



## **Department of Computer Science & Software Engineering**

### **Main Project**

**COMP 353 - Section CC Databases  
Instructor: Prof. Khaled Jababo  
Summer 2023**

**Project done by:  
Group jdc353\_1  
Yujing Wang - 40098623  
Yifan Wu - 40153584  
Wei Su - 27072473  
Mikelson Benjamin - 40158499**

**Submitted:**

**Aug 11th, 2023**

## **Faculty of Engineering and Computer Science**

### **Expectations of Originality**

This form sets out the requirements for originality for work submitted by students in the Faculty of Engineering

and Computer Science. Submissions such as assignments, lab reports, project reports, computer programs and

take-home exams must conform to the requirements stated on this form and to the Academic Code of Conduct.

The course outline may stipulate additional requirements for the course.

1. Your submissions must be your own original work. Group submissions must be the original work of the

students in the group.

2. Direct quotations must not exceed 5% of the content of a report, must be enclosed in quotation marks,

and must be attributed to the source by a numerical reference citation<sup>1</sup>. Note that engineering reports

rarely contain direct quotations.

3. Material paraphrased or taken from a source must be attributed to the source by a numerical reference

citation.

4. Text that is inserted from a web site must be enclosed in quotation marks and attributed to the web site

by numerical reference citation.

5. Drawings, diagrams, photos, maps or other visual material taken from a source must be attributed to that

source by a numerical reference citation.

6. No part of any assignment, lab report or project report submitted for this course can be submitted for

any other course.

7. In preparing your submissions, the work of other past or present students cannot be consulted, used,

copied, paraphrased or relied upon in any manner whatsoever.

8. Your submissions must consist entirely of your own or your group's ideas, observations, calculations,

information and conclusions, except for statements attributed to sources by numerical citation.

9. Your submissions cannot be edited or revised by any other student.

10. For lab reports, the data must be obtained from your own or your lab group's experimental work.

11. For software, the code must be composed by you or by the group submitting the work, except for code

that is attributed to its sources by numerical reference.

You must write one of the following statements on each piece of work that you submit:

For individual work: "I certify that this submission is my original work and meets the Faculty's Expectations of Originality", with your signature, I.D. #, and the date.

For group work: "We certify that this submission is the original work of members of the group and meets

the Faculty's Expectations of Originality", with the signatures and I.D. #s of all the team members and the

date.

A signed copy of this form must be submitted to the instructor at the beginning of the semester in each course.

I certify that I have read the requirements set out on this form, and that I am aware of these requirements. I

certify that all the work I will submit for this course will comply with these requirements and with additional

requirements stated in the course outline.

Course Number: COMP 353 section CC

Instructor: Khaled Jababo

Name: Yifan Wu

I.D. # 40153584

Signature: Yifan Wu

Date: Aug 11th, 2023

Name: Wei Su

I.D. # 27072473

Signature: Wei Su

Date: Aug 11th, 2023

Name: Yujing Wang

I.D. # 40098623

Signature: Yujing Wang

Date: Aug 11th, 2023

Name: Mikelson Benjamin

I.D. # 40158499

Signature: Mikelson Benjamin

Date: Aug 11th, 2023

1 Rules for reference citation can be found in “Form and Style” by Patrick MacDonagh and Jack Bordan, fourth edition, May, 2000,

available at <http://www.elps.concordia.ca/scs/Forms/Form&Style.pdf>.

Approved by the ENCS Faculty Council February 10, 2012

# **Table of Contents**

**Expectation of Originality**

**E/R - Diagram**

**E/R - Diagram to Relation Conversions**

**Normalization**

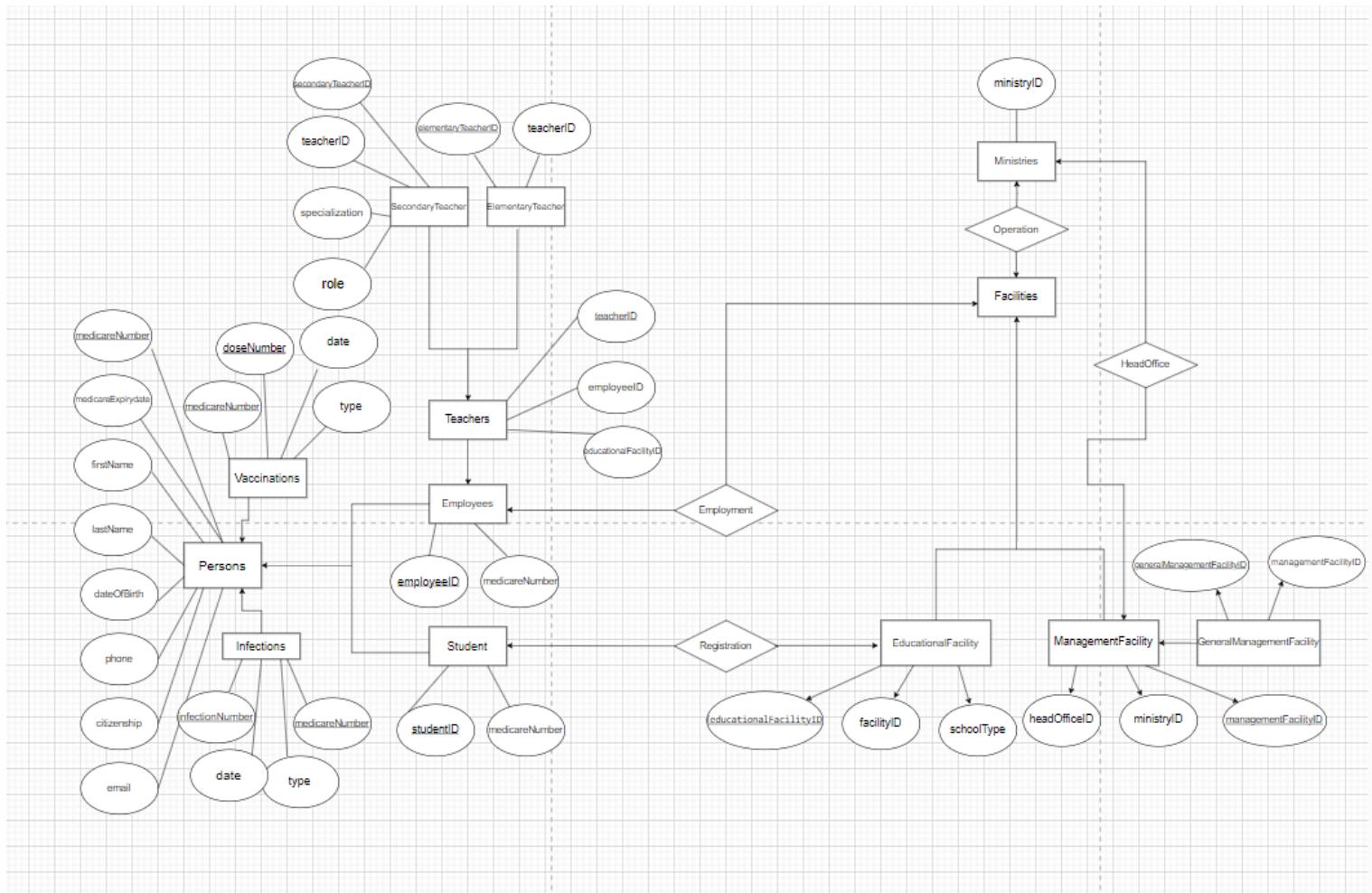
**SQL Declarations**

**SQL Queries**

**Implemented Functionalities**

**Contributions**

## 1. Conceptual DB Design (E/R)



## **2. Relational Database Schema**

Persons (medicareNumber, medicareExpiryDate, firstName, lastName, dateOfBirth, phone, citizenship, email)

Employees (employeeID, medicareNumber)

### **PERSON-EMPLOYEE IS 1:1**

Students (studentID, medicareNumber)

### **PERSON-STUDENT IS 1:1**

Vaccinations (doseNumber, medicareNumber, date, type)

