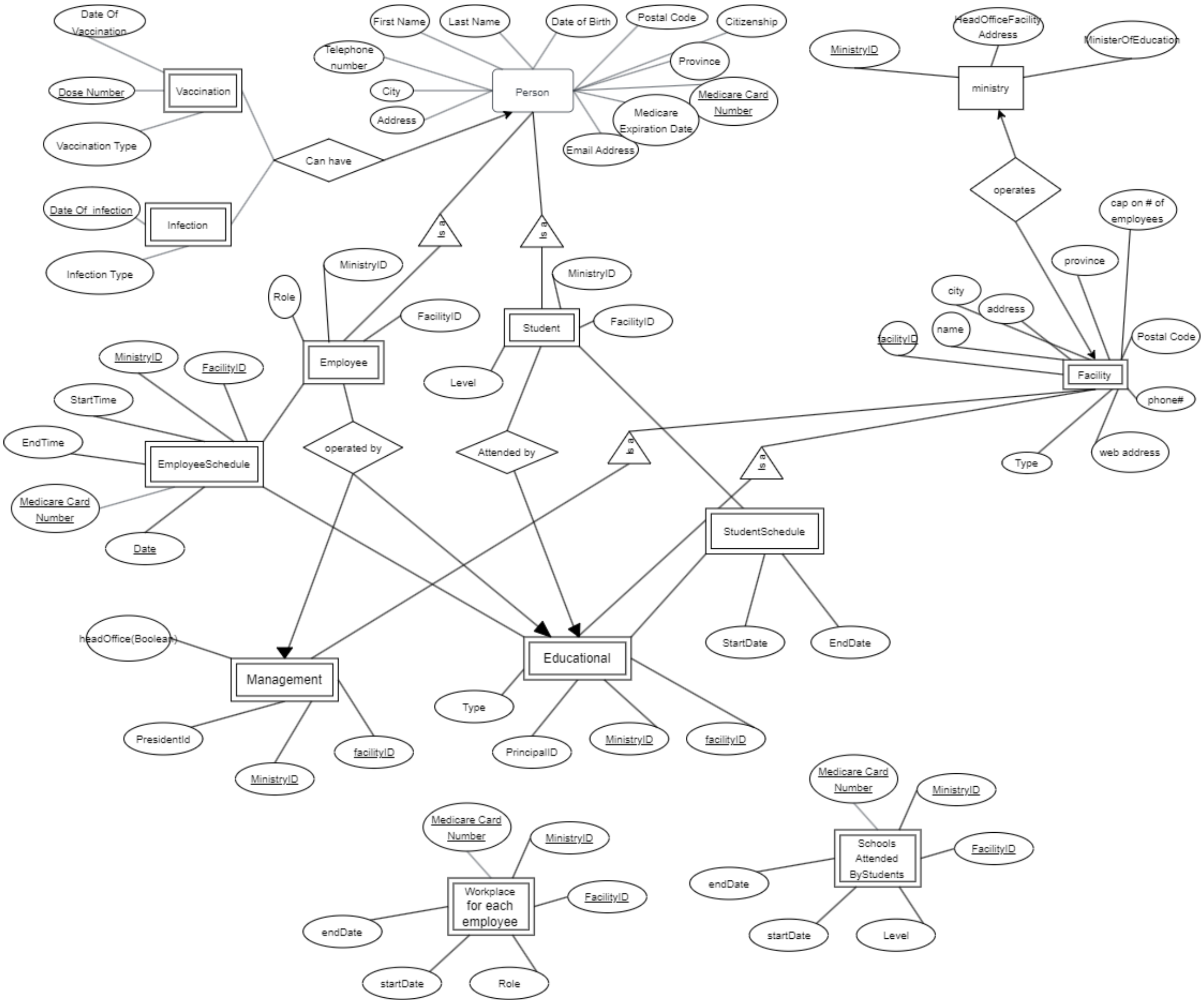
Due: August 11th, 2023 at 12:00

Tables of Content

1. Reasonable Assumptions

- No null values for any primary or foreign key within our database.
- the secondary teachers cannot have a second role as explained in the guidelines.
- Two people can’t have same phone #
- Two people can’t have same email address