### **VACCINATIONS-PERSON IS N:1**

Infections (infectionNumber, medicareNumber, date, type)

### **INFECTIONS-PERSON IS N:1**

Ministries (ministryID, sector)

Facilities (facilityID, name, phone, webAddress, capacity)

### **MINISTRIES-FACILITIES IS 1:N**

Operation (facilityID, ministryID)

EducationalFacility (educationalFacilityID, facilityID, schoolType)

### **FACILITIES-EDUCATIONALFACILITY IS 1:1**

ManagementFacility (managementFacilityID, facilityID, president)

### **FACILITIES-MANAGEMENTFACILITY IS 1:1**

HeadOffice (managementFacilityID, ministryID, headOfficeID)

### **MANAGEMENTFACILITY-HEADOFFICE IS 1:1**

GeneralManagementFacility(generalManagementFacilityID, managementFacilityID)

### **MANAGEMENTFACILITY-GENERALMANAGEMENTFACILITY IS 1:1**

Employment (shiftID, employeeID, facilityID, startDate, endDate, date, startTime, endTime, role)

**EMPLOYMENT-EMPLOYEE IS N:1**

Registration (studentID, facilityID, startDate, endDate, level)

**REGISTRATION-STUDENT IS N:1**

Teachers (teacherID, employeeID, educationalFacilityID)

**TEACHERS-EMPLOYEE IS 1:1**

SecondaryTeacher (secondaryTeacherID, teacherID, specialization, role)

**SECONDARYTEACHER-TEACHERS IS 1:1**

ElementaryTeacher (elementaryTeacherID, teacherID)

**ELEMENTARYTEACHER-TEACHERS IS 1:1**

Location (addressID, address, city, province, postalCode)

Residence (medicareNumber, addressID)

**RESIDENCE-LOCATION IS N:1**

Situated (facilityID, addressID)

**SITUATED-LOCATION IS 1:1**

Schedule (employeeID, scheduleDate, startTime, endtime, facilityID)

**EMPLOYEES-SCHEDULE-FACILITIES IS N:N**

Contract (employeeID, startDate, endDate, facilityID, role)

**EMPLOYEES-SCHEDULE-CONTRACT IS N:N**

EmailLog (emailID, facilityID, employeeID, sender, receiverFirstName, receiverLastName, date, subject, body)

**EMPLOYEES-EMAILLOG-FACILITIES IS N:N**

## **Functional dependencies**

For the Persons table:

{medicareNumber} → {medicareExpiryDate, firstName, lastName, dateOfBirth, phone, citizenship, email}

For the Employees table:

{employeeID} → {medicareNumber}

For the Students table:

{studentID} → {medicareNumber}

For the Vaccinations table:

{doseNumber, medicareNumber} → {date, type}

For the Infections table:

{infectionNumber, medicareNumber} → {date, type}

For the Ministries table:

{ministryID} → {sector}

For the Facilities table:

{facilityID} → {name, phone, webAddress, capacity}

For the Operation table:

{facilityID} -> {ministryID}

For the EducationalFacility table:

{educationalFacilityID} -> {facilityID, schoolType}

For the ManagementFacility table:

{managementFacilityID} -> {facilityID, president}

For the HeadOffice table:

{managementFacilityID} -> {ministryID, headOfficeID}

For the GeneralManagementFacility table:

{generalManagementFacilityID} -> {managementFacilityID}

For the Employment table:

{shiftID} -> {employeeID, facilityID, startDate, endDate, date, startTime, endTime, role}

For the Registration table:

{studentID} -> {facilityID, startDate, endDate, level}

For the Teachers table:

{teacherID} -> {employeeID, educationalFacilityID}

For the SecondaryTeacher table:

{secondaryTeacherID} -> {teacherID, specialization, role}

For the ElementaryTeacher table:

{elementaryTeacherID} -> {teacherID}

For the Location table:

{addressID} -> {address, city, province, postalCode}

For the Residence table:

{medicareNumber} -> {addressID}

For the Situated table:

{facilityID} -> {addressID}

For the Schedule table:

{employeeID, scheduleDate} -> {startTime, endTime}

For the Contract table:

{employeeID, startDate, endDate} -> {facilityID, role}

For the EmailLog table:

{emailID} -> {facilityID, employeeID, date, subject, body}

## DDL Statements

```
CREATE TABLE Persons (
    medicareNumber INT PRIMARY KEY,
    medicareExpiryDate DATE NOT NULL,
    firstName VARCHAR(255) NOT NULL,
    lastName VARCHAR(255) NOT NULL,
    dateOfBirth DATE NOT NULL,
    phone BIGINT UNSIGNED NOT NULL,
    citizenship VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE Employees (
    employeeID INT AUTO_INCREMENT PRIMARY KEY,
    medicareNumber INT,
    FOREIGN KEY (medicareNumber) REFERENCES Persons(medicareNumber) ON DELETE CASCADE
);
```

```
CREATE TABLE Students (
    studentID INT AUTO_INCREMENT PRIMARY KEY,
    medicareNumber INT,
    FOREIGN KEY (medicareNumber) REFERENCES Persons(medicareNumber) ON DELETE CASCADE
);
```

```
CREATE TABLE Vaccinations (
    doseNumber INT NOT NULL,
    medicareNumber INT,
    date DATE NOT NULL,
    type VARCHAR(255) NOT NULL,
    FOREIGN KEY (medicareNumber) REFERENCES Persons(medicareNumber) ON DELETE CASCADE,
    PRIMARY KEY (doseNumber, medicareNumber)
);
```

```
CREATE TABLE Infections (
    infectionID INT AUTO_INCREMENT PRIMARY KEY,
    infectionNumber INT NOT NULL,
    medicareNumber INT,
    date DATE NOT NULL,
    type VARCHAR(255) NOT NULL,
    FOREIGN KEY (medicareNumber) REFERENCES Persons(medicareNumber) ON DELETE CASCADE
);
```

```
CREATE TABLE Ministries (
    ministryID INT PRIMARY KEY,
    sector VARCHAR(255)
);
```

```
CREATE TABLE Facilities (
```

```
facilityID INT AUTO_INCREMENT PRIMARY KEY,  
name VARCHAR(255) NOT NULL,  
phone BIGINT UNSIGNED NOT NULL,  
webAddress VARCHAR(255) NOT NULL,  
capacity INT UNSIGNED NOT NULL  
);
```

```
CREATE TABLE Operation (  
ministryID INT,  
facilityID INT,  
FOREIGN KEY (facilityID) REFERENCES Facilities(facilityID) ON DELETE CASCADE,  
FOREIGN KEY (ministryID) REFERENCES Ministries(ministryID) ON DELETE CASCADE,  
PRIMARY KEY (facilityID)  
);
```

```
CREATE TABLE ManagementFacility (  
managementFacilityID INT AUTO_INCREMENT PRIMARY KEY,  
facilityID INT,  
president INT NOT NULL,  
FOREIGN KEY (facilityID) REFERENCES Facilities(facilityID) ON DELETE CASCADE,  
FOREIGN KEY (president) REFERENCES Employees(employeeID) ON DELETE CASCADE  
);
```

```
CREATE TABLE GeneralManagementFacility (
```

```
generalManagementFacilityID INT AUTO_INCREMENT PRIMARY KEY,  
managementFacilityID INT,  
FOREIGN KEY (managementFacilityID) REFERENCES ManagementFacility(managementFacilityID)  
ON DELETE CASCADE  
);
```

```
CREATE TABLE HeadOffice (  
headOfficeID INT UNIQUE NOT NULL,  
managementFacilityID INT PRIMARY KEY,  
ministryID INT,  
FOREIGN KEY (managementFacilityID) REFERENCES ManagementFacility(managementFacilityID)  
ON DELETE CASCADE,  
FOREIGN KEY (ministryID) REFERENCES Ministries(ministryID) ON DELETE CASCADE  
);
```

```
CREATE TABLE EducationalFacility (  
educationalFacilityID INT AUTO_INCREMENT PRIMARY KEY,  
facilityID INT,  
schoolType ENUM ('Primary', 'Middle', 'High', 'Combo'),  
FOREIGN KEY (facilityID) REFERENCES Facilities(facilityID) ON DELETE CASCADE  
);
```

```
CREATE TABLE Registration (  
studentID INT,  
facilityID INT,
```

```
startDate DATE NOT NULL,  
endDate DATE,  
level VARCHAR(255) NOT NULL,  
FOREIGN KEY (studentID) REFERENCES Students(studentID) ON DELETE CASCADE,  
FOREIGN KEY (facilityID) REFERENCES Facilities(facilityID) ON DELETE CASCADE,  
PRIMARY KEY (studentID, facilityID, startDate)  
);
```

```
CREATE TABLE Teachers (  
teacherID INT AUTO_INCREMENT PRIMARY KEY,  
employeeID INT,  
educationalFacilityID INT,  
FOREIGN KEY (employeeID) REFERENCES Employees(employeeID) ON DELETE CASCADE,  
FOREIGN KEY (educationalFacilityID) REFERENCES EducationalFacility(educationalFacilityID) ON  
DELETE CASCADE  
);
```

```
CREATE TABLE ElementaryTeacher (  
elementaryTeacherID INT AUTO_INCREMENT PRIMARY KEY,  
teacherID INT,  
FOREIGN KEY (teacherID) REFERENCES Teachers(teacherID) ON DELETE CASCADE  
);
```

```
CREATE TABLE SecondaryTeacher (
```

```
secondaryTeacherID INT AUTO_INCREMENT PRIMARY KEY,  
teacherID INT,  
specialization VARCHAR(255) NOT NULL,  
role ENUM('Counselor', 'Director', 'Administrator'),  
FOREIGN KEY (teacherID) REFERENCES Teachers(teacherID) ON DELETE CASCADE  
);
```

```
CREATE TABLE Location (  
addressID INT AUTO_INCREMENT PRIMARY KEY,  
address INT NOT NULL,  
city VARCHAR(255) NOT NULL,  
province VARCHAR(255) NOT NULL,  
postalCode VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Residence (  
medicareNumber INT PRIMARY KEY,  
addressID INT,  
FOREIGN KEY (medicareNumber) REFERENCES Persons (medicareNumber) ON DELETE CASCADE,  
FOREIGN KEY (addressID) REFERENCES Location (addressID) ON DELETE CASCADE  
);
```

```
CREATE TABLE Situated (  
facilityID INT PRIMARY KEY,
```

```
addressID INT,  
FOREIGN KEY (facilityID) REFERENCES Facilities (facilityID) ON DELETE CASCADE,  
FOREIGN KEY (addressID) REFERENCES Location (addressID) ON DELETE CASCADE  
);
```

```
CREATE TABLE EmailLog (  
emailID INT AUTO_INCREMENT PRIMARY KEY,  
date DATE NOT NULL,  
facilityID INT,  
employeeID INT,  
sender VARCHAR(255) NOT NULL,  
receiverFirstName VARCHAR(255) NOT NULL,  
receiverLastName VARCHAR(255) NOT NULL,  
subject VARCHAR(255) NOT NULL,  
body VARCHAR(255) NOT NULL,  
FOREIGN KEY (facilityID) REFERENCES Facilities (facilityID) ON DELETE CASCADE,  
FOREIGN KEY (employeeID) REFERENCES Employees (employeeID) ON DELETE CASCADE  
);
```

```
CREATE TABLE Schedule (  
employeeID INT,  
scheduleDate DATE NOT NULL,  
startTime TIME NOT NULL,  
endTime TIME NOT NULL,
```

```
facilityID int,  
FOREIGN KEY (facilityID)  
REFERENCES Facilities (facilityID),  
FOREIGN KEY (employeeID)  
REFERENCES Employees (employeeID),  
UNIQUE (employeeID , scheduleDate , endTime),  
PRIMARY KEY (employeeID , scheduleDate , startTime)  
);
```

```
CREATE TABLE Contract (  
employeeID INT,  
facilityID INT,  
startDate DATE NOT NULL,  
endDate DATE,  
role VARCHAR(255) NOT NULL,  
FOREIGN KEY (employeeID) REFERENCES Employees(employeeID) ON DELETE CASCADE,  
FOREIGN KEY (facilityID) REFERENCES Facilities(facilityID) ON DELETE CASCADE,  
PRIMARY KEY (employeeID,facilityID,startDate)  
);
```

## DML statements

```
INSERT INTO Persons (medicareNumber, medicareExpiryDate, firstName, lastName, dateOfBirth, phone, citizenship, email)
```

```
VALUES
```

```
(5048,      '2023-09-26',      'Tom',      'Phillips',      '2002-11-04',      218038875,      'Canadian',  
'person1@example.com'),  
  
(7633,      '2022-10-07',      'Nayeli',     'Harris',      '1976-05-07',      914546857,      'German',  
'person2@example.com'),  
  
(3792,      '2022-01-26',      'Steven',     'James',      '1995-10-23',      417204837,      'American',  
'person3@example.com'),  
  
(3295,      '2025-06-12',      'Natali',     'White',      '1990-10-16',      303711351,      'French',  
'person4@example.com'),  
  
(9508,      '2022-07-15',      'Michael',    'Martin',      '2003-06-11',      856895891,      'Australian',  
'person5@example.com'),  
  
(1805,      '2022-09-19',      'Maria',      'Evans',      '1962-06-19',      958736466,      'French',  
'person6@example.com'),  
  
(6349,      '2024-11-05',      'Nicole',     'Jackson',     '1988-09-14',      777745522,      'Canadian',  
'person7@example.com'),  
  
(2068,      '2023-03-24',      'Natalya',    'Baker',      '1967-07-05',      962355955,      'German',  
'person8@example.com'),  
  
(6013,      '2024-08-22',      'Joe',        'Miller',     '1988-09-16',      295279935,      'Canadian',  
'person9@example.com'),  
  
(9671,      '2022-08-22',      'John',       'Wright',     '1976-01-11',      164511484,      'French',  
'person10@example.com'),  
  
(5721,      '2023-03-23',      'Tom',        'Martinez',    '1974-09-01',      639881652,      'German',  
'person11@example.com'),  
  
(2941,      '2023-09-28',      'Steven',     'Murphy',     '1960-09-22',      736149025,      'French',  
'person12@example.com'),  
  
(6736, '2025-06-12', 'Joe', 'Flores', '1991-07-11', 626373293, 'French', 'person13@example.com'),
```

(6185, '2025-07-08', 'Nancy', 'Cook', '1998-04-11', 335217668, 'Australian', 'person14@example.com'),
(6760, '2022-06-04', 'Steven', 'Bell', '2002-02-14', 751558232, 'Australian', 'person15@example.com'),
(3837, '2023-07-03', 'Maria', 'Russell', '1996-08-23', 957211208, 'Canadian', 'person16@example.com'),
(4997, '2023-06-26', 'Jack', 'Robinson', '1992-01-08', 581147180, 'French', 'person17@example.com'),
(2165, '2024-08-14', 'Jack', 'Roberts', '1998-04-14', 648481342, 'American', 'person18@example.com'),
(6560, '2023-10-08', 'John', 'Gonzalez', '1962-03-18', 143748391, 'Australian', 'person19@example.com'),
(6661, '2025-04-17', 'Nia', 'Murphy', '1967-02-27', 912851240, 'German', 'person20@example.com'),
(4874, '2024-07-21', 'Nathalie', 'Walker', '1998-06-18', 783500106, 'German', 'person21@example.com'),
(4233, '2025-09-09', 'Nora', 'White', '1994-01-09', 785019153, 'British', 'person22@example.com'),
(2506, '2023-02-10', 'Nathaly', 'Howard', '1977-02-19', 945519013, 'Canadian', 'person23@example.com'),
(2515, '2022-03-05', 'Nathan', 'Kelly', '1976-03-09', 278047016, 'American', 'person24@example.com'),
(4628, '2023-02-02', 'Nancy', 'Rogers', '1984-06-22', 720769265, 'French', 'person25@example.com'),
(9388, '2022-11-25', 'Steven', 'Washington', '1998-09-11', 127109103, 'French', 'person26@example.com'),
(6706, '2022-09-13', 'Nora', 'Scott', '1971-07-06', 426885138, 'German', 'person27@example.com'),
(9830, '2024-03-06', 'Nathalia', 'Sanders', '1976-10-13', 412807743, 'Canadian', 'person28@example.com'),
(1799, '2025-11-19', 'Natalya', 'Carter', '1964-11-20', 859210472, 'German', 'person29@example.com'),