2. E/R Diagram


Date Of Vaccination
Dose Number
Vaccination
Vaccination Type
Can have
Date Of infection
Infection
Infection Type
First Name
Last Name
Date of Birth
Postal Code
Citizenship
Telephone number
Province
City
Person
Medicare Card Number
Address
Medicare Expiration Date
Email Address
MinistryID
HeadOfficeFacility Address
MinisterOfEducation
ministry
operates
cap on # of employees
province
city
address
name
facilityID
Postal Code
Facility
phone#
Type
web address
Is a
Is a
MinistryID
Role
FacilityID
Employee
MinistryID
FacilityID
Student
Level
MinistryID
FacilityID
StartTime
EndTime
EmployeeSchedule
Medicare Card Number
Date
operated by
Attended by
Is a
Is a
StudentSchedule
StartDate
EndDate
headOffice(Boolean)
Management
PresidentId
MinistryID
facilityID
Educational
Type
PrincipalID
MinistryID
facilityID
Medicare Card Number
MinistryID
Schools Attended ByStudents
FacilityID
endDate
startDate
Level
Medicare Card Number
MinistryID
Workplace for each employee
FacilityID
endDate
startDate
Role

3. DB Design with Normalization

Person (FirstName, LastName, DateOfBirth, Address, City, Province, PostalCode, Citizenship, MedicareCardNumber (PK), emailAddress, TelephoneNumber, MedicareExpirationDate)

Student(MedicareCardNumber (PK,FK), FacilityID (FK), MinistryID(FK), level) - 3NF

Employee(Role, MedicareCardNumber (PK,FK), FacilityID, MinistryID) - 3NF

SchoolsAttentedByStudents(MedicareCardNumber (PK,FK), FacilityID (FK), MinistryID(FK), level, startDate, endDate)

WorkplaceForEachEmployee((Role, MedicareCardNumber (PK,FK), FacilityID(PK,FK), MinistryID(PK,FK), startDate, endDate)

Infection(DateOfInfection, MedicareCardNumber(FK), infectionType) - 3NF

Vaccination(DoseNumber (PK), Date, Type, MedicareCardNumber(FK)) : F={MCN,DN->Date, type ___ MCN,Date -> DN,Type) - 3NF

Ministry(ministryID(PK),headOfficeFacilityAddress(FK), ministerOfEducation(president of head office FK)) F={ministryID->everything, ministerOfEdu->everything) 3NF

Facility(FacilityID (PK),MinistryID(PK)(FK), Name, Address, City, Province, CapOnEmployees, PostalCode, Phone, WebAddress, Type) ____ F={FacIDMinID->everything, phone#->everything, webAdd->everything) - 3NF because all the FD LHS are superkeys also this is the canonical cover of the relation

Educational(FacilityID(PK,FK),MinistryID principleID(personID), type) -
F = {FacIDMinID->everything, principle->everything) 3NF

Management(FacilityID (PK,FK),MinistryID(PK,FK), presidentID(personID)(FK), headOffice (boolean)) - 3NF F={facIDminID-> everything, presID->everything)

EmployeeSchedule(FacilityID (FK),MinistryID(FK), MedicareCardNumber (FK), Date, StartTime EndTime) - 3NF