'person29@example.com'),  
(3667, '2024-06-15', 'Julia', 'Kelly', '1979-07-19', 122821961, 'Australian',  
'person30@example.com'),  
(9801, '2024-10-03', 'Natalya', 'Adams', '1972-12-20', 100886725, 'French',  
'person31@example.com'),  
(1132, '2024-10-27', 'Jack', 'Alexander', '2001-02-07', 997905683, 'American',  
'person32@example.com'),  
(1556, '2023-05-28', 'Mary', 'Rogers', '1985-05-14', 581046349, 'American',  
'person33@example.com'),  
(5728, '2024-07-11', 'Nia', 'Cook', '1975-04-26', 681659378, 'American',  
'person34@example.com'),  
(3682, '2024-11-13', 'Nevaeh', 'Johnson', '1995-06-11', 938775297, 'Canadian',  
'person35@example.com'),  
(5860, '2022-03-07', 'Nathan', 'Diaz', '2003-08-12', 692850501, 'British',  
'person36@example.com'),  
(6640, '2024-12-27', 'Natalia', 'Lewis', '2003-04-05', 373676914, 'French',  
'person37@example.com'),  
(8456, '2025-12-14', 'Joe', 'Phillips', '1978-10-11', 496423866, 'Canadian',  
'person38@example.com'),  
(8179, '2025-03-28', 'Noah', 'Powell', '1961-09-18', 341125934, 'Canadian',  
'person39@example.com'),  
(1947, '2024-03-13', 'Natalie', 'Williams', '1978-04-25', 821922687, 'Australian',  
'person40@example.com'),  
(9620, '2022-01-01', 'Natali', 'Carter', '1986-04-23', 101158407, 'Australian',  
'person41@example.com'),  
(5400, '2024-02-20', 'Nathan', 'Carter', '1996-05-12', 781192683, 'Canadian',  
'person42@example.com'),  
(6011, '2025-11-24', 'Tom', 'Price', '2000-11-27', 596009224, 'Australian',  
'person43@example.com'),  
(8280, '2023-10-11', 'Nina', 'Williams', '1996-05-05', 274731623, 'British',  
'person44@example.com'),

(5905, '2023-04-10', 'Nathalie', 'Cooper', '1964-02-18', 974005556, 'Canadian', 'person45@example.com');

INSERT INTO Employees (employeeID, medicareNumber)

VALUES

(1001, 1132),

(1002, 1556),

(1003, 1799),

(1004, 1805),

(1005, 1947),

(1006, 2068),

(1007, 2165),

(1008, 2506),

(1009, 2515),

(1010, 2941),

(1011, 3295),

(1012, 3667),

(1013, 3682),

(1014, 3792),

(1015, 3837),

(1016, 4233),

(1017, 4628),

(1018, 4874),

(1019, 4997),

(1020, 5048),

(1021, 5400),

(1022, 5721),

(1023, 5728),

(1024, 5860),

(1025, 5905),

(1026, 6011),

(1027, 6013),

(1028, 6185);

INSERT INTO Students (studentID, medicareNumber)

VALUES

(2001, 6349),

(2002, 6560),

(2003, 6640),

(2004, 6661),

(2005, 6706),

(2006, 6736),

(2007, 6760),

(2008, 7633),

(2009, 8179),

(2010, 8280),

(2011, 8456),

(2012, 9388),

(2013, 9508),

(2014, 9620),

(2015, 9671),

(2016, 9801),

(2017, 9830);

INSERT INTO Infections (infectionNumber, medicareNumber, date, type)

VALUES

(1, 9830, '2022-12-17', 'Other'),

(1, 9801, '2023-01-28', 'SARS-Cov-2 Variant'),

(1, 5400, '2021-11-08', 'Other'),

(1, 5860, '2023-03-26', 'Other'),

(3, 4874, '2023-07-14', 'COVID-19'),

(1, 6349, '2023-03-09', 'COVID-19'),

(1, 1947, '2023-01-04', 'COVID-19'),

(1, 4233, '2022-11-17', 'COVID-19'),

(1, 6706, '2022-03-22', 'COVID-19'),

(1, 6736, '2022-08-15', 'SARS-Cov-2 Variant'),

(3, 6011, '2023-06-24', 'Other'),

(2, 8179, '2023-01-03', 'SARS-Cov-2 Variant'),

(1, 4874, '2022-05-03', 'COVID-19'),

(2, 1947, '2023-05-26', 'SARS-Cov-2 Variant'),

(1, 8179, '2022-08-22', 'SARS-Cov-2 Variant'),

(2, 1799, '2023-05-26', 'Other'),

(1, 8280, '2022-06-19', 'SARS-Cov-2 Variant'),  
(1, 9508, '2021-03-04', 'COVID-19'),  
(2, 2506, '2023-06-10', 'SARS-Cov-2 Variant'),  
(2, 2515, '2023-01-22', 'COVID-19'),  
(1, 2506, '2022-08-14', 'SARS-Cov-2 Variant'),  
(1, 1805, '2022-05-21', 'SARS-Cov-2 Variant'),  
(3, 8280, '2023-07-17', 'SARS-Cov-2 Variant'),  
(1, 3295, '2023-01-21', 'SARS-Cov-2 Variant'),  
(1, 4628, '2023-02-17', 'SARS-Cov-2 Variant'),  
(1, 6011, '2022-01-13', 'Other'),  
(2, 4874, '2023-06-19', 'COVID-19'),  
(1, 3667, '2023-03-02', 'SARS-Cov-2 Variant'),  
(1, 9671, '2022-08-05', 'Other'),  
(2, 6736, '2023-03-06', 'COVID-19'),  
(1, 1556, '2022-07-09', 'COVID-19'),  
(2, 1556, '2023-01-01', 'COVID-19'),  
(3, 1556, '2023-03-09', 'COVID-19'),  
(2, 9508, '2022-01-27', 'Other'),  
(1, 6013, '2022-01-01', 'SARS-Cov-2 Variant'),  
(1, 8456, '2023-03-19', 'SARS-Cov-2 Variant'),  
(1, 1799, '2023-08-01', 'COVID-19'),  
(3, 9508, '2023-01-12', 'Other'),  
(1, 6640, '2022-01-24', 'SARS-Cov-2 Variant'),  
(2, 6013, '2023-03-07', 'COVID-19'),

(1, 5728, '2023-04-13', 'Other'),  
(1, 2515, '2022-11-24', 'SARS-Cov-2 Variant'),  
(1, 3792, '2022-11-01', 'Other'),  
(1, 9620, '2023-07-13', 'SARS-Cov-2 Variant'),  
(1, 7633, '2022-09-02', 'Other'),  
(1, 1132, '2023-04-13', 'Other'),  
(2, 6011, '2023-05-17', 'COVID-19'),  
(2, 9671, '2022-12-22', 'COVID-19'),  
(2, 6640, '2023-03-28', 'Other'),  
(2, 5400, '2023-05-21', 'COVID-19'),  
(2, 8280, '2023-02-03', 'Other'),  
(2, 4233, '2023-01-09', 'SARS-Cov-2 Variant'),  
(2, 4628, '2023-06-17', 'Other'),  
(1, 5721, '2022-02-05', 'COVID-19'),  
(2, 5721, '2023-06-24', 'Other'),  
(1, 3837, '2022-02-18', 'SARS-Cov-2 Variant'),  
(2, 1132, '2023-08-01', 'COVID-19'),  
(2, 3792, '2023-01-09', 'COVID-19'),  
(3, 5721, '2023-08-02', 'Other'),  
(3, 9671, '2023-07-25', 'COVID-19');

INSERT INTO Vaccinations (doseNumber, medicareNumber, date, type)

VALUES

(1, 6013, '2021-10-14', 'AstraZeneca'),

(1, 3667, '2022-02-18', 'Sinovac'),  
(1, 3682, '2021-04-22', 'Novavax'),  
(1, 9671, '2022-08-05', 'Novavax'),  
(1, 1556, '2023-07-13', 'Sinovac'),  
(1, 9830, '2022-10-28', 'AstraZeneca'),  
(1, 6560, '2022-05-22', 'Novavax'),  
(2, 6560, '2023-01-05', 'Johnson & Johnson'),  
(1, 4628, '2023-05-18', 'Sinovac'),  
(1, 8456, '2022-03-26', 'Pfizer'),  
(1, 9801, '2022-04-01', 'Sinovac'),  
(1, 6011, '2022-02-12', 'Johnson & Johnson'),  
(1, 6736, '2022-11-03', 'Moderna'),  
(1, 6640, '2022-07-11', 'Johnson & Johnson'),  
(1, 6661, '2022-09-17', 'Novavax'),  
(1, 2515, '2021-09-13', 'Johnson & Johnson'),  
(1, 9508, '2022-02-26', 'Johnson & Johnson'),  
(1, 2941, '2021-01-03', 'Novavax'),  
(1, 2068, '2022-09-03', 'Novavax'),  
(2, 6640, '2023-07-24', 'Pfizer'),  
(1, 7633, '2022-01-07', 'Johnson & Johnson'),  
(1, 3837, '2022-10-17', 'AstraZeneca'),  
(1, 1805, '2022-10-08', 'Moderna'),  
(1, 6760, '2021-02-03', 'Novavax'),  
(1, 5048, '2022-11-03', 'Novavax'),

(2, 5728, '2022-10-02', 'Moderna'),  
(1, 3295, '2021-03-12', 'Moderna'),  
(1, 6349, '2021-12-03', 'Johnson & Johnson'),  
(2, 9830, '2022-08-26', 'AstraZeneca'),  
(1, 5721, '2022-09-27', 'Moderna'),  
(2, 8179, '2023-02-26', 'AstraZeneca'),  
(3, 6560, '2023-03-22', 'Moderna'),  
(2, 6011, '2023-01-25', 'Moderna'),  
(2, 2515, '2022-07-07', 'Novavax'),  
(2, 2068, '2023-05-06', 'Moderna'),  
(3, 6011, '2022-08-28', 'Sinovac'),  
(2, 9671, '2022-05-03', 'AstraZeneca'),  
(2, 3837, '2022-11-10', 'Moderna'),  
(2, 3295, '2023-07-24', 'Sinovac'),  
(3, 3837, '2022-11-21', 'Moderna'),  
(2, 6013, '2023-07-18', 'Pfizer'),  
(2, 5905, '2022-11-22', 'Novavax'),  
(1, 2506, '2022-11-10', 'Sinovac'),  
(4, 3837, '2023-02-26', 'Sinovac'),  
(2, 6760, '2023-02-10', 'Novavax'),  
(1, 6185, '2022-04-03', 'Johnson & Johnson'),  
(2, 9801, '2023-06-28', 'AstraZeneca'),  
(1, 5905, '2022-01-17', 'Sinovac'),  
(1, 8280, '2022-04-02', 'Pfizer'),

(1, 4874, '2023-02-27', 'Moderna'),  
(1, 8179, '2023-08-02', 'Johnson & Johnson'),  
(1, 5728, '2022-09-24', 'Moderna'),  
(1, 4997, '2022-10-09', 'Sinovac'),  
(2, 3682, '2023-07-17', 'Novavax'),  
(2, 6349, '2023-07-09', 'Sinovac'),  
(2, 5048, '2023-07-13', 'Sinovac'),  
(2, 3667, '2023-05-21', 'Johnson & Johnson'),  
(1, 1799, '2023-07-15', 'Johnson & Johnson'),  
(1, 4233, '2022-03-02', 'AstraZeneca'),  
(3, 6013, '2023-06-15', 'Pfizer');

INSERT INTO Ministries (ministryID, sector)

VALUES

(1, 'Agriculture'),  
(2, 'Health'),  
(3, 'Energy');

INSERT INTO Facilities (facilityID, name, phone, webAddress, capacity)

VALUES

(1, 'Base', 930727010, 'www.facility1.com', 152),  
(2, 'Head', 542731214, 'www.facility2.com', 108),  
(3, 'Knox', 699068911, 'www.facility3.com', 185),  
(4, 'Jackson', 306677544, 'www.facility4.com', 462),

```
(5, 'Levi', 510938734, 'www.facility5.com', 72),  
(6, 'Home', 456290238, 'www.facility6.com', 38),  
(7, 'Tree', 497364493, 'www.facility7.com', 49),  
(8, 'Sea', 585972116, 'www.facility8.com', 310),  
(9, 'Ship', 513477902, 'www.facility9.com', 163);
```

INSERT INTO Operation (ministryID, facilityID)

VALUES

```
(1, 1),  
(1, 4),  
(1, 5),  
(2, 2),  
(2, 6),  
(2, 7),  
(3, 3),  
(3, 8),  
(3, 9);
```

INSERT INTO ManagementFacility (managementFacilityID, facilityID, president)

VALUES

```
(1, 4, 1004),  
(2, 5, 1005),  
(3, 6, 1006),  
(4, 7, 1007),
```

(5, 8, 1008),

(6, 9, 1009);

INSERT INTO GeneralManagementFacility (generalManagementFacilityID, managementFacilityID)

VALUES

(1, 4),

(2, 5),

(3, 6);

INSERT INTO HeadOffice (headOfficeID, managementFacilityID, ministryID)

VALUES

(1, 1, 1),

(2, 5, 2),

(3, 3, 3);

INSERT INTO EducationalFacility (educationalFacilityID, facilityID, schoolType)

VALUES

(1, 1, 'Primary'),

(2, 2, 'Middle'),

(3, 3, 'High');

INSERT INTO Teachers (teacherID, employeeID, educationalFacilityID)

VALUES

(1, 1001, 1),

(2, 1002, 2),

(3, 1003, 3);

INSERT INTO ElementaryTeacher (elementaryTeacherID, teacherID)

VALUES

(1, 1);

INSERT INTO SecondaryTeacher (secondaryTeacherID, teacherID, specialization, role)

VALUES

(2, 2, 'Math', 'Counselor'),

(3, 3, 'Geography', 'Director');

INSERT INTO Registration (studentID, facilityID, startDate, endDate, level)

VALUES

(2001, 1, '2023-11-23', '2024-12-15', 5),

(2002, 1, '2022-07-16', NULL, 12),

(2003, 3, '2022-12-28', NULL, 6),

(2004, 1, '2023-05-14', NULL, 10),

(2005, 3, '2022-10-06', '2024-01-27', 2),

(2006, 3, '2022-01-17', '2024-04-23', 8),

(2007, 1, '2022-02-14', '2023-05-01', 6),

(2008, 1, '2022-04-08', '2024-09-15', 9),

(2009, 2, '2022-08-21', '2023-05-05', 5),

(2010, 2, '2023-08-06', '2023-07-24', 2),  
(2011, 2, '2023-09-09', '2023-05-16', 12),  
(2012, 3, '2023-03-17', '2023-04-23', 1),  
(2013, 2, '2023-06-03', '2024-07-11', 8),  
(2014, 1, '2022-11-03', '2023-08-01', 1),  
(2015, 2, '2022-11-21', '2024-06-06', 4),  
(2016, 1, '2022-04-05', NULL, 1),  
(2017, 1, '2022-11-10', '2024-05-05', 12);

INSERT INTO Location (addressID, address, city, province, postalCode)