4. DDL Creation

```
CREATE TABLE Person (
    MedicareCardNumber INT PRIMARY KEY NOT NULL,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    DateOfBirth DATE NOT NULL,
    Address VARCHAR(100) NOT NULL,
    City VARCHAR(50) NOT NULL,
    Province VARCHAR(50) NOT NULL,
    PostalCode VARCHAR(10) NOT NULL,
    Citizenship VARCHAR(50) NOT NULL,
    EmailAddress VARCHAR(100) NOT NULL,
    TelephoneNumber VARCHAR(20) NOT NULL,
    MedicareExpirationDate DATE NOT NULL
);

CREATE TABLE Ministry (
    MinistryID INT PRIMARY KEY NOT NULL,
    HeadOfficeFacilityAddress VARCHAR(100) NOT NULL,
    MinisterOfEducation INT NOT NULL,
    FOREIGN KEY (MinisterOfEducation) REFERENCES Person(MedicareCardNumber)
);

CREATE TABLE Facility (
    FacilityID INT NOT NULL,
    MinistryID INT NOT NULL,
    Name VARCHAR(100) NOT NULL,
    Address VARCHAR(100) NOT NULL,
    City VARCHAR(50) NOT NULL,
    Province VARCHAR(50) NOT NULL,
    CapOnEmployees INT NOT NULL,
    PostalCode VARCHAR(10) NOT NULL,
    Phone VARCHAR(20) NOT NULL,
    WebAddress VARCHAR(100) NOT NULL,
    Type VARCHAR(50) NOT NULL,
    FOREIGN KEY (MinistryID) REFERENCES Ministry(MinistryID),
    PRIMARY KEY (FacilityID, MinistryID)
);

CREATE TABLE Infection (
    DateOfInfection DATE NOT NULL,
    MedicareCardNumber INT NOT NULL,
    InfectionType VARCHAR(50) NOT NULL,
    FOREIGN KEY (MedicareCardNumber) REFERENCES Person(MedicareCardNumber),
    PRIMARY KEY (MedicareCardNumber, DateOfInfection)
);

CREATE TABLE Vaccination (
    DoseNumber INT NOT NULL,
    Date DATE NOT NULL,
    Type VARCHAR(50) NOT NULL,
    MedicareCardNumber INT NOT NULL,
    FOREIGN KEY (MedicareCardNumber) REFERENCES Person(MedicareCardNumber),
    PRIMARY KEY (DoseNumber, MedicareCardNumber)
);

CREATE TABLE Educational (
    FacilityID INT NOT NULL ,
    MinistryID INT NOT NULL,
    Principle INT NOT NULL,
    Type VARCHAR(50) NOT NULL,
    FOREIGN KEY (FacilityID, MinistryID) REFERENCES Facility(FacilityID, MinistryID),
    FOREIGN KEY (Principle) REFERENCES Person(MedicareCardNumber),
    PRIMARY KEY (FacilityID, MinistryID)
);

CREATE TABLE Student (
    MedicareCardNumber INT PRIMARY KEY NOT NULL,
    FacilityID INT NOT NULL,
    MinistryID INT NOT NULL,
    Level VARCHAR(50) NOT NULL,
    FOREIGN KEY (FacilityID, MinistryID) REFERENCES Facility(FacilityID, MinistryID),
    FOREIGN KEY (MedicareCardNumber) REFERENCES Person(MedicareCardNumber)
);

CREATE TABLE Employee (
    Role VARCHAR(50) NOT NULL,
    MedicareCardNumber INT PRIMARY KEY NOT NULL,
    FacilityID INT NOT NULL,
    MinistryID INT NOT NULL,
    FOREIGN KEY (FacilityID, MinistryID) REFERENCES Facility(FacilityID, MinistryID),
    FOREIGN KEY (MedicareCardNumber) REFERENCES Person(MedicareCardNumber)
);

CREATE TABLE Management (
    FacilityID INT NOT NULL,
    MinistryID INT NOT NULL,
    PresidentID INT NOT NULL,
    HeadOffice BOOLEAN NOT NULL,
    PRIMARY KEY (FacilityID, MinistryID),
    FOREIGN KEY (FacilityID, MinistryID) REFERENCES Facility(FacilityID, MinistryID),
    FOREIGN KEY (PresidentID) REFERENCES Person(MedicareCardNumber)
);

CREATE TABLE SchoolsAttendedByStudents (
    MedicareCardNumber INT NOT NULL,
    FacilityID INT NOT NULL,
    MinistryID INT NOT NULL,
    Level VARCHAR(50) NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE,
    PRIMARY KEY (MedicareCardNumber, FacilityID, MinistryID),
    FOREIGN KEY (FacilityID, MinistryID) REFERENCES Facility(FacilityID, MinistryID),
    FOREIGN KEY (MedicareCardNumber) REFERENCES Person(MedicareCardNumber)
);

CREATE TABLE WorkplaceForEachEmployee (
    Role VARCHAR(50) NOT NULL,
    MedicareCardNumber INT NOT NULL,
    FacilityID INT NOT NULL,
    MinistryID INT NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE,
    PRIMARY KEY (MedicareCardNumber, FacilityID, MinistryID),
    FOREIGN KEY (FacilityID, MinistryID) REFERENCES Facility(FacilityID, MinistryID),
    FOREIGN KEY (MedicareCardNumber) REFERENCES Person(MedicareCardNumber)
);

CREATE TABLE EmployeeSchedule (
    FacilityID INT NOT NULL,
    MinistryID INT NOT NULL,
    MedicareCardNumber INT NOT NULL,
    Date DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME NOT NULL,
    PRIMARY KEY (FacilityID, MinistryID, MedicareCardNumber, Date),
    FOREIGN KEY (FacilityID, MinistryID) REFERENCES Facility(FacilityID, MinistryID),
    FOREIGN KEY (MedicareCardNumber) REFERENCES Person(MedicareCardNumber)
);
```

5. SQL Queries

Query 1-7

Refer to: /www/groups/f/fd_comp353_1 or the attached zip file with all the html and php codes.