VALUES

(1, 301, 'Montreal', 'Quebec', 'H0KG0L'),  
(2, 659, 'Laval', 'Quebec', 'H2KG3L'),  
(3, 772, 'Laval', 'Quebec', 'H4KG6L'),  
(4, 374, 'Toronto', 'Ontario', 'H5KG2L'),  
(5, 659, 'Toronto', 'Ontario', 'H3KG4L'),  
(6, 611, 'Montreal', 'Quebec', 'H1KG3L'),  
(7, 124, 'Montreal', 'Quebec', 'H8KG0L'),  
(8, 973, 'Toronto', 'Ontario', 'H4KG8L'),  
(9, 228, 'Toronto', 'Ontario', 'H7KG5L'),  
(10, 842, 'Montreal', 'Quebec', 'H6KG8L'),  
(11, 717, 'Laval', 'Quebec', 'H8KG0L'),  
(12, 852, 'Ottawa', 'Ontario', 'H4KG4L'),  
(13, 462, 'Montreal', 'Quebec', 'H2KG2L'),  
(14, 753, 'Montreal', 'Quebec', 'H5KG6L'),  
(15, 369, 'Montreal', 'Quebec', 'H1KG7L'),  
(16, 959, 'Montreal', 'Quebec', 'H1KG9L'),  
(17, 411, 'Laval', 'Quebec', 'H8KG3L'),  
(18, 587, 'Toronto', 'Ontario', 'H1KG1L'),  
(19, 228, 'Toronto', 'Ontario', 'H4KG5L'),  
(20, 395, 'Toronto', 'Ontario', 'H5KG0L'),  
(21, 920, 'Montreal', 'Quebec', 'H7KG2L'),

(22, 299, 'Laval', 'Quebec', 'H3KG3L'),  
(23, 588, 'Laval', 'Quebec', 'H1KG4L'),  
(24, 284, 'Toronto', 'Ontario', 'H2KG4L'),  
(25, 736, 'Laval', 'Quebec', 'H4KG6L'),  
(26, 585, 'Montreal', 'Quebec', 'H9KG8L'),  
(27, 825, 'Ottawa', 'Ontario', 'H8KG3L'),  
(28, 341, 'Montreal', 'Quebec', 'H8KG6L'),  
(29, 930, 'Montreal', 'Quebec', 'H1KG1L'),  
(30, 233, 'Montreal', 'Quebec', 'H5KG8L'),  
(31, 106, 'Ottawa', 'Ontario', 'H3KG9L'),  
(32, 228, 'Ottawa', 'Ontario', 'H4KG8L'),  
(33, 514, 'Laval', 'Quebec', 'H0KG4L'),  
(34, 246, 'Ottawa', 'Ontario', 'H3KG7L'),  
(35, 707, 'Montreal', 'Quebec', 'H7KG1L'),  
(36, 904, 'Montreal', 'Quebec', 'H9KG8L'),  
(37, 327, 'Laval', 'Quebec', 'H3KG5L'),  
(38, 905, 'Laval', 'Quebec', 'H8KG7L'),  
(39, 876, 'Ottawa', 'Ontario', 'H9KG7L'),  
(40, 673, 'Laval', 'Quebec', 'H4KG3L'),  
(41, 970, 'Montreal', 'Quebec', 'H3KG8L'),  
(42, 943, 'Montreal', 'Quebec', 'H9KG5L'),  
(43, 829, 'Ottawa', 'Ontario', 'H7KG4L'),  
(44, 349, 'Montreal', 'Quebec', 'H0KG7L'),  
(45, 452, 'Montreal', 'Quebec', 'H6KG5L'),  
(46, 462, 'Laval', 'Quebec', 'H3K9L4'),  
(47, 714, 'Laval', 'Quebec', 'H2K5L1'),  
(48, 937, 'Toronto', 'Ontario', 'H2K8L7'),  
(49, 118, 'Laval', 'Quebec', 'H1K5L1'),  
(50, 253, 'Laval', 'Quebec', 'H1K3L1'),  
(51, 424, 'Montreal', 'Quebec', 'H9K9L2'),  
(52, 429, 'Toronto', 'Ontario', 'H0K0L2'),  
(53, 391, 'Montreal', 'Quebec', 'H5K5L9'),  
(54, 216, 'Ottawa', 'Ontario', 'H7K6L9');

INSERT INTO Situated (facilityID, addressID)

VALUES

(1, 46),

(2, 47),

(3, 48),

(4, 49),

(5, 50),

(6, 51),

(7, 52),

(8, 53),

(9, 54);

INSERT INTO Residence (medicareNumber, addressID)

VALUES

(5048, 1),

(7633, 2),

(3792, 3),

(3295, 4),

(9508, 5),

(1805, 6),

(6349, 7),

(2068, 8),

(6013, 9),

(9671, 10),

(5721, 11),

(2941, 12),

(6736, 13),

(6185, 14),

(6760, 15),

(3837, 16),

(4997, 17),

(2165, 18),

(6560, 19),

(6661, 20),

(4874, 21),

(4233, 22),

(2506, 23),

(2515, 24),

(4628, 25),  
(9388, 26),  
(6706, 27),  
(9830, 28),  
(1799, 29),  
(3667, 30),  
(9801, 31),  
(1132, 32),  
(1556, 33),  
(5728, 34),  
(3682, 35),  
(5860, 36),  
(6640, 37),  
(8456, 38),  
(8179, 39),  
(1947, 40),  
(9620, 41),  
(5400, 42),  
(6011, 43),  
(8280, 44),  
(5905, 45);

INSERT INTO Contract (employeeID, facilityID, startDate, endDate, role)

VALUES

(1001, 1, '2023-01-01', '2023-03-01', 'Teacher'),  
(1001, 2, '2023-01-03', NULL, 'Teacher'),  
(1001, 3, '2023-04-03', NULL, 'Teacher'),  
(1002, 2, '2023-01-01', '2023-03-01', 'Teacher'),  
(1002, 3, '2023-03-02', NULL, 'Teacher'),  
(1002, 1, '2023-06-02', NULL, 'Teacher'),  
(1003, 3, '2023-01-03', NULL, 'Teacher'),  
(1004, 4, '2023-01-01', '2023-03-01', 'President'),  
(1005, 5, '2023-01-01', '2023-03-01', 'President'),

(1006, 6, '2023-01-01', '2023-03-01', 'President'),  
(1007, 7, '2023-01-01', '2023-03-01', 'President'),  
(1008, 8, '2023-01-01', '2023-03-01', 'President'),  
(1009, 9, '2023-01-01', '2023-03-01', 'President'),  
(1010, 1, '2023-01-01', '2023-03-01', 'Principal'),  
(1011, 2, '2023-01-01', '2023-03-01', 'Principal'),  
(1012, 3, '2023-01-01', '2023-03-01', 'Principal'),  
(1013, 4, '2023-01-01', NULL, 'Secretary'),  
(1014, 4, '2023-01-02', '2023-03-01', 'Secretary'),  
(1015, 5, '2023-01-03', '2023-03-01', 'Secretary'),  
(1016, 5, '2023-01-04', '2023-03-01', 'Secretary'),  
(1017, 6, '2023-01-05', NULL, 'Secretary'),  
(1018, 7, '2023-01-01', NULL, 'Secretary'),  
(1019, 8, '2023-01-02', NULL, 'Secretary'),  
(1020, 9, '2023-01-03', '2023-03-01', 'Secretary'),  
(1021, 4, '2023-01-01', '2023-03-01', 'Security'),  
(1022, 5, '2023-01-02', '2023-03-01', 'Security'),  
(1023, 6, '2023-01-03', '2023-03-01', 'Security'),  
(1024, 7, '2023-01-04', '2023-03-01', 'Security'),  
(1025, 8, '2023-01-05', '2023-03-01', 'Security'),  
(1026, 9, '2023-01-06', '2023-03-01', 'Security'),  
(1027, 9, '2023-01-01', '2023-03-01', 'other'),  
(1028, 9, '2023-01-02', '2023-03-01', 'other');

```
INSERT INTO Schedule (employeeID,facilityID,scheduleDate , startTime ,  
endTime )  
VALUES  
(1013,4,'2023-08-11','10:00:00','12:00:00'),  
(1017,6,'2023-08-20','8:00:00','11:00:00'),  
(1002,1,'2023-08-10','11:00:00','13:00:00'),  
(1002,3,'2023-08-13','08:00:00','09:00:00'),  
(1002,3,'2023-08-14','11:00:00','13:00:00');
```

### **Normalization: Analysis of 3NF and BCNF**

Normalization is a database design technique that minimizes data redundancy and dependency by organizing fields and tables of a database. The main aim of normalization is to add, delete, and modify so that one can have consistent data across all the tables.