Query 8

```
SELECT
    F.Name,
    F.Address,
    F.City,
    F.Province,
    F.PostalCode,
    F.Phone,
    F.WebAddress,
    F.Type,
    F.CapOnEmployees AS Capacity,
CASE
    WHEN F.Type = 'educational' THEN E.principle
    WHEN F.Type = 'management' THEN M.presidentID
    ELSE NULL
    END AS PrincipalOrPresidentID,
CASE
    WHEN F.Type = 'educational' THEN (SELECT FirstName FROM Person WHERE
MedicareCardNumber = E.principle)
    WHEN F.Type = 'management' THEN (SELECT FirstName FROM Person
WHERE MedicareCardNumber = M.presidentID)
    ELSE NULL
    END AS FirstName,
CASE
    WHEN F.Type = 'educational' THEN (SELECT LastName FROM Person WHERE
MedicareCardNumber = E.principle)
    WHEN F.Type = 'management' THEN (SELECT LastName FROM Person
WHERE MedicareCardNumber = M.presidentID)
    ELSE NULL
    END AS LastName,
(SELECT COUNT(*) FROM Employee WHERE FacilityID = F.FacilityID) AS
NumberOfEmployees
FROM
    Facility F
LEFT JOIN Educational E ON F.FacilityID = E.FacilityID AND F.MinistryID = E.MinistryID
```

```
    LEFT JOIN Management M ON F.FacilityID = M.FacilityID AND F.MinistryID =
    M.MinistryID
    ORDER BY
        F.Province ASC,
        F.City ASC,
        F.Type ASC,
        NumberOfEmployees ASC;
```

Query 9

```
    SELECT
        E.Role,
        P.FirstName,
        P.LastName,
        W.StartDate,
        P.DateOfBirth,
        P.MedicareCardNumber,
        P.TelephoneNumber,
        P.Address,
        P.City,
        P.Province,
        P.PostalCode,
        P.Citizenship,
        P.EmailAddress
    FROM Employee AS E
    INNER JOIN Person AS P ON E.MedicareCardNumber = P.MedicareCardNumber
    INNER JOIN WorkplaceForEachEmployee AS W ON E.MedicareCardNumber =
W.MedicareCardNumber
    WHERE W.FacilityID = $facilityID
    ORDER BY E.Role ASC, P.FirstName ASC, P.LastName ASC
```

Query 10

```
    SELECT
        F.Name AS FacilityName,
        ES.Date,
        ES.StartTime,
        ES.EndTime
    FROM EmployeeSchedule AS ES
    JOIN Facility AS F ON ES.FacilityID = F.FacilityID AND ES.MinistryID = F.MinistryID
    WHERE ES.MedicareCardNumber = ? AND ES.Date BETWEEN ? AND ?
    ORDER BY F.Name, ES.Date, ES.StartTime
```

Query 11

```
    SELECT
        P.FirstName,
        P.LastName,
        I.DateOfInfection,
        F.Name AS FacilityName
    FROM
        Infection AS I
    JOIN
        Employee AS E ON I.MedicareCardNumber = E.MedicareCardNumber
    JOIN
        Person AS P ON I.MedicareCardNumber = P.MedicareCardNumber
    JOIN
        Facility AS F ON E.FacilityID = F.FacilityID AND
        E.MinistryID = F.MinistryID
    WHERE
        I.DateOfInfection >= DATE_SUB(CURRENT_DATE(), INTERVAL 2 WEEK) AND
        E.Role = 'Teacher' AND
        I.InfectionType = 'COVID-19'
    GROUP BY
        P.FirstName
    ORDER BY
        F.Name ASC, P.FirstName ASC
```

Query 12

```
SELECT * FROM EmailLog WHERE SenderFacilityName = ? ORDER BY DateSent ASC"
```

Query 13

```
    SELECT DISTINCT
        Person.FirstName,
        Person.LastName,
        Employee.Role
    FROM Person
    INNER JOIN Employee ON Person.MedicareCardNumber =
Employee.MedicareCardNumber
    INNER JOIN EmployeeSchedule ON Employee.MedicareCardNumber =
EmployeeSchedule.MedicareCardNumber
    WHERE EmployeeSchedule.FacilityID = ? AND EmployeeSchedule.Date >= ?
```

```
        ORDER BY Employee.Role ASC, Person.FirstName ASC
```

Query 14

```
SELECT Person.FirstName, Person.LastName, EmployeeSchedule.MedicareCardNumber,
                SUM(TIME_TO_SEC(TIMEDIFF(EndTime, StartTime))) / 3600 AS TotalHours
                FROM Person
                INNER JOIN Employee ON Person.MedicareCardNumber =
Employee.MedicareCardNumber
                INNER JOIN EmployeeSchedule ON Employee.MedicareCardNumber =
EmployeeSchedule.MedicareCardNumber
                WHERE EmployeeSchedule.FacilityID = ? AND EmployeeSchedule.Date
BETWEEN ? AND ?
                GROUP BY Person.FirstName, Person.LastName,
EmployeeSchedule.MedicareCardNumber
                ORDER BY Person.FirstName ASC, Person.LastName ASC
```

Query 15

```
        SELECT
                F.Province,
                F.Name AS SchoolName,
                F.CapOnEmployees AS Capacity,
                COUNT(DISTINCT I.MedicareCardNumber) AS InfectedTeachers,
                COUNT(DISTINCT S.MedicareCardNumber) AS InfectedStudents
        FROM Facility F
        LEFT JOIN
                Employee e ON F.FacilityID = e.FacilityID AND
                F.MinistryID = e.MinistryID
        LEFT JOIN
                Infection I ON e.MedicareCardNumber = I.MedicareCardNumber AND
                I.DateOfInfection BETWEEN '$twoWeeksAgo' AND '$currentDate'
        LEFT JOIN
                Student S ON F.FacilityID = S.FacilityID AND
                F.MinistryID = S.MinistryID
        LEFT JOIN
                Infection st ON S.MedicareCardNumber = st.MedicareCardNumber AND
                st.DateOfInfection BETWEEN '$twoWeeksAgo' AND '$currentDate'
        WHERE
                F.Type = 'Education' AND
                I.InfectionType = 'COVID-19' AND
                st.InfectionType = 'COVID-19' AND
                S.Level = 'High School'
```

```sql
GROUP BY
        F.FacilityID, F.MinistryID, F.Province, F.Name, F.CapOnEmployees
ORDER BY
        F.Province ASC, InfectedTeachers ASC;
```

Query 16

```sql
SELECT
        min.MinistryID AS MinistryID,
        p.FirstName AS MinisterFirstName,
        p.LastName AS MinisterLastName,
        p.City AS MinisterCity,
        COUNT(DISTINCT m.FacilityID) AS TotalManagementFacilities,
        COUNT(DISTINCT e.FacilityID) AS TotalEducationalFacilities
FROM Ministry AS min
JOIN Person AS p ON min.MinisterOfEducation = p.MedicareCardNumber
LEFT JOIN Facility AS f ON min.MinistryID = f.MinistryID
LEFT JOIN Management AS m ON f.FacilityID = m.FacilityID AND f.MinistryID =
m.MinistryID AND min.MinisterOfEducation = m.PresidentID
LEFT JOIN Educational AS e ON f.FacilityID = e.FacilityID AND f.MinistryID =
e.MinistryID
GROUP BY min.MinistryID
HAVING COUNT(DISTINCT m.FacilityID) > 0
ORDER BY p.City ASC, TotalEducationalFacilities DESC;
```

Query 20

```sql
SELECT
        F.Name AS FacilityName,
        F.Address AS FacilityAddress,
        P.FirstName AS EmployeeFirstName,
        P.LastName AS EmployeeLastName,
        P.EmailAddress AS EmployeeEmail,
        S.Date AS ScheduleDate,
        S.StartTime AS ScheduleStartTime,
        S.EndTime AS ScheduleEndTime
    FROM Facility F
    JOIN EmployeeSchedule S ON F.FacilityID = S.FacilityID AND
F.MinistryID = S.MinistryID
    JOIN Person P ON S.MedicareCardNumber = P.MedicareCardNumber
    WHERE S.Date BETWEEN ? AND ?
```

6. CONTRIBUTIONS

Reasonable Assumptions (Andrei Barbulescu)
Conceptual DB design (Andrei Barbulescu and Dimitri El-Choueiry)
E/R diagram (Dimitri El-Choueiry and Andrei Barbulescu)
E/R to DB schema conversion (Andrei Barbulescu and Dimitri El-Choueiry)
Normalizing Relations and Refining DB schema - 3NF/BCNF (Dimitri El-Choueiry)
DDL (Andrei Barbulescu and Dimitri El-Choueiry)

Queries:
Ziyi: 1,13,14,12,20
Andrei : 3,7,9,10
Andre: 4,5,11,15
Dimitri: 2,6,8,16