#### **Third Normal Form (3NF):**

A table is in 3NF if:

1:It is in 2NF.

2:All the attributes in a table are functionally dependent only on the primary key.

#### **Boyce-Codd Normal Form (BCNF):**

A table is in BCNF if:

1: It is in 3NF.

2: For any non-trivial functional dependency  $X \rightarrow Y$ , X is a superkey. This means that the left-hand side of every non-trivial functional dependency must be a superkey.

In the DDL:

1: The introduction of the '**Location**' table decouples address details from other entities. This ensures attributes like city, province, and postal code aren't intertwined with non-key attributes in other entities. By referencing '**addressID**' in tables such as '**Residence**' and '**Situated**', we maintain 3NF.

2: Tables like '**Infections**', '**Teachers**', '**EmailLog**', and others have attributes that directly hinge on the primary key, and not on any other attribute. This design is in line with 3NF, where every non-prime attribute is non-transitively dependent on the primary key.

3: The design doesn't exhibit any non-trivial functional dependency where a non-superkey is determining another attribute. This aligns with BCNF principles.

The DDL we've designed embodies normalization principles, adhering closely to 3NF and BCNF standards. We've instituted a dedicated '**Location**' table to encapsulate address details, ensuring city, province, and postal code remain distinct from non-key attributes in other tables. Each table, from '**Infections**' to '**Teachers**', is structured so that attributes pivot solely on the primary key, reinforcing our commitment to 3NF. Furthermore, our design respects BCNF, evident in the absence of non-superkeys determining other attributes. This approach bolsters data integrity while minimizing redundancy.

## **Procedure(s) used**

### **– Email system**

SendScheduleEmails() can be called on any day and will send out emails to each employee of every facility outlining their schedule from the current day to the next Sunday. The sent emails are logged into the emailLog table.

For every employee working in every facility, the system sends an email to every employee indicating the schedule of the employee in the facility for the coming week. The subject of the email includes the facility name, and the dates covered by the schedule. The email body includes the facility name, the address of the facility, the employee's first-name, last-name, email address, and details of the schedule for the coming week. Details include the day of the week, start time and end time. The body of the message should includes an entry for every day of the week followed by the starting hour and end hour for that day. A message "No Assignment" is displayed if the employee is not scheduled for that specific entry.

DELIMITER //

```
CREATE PROCEDURE SendScheduleEmails()
```

```
BEGIN
```

```
-- table attributes
```

```
DECLARE fetchedScheduleDate DATE;
```

```
DECLARE fetchedFacilityName VARCHAR(255);
```

```
DECLARE fetchedFacilityAddress INT;
```

```
DECLARE fetchedFacilityID INT;
```

```
DECLARE fetchedEmployeeID INT;
```

```
DECLARE fetchedFirstName VARCHAR(255);
```

```
DECLARE fetchedLastName VARCHAR(255);
```

```
DECLARE fetchedEmail VARCHAR(255);
```

```
DECLARE scheduleSubject VARCHAR(255);
```

```
DECLARE scheduleBody TEXT;
DECLARE first80char VARCHAR(80);

DECLARE emailBody TEXT;

-- store in memory attributes

DECLARE prevScheduleDate DATE;
DECLARE prevFacilityID INT DEFAULT NULL;
DECLARE prevEmployeeID INT DEFAULT NULL;
DECLARE prevFacilityName VARCHAR(255);
DECLARE prevFacilityAddress INT;
DECLARE prevFirstName VARCHAR(255);
DECLARE prevLastName VARCHAR(255);
DECLARE prevEmail VARCHAR(255);

-- dates

DECLARE currentDate DATE;
DECLARE nextSunday DATE;

DECLARE dayOfWeek VARCHAR(255);
DECLARE scheduleDayName VARCHAR(255);
```

-- flags

DECLARE done1 INT DEFAULT 0;

-- cursor for table to check for facilityID and employeeID

DECLARE scheduleCursor CURSOR FOR

-- Your query to retrieve necessary schedule data

SELECT

S.scheduleDate,

F.facilityID,

F.name AS facilityName,

L.address AS facilityAddress,

E.employeeID,

P.firstName,

P.lastName,

P.email

FROM Schedule S

LEFT JOIN Contract C ON S.employeeID = C.employeeID

LEFT JOIN Employees E ON S.employeeID = E.employeeID

LEFT JOIN Facilities F ON C.facilityID = F.facilityID

LEFT JOIN Situated SI ON F.facilityID = SI.facilityID

LEFT JOIN Location L ON SI.addressID = L.addressID

LEFT JOIN Persons P ON E.medicareNumber = P.medicareNumber

```
        WHERE S.scheduleDate <= DATE_ADD(CURDATE(), INTERVAL (7 - DAYOFWEEK(CURDATE())) + 1 DAY) AND endDate IS NULL
```

```
        ORDER BY F.facilityID ASC, S.employeeID ASC, S.scheduleDate ASC, S.startTime ASC, S.endTime ASC;
```

```
-- scheduleLoop cursor checker
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done1 = 1;
```

```
-- Find closest next Sunday
```

```
SET currentDate = CURDATE();
```

```
SET nextSunday = DATE_ADD(currentDate, INTERVAL (7 - DAYOFWEEK(currentDate)) + 1 DAY);
```

```
-- initialize emailBody
```

```
SET emailBody = "";
```

```
-- Open the cursor
```

```
OPEN scheduleCursor;
```

```
scheduleLoop: LOOP
```

```
-- get data row by row
```

```
        FETCH scheduleCursor INTO fetchedScheduleDate, fetchedFacilityID, fetchedFirstName,  
fetchedFacilityName, fetchedFacilityAddress, fetchedEmployeeID, fetchedLastName, fetchedEmail;
```

-- checks end of cursor

IF done1 THEN

LEAVE scheduleLoop; -- Exit the loop

END IF;

-- Checks first table entry or if there's a change in facilityID or employeeID

IF prevFacilityID IS NULL OR prevFacilityID <> fetchedFacilityID OR prevEmployeeID <> fetchedEmployeeID THEN

IF prevFacilityID IS NOT NULL OR prevEmployeeID IS NOT NULL THEN

-- Send the email for the previous facility and employee pair

SET first80char = LEFT(scheduleBody, 80);

INSERT INTO EmailLog (date, facilityID, employeeID, sender, receiverFirstName, receiverLastName, subject, body) VALUES (currentDate, prevFacilityID, prevEmployeeID, prevFacilityName, prevFirstName, prevLastName, scheduleSubject, first80char);

SET scheduleSubject = ";

SET scheduleBody = ";

SET currentDate = CURDATE();

END IF;

-- Compose new email subject

SET scheduleSubject = CONCAT(fetchedFacilityName, ' Schedule for ', DATE\_FORMAT(currentDate, '%d-%b-%Y'), ' to ', DATE\_FORMAT(nextSunday, '%d-%b-%Y'));

-- Compose new email body

SET scheduleBody = CONCAT(

'Facility: ', fetchedFacilityName, '\n',

```

'Address: ', fetchedFacilityAddress, '\n',
'Employee: ', fetchedFirstName, ' ', fetchedLastName, ' (', fetchedEmail, ')'\n\n'
);

SET prevScheduleDate = fetchedScheduleDate;
SET prevFacilityID = fetchedFacilityID;
SET prevEmployeeID = fetchedEmployeeID;
SET prevFacilityName = fetchedFacilityName;
SET prevFacilityAddress = fetchedFacilityAddress;
SET prevFirstName = fetchedFirstName;
SET prevLastName = fetchedLastName;
SET prevEmail = fetchedEmail;

-- call to emailGenerator that passes parameters facilityID and employeeID and assign the
returned text
CALL emailBodyGenerator (fetchedFacilityID, fetchedEmployeeID, emailBody);
SET scheduleBody = CONCAT(scheduleBody, ' ', emailBody);

END IF;

END LOOP scheduleLoop;

-- Outputs last email
SET first80char = LEFT(scheduleBody, 80);

INSERT INTO EmailLog (date, facilityID, employeeID, sender, receiverFirstName,
receiverLastName, subject, body) VALUES (currentDate, prevFacilityID, prevEmployeeID,

```

```
prevFacilityName, prevFirstName, prevLastName, scheduleSubject, first80char);
```

```
CLOSE scheduleCursor;
```

```
END
```

```
//
```

```
DELIMITER ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE emailBodyGenerator (
```

```
    IN newFacilityID INT,
```

```
    IN newEmployeeID INT,
```

```
    OUT emailBody TEXT
```

```
)
```

```
BEGIN
```

```
    DECLARE dayOfWeek VARCHAR(255);
```

```
    DECLARE scheduleDateDayOfWeek VARCHAR(255);
```

```
-- dates and times
```

```
    DECLARE fetchedScheduleDate DATE;
```

```
    DECLARE fetchedStartTime TIME;
```

```
    DECLARE fetchedEndTime TIME;
```

```
    DECLARE currentDate DATE;
```

```
    DECLARE nextSunday DATE;
```

```
-- save last rows

DECLARE prevScheduleDate DATE;

DECLARE prevStartTime TIME;

DECLARE prevEndTime TIME;

-- flags

DECLARE done1 INT DEFAULT 0;

DECLARE timeExists INT DEFAULT 0;

-- cursor for table to check for facilityID and employeeID

DECLARE dayScheduleCursor CURSOR FOR

-- Your query to retrieve necessary schedule data

SELECT

    S.scheduleDate,

    S.startTime,

    S.endTime

FROM Schedule S

LEFT JOIN Contract C ON S.employeeID = C.employeeID

    LEFT JOIN Employees E ON S.employeeID = E.employeeID

    LEFT JOIN Facilities F ON C.facilityID = F.facilityID

    LEFT JOIN Situated SI ON F.facilityID = SI.facilityID

    LEFT JOIN Location L ON SI.addressID = L.addressID

    LEFT JOIN Persons P ON E.medicareNumber = P.medicareNumber
```

```
        WHERE F.facilityID = newFacilityID AND S.employeeID = newEmployeeID AND
S.scheduleDate <= DATE_ADD(CURDATE(), INTERVAL (7 - DAYOFWEEK(CURDATE())) + 1 DAY) AND
endDate IS NULL
```

```
        ORDER BY F.facilityID ASC, S.employeeID ASC, S.scheduleDate ASC, S.startTime ASC,
S.endTime ASC;
```

```
-- scheduleLoop cursor checker
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done1 = 1;
```

```
-- set dates
```

```
SET currentDate = CURDATE();
```

```
SET nextSunday = DATE_ADD(currentDate, INTERVAL (7 - DAYOFWEEK(currentDate)) + 1 DAY);
```

```
-- initialize email body for new day
```

```
SET emailBody = ";
```

```
-- Open the cursor
```

```
OPEN dayScheduleCursor;
```

```
dayScheduleLoop: LOOP
```

```
-- get data row by row
```

```
    FETCH dayScheduleCursor INTO fetchedScheduleDate, fetchedStartTime,
fetchedEndTime;
```

```
-- checks end of cursor
```

```

IF done1 THEN

    LEAVE dayScheduleLoop; -- Exit the loop

    END IF;

-- initialize/reassign dayOfWeek and scheduleDateDayOfWeek

    SET dayOfWeek = DAYNAME(currentDate);

    SET scheduleDateDayOfWeek = DAYNAME(fetchedScheduleDate);

-- iterates from today up until nextSunday

    WHILE currentDate <= nextSunday DO

        -- outputs 'Day: ' into body

        SET emailBody = CONCAT(emailBody, dayOfWeek, ': ');

        -- reset time exists checker

        SET timeExists = 0;

-- check if currentDate matches fetchedScheduleDate and output all shifts

        INTRADAY: WHILE scheduleDateDayOfWeek = dayOfWeek DO

            SET emailBody = CONCAT(emailBody,
TIME_FORMAT(fetchedStartTime, '%H:%i'), ' - ', TIME_FORMAT(fetchedEndTime, '%H:%i'), ' ');

            SET timeExists = 1;

            FETCH dayScheduleCursor INTO fetchedScheduleDate, fetchedStartTime,
fetchedEndTime;

            IF done1 THEN

                LEAVE INTRADAY; -- Exit the loop

```

```
        END IF;

        SET scheduleDateDayOfWeek = DAYNAME(fetchedScheduleDate);

        END WHILE;

-- if no time shifts found during that given day

IF timeExists = 0 THEN

    SET emailBody = CONCAT(emailBody, 'No Assignment \n');

END IF;

-- increment currentDate by 1

SET currentDate = currentDate + INTERVAL 1 DAY;

-- rename dayOfWeek

SET dayOfWeek = DAYNAME(currentDate);

END WHILE; -- end of day loop

END LOOP dayScheduleLoop;

CLOSE dayScheduleCursor;

END;

//



DELIMITER ;
```

	emailID	date	facilityID	employeeID	sender	receiverFirstName	receiverLastName	subject	body
▶	1	2023-08-10	1	1002	Base	Mary	Rogers	Base Schedule for 10-Aug-2023 to 13-Aug-2023	Facility: Base Address: 462 Employee: Mary Rogers (person33@example.com) Thurs
	2	2023-08-10	3	1002	Knox	Mary	Rogers	Knox Schedule for 10-Aug-2023 to 13-Aug-2023	Facility: Knox Address: 937 Employee: Mary Rogers (person33@example.com) Thurs
	3	2023-08-10	4	1013	Jackson	Nevaeh	Johnson	Jackson Schedule for 10-Aug-2023 to 13-Aug-2023	Facility: Jackson Address: 118 Employee: Nevaeh Johnson (person35@example.com)

## Trigger(s) used

### 1) CheckFutureSchedule

A schedule of four weeks ahead of time is supported by the system.

-- trigger 1) A schedule of four weeks ahead of time is supported by the system.

DELIMITER //

CREATE TRIGGER CheckFutureSchedule

BEFORE INSERT ON Schedule

FOR EACH ROW

BEGIN

DECLARE maxAllowedDate DATE;

DECLARE currentDate DATE;

SET maxAllowedDate = DATE\_ADD(CURDATE(), INTERVAL 4 WEEK);

SET currentDate = NEW.scheduleDate;

IF currentDate > maxAllowedDate THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Scheduling more than four weeks ahead is not allowed';

```

END IF;

IF currentDate < CURDATE() THEN

    SIGNAL SQLSTATE '45000'

    SET MESSAGE_TEXT = 'Scheduling before current date is not allowed';

END IF;

END;

//



DELIMITER ;

```

704  
705 -- A schedule of four weeks ahead of time is supported by the system.  
706  
707 • select curdate();  
708  
709 • Ⓜ insert into Schedule (  
710 employeeID,  
711 facilityID ,  
712 scheduleDate ,  
713 startTime ,  
714 endTime )  
715 values (1017,2,'2023-10-01','10:00:00','12:00:00');  
716



	curdate()
1	2023-08-07

Result 31 ×

Output:

Action Output

#	Time	Action	Message
1	19:46:34	insert into Schedule ( employeeID, facilityID , scheduleDate , sta...	Error Code: 1644. Scheduling more than four weeks ahead is not allowed

## 2) CheckEmployeeConflictingSchedule

An employee cannot be scheduled at two different conflicting times neither at the same facility nor at different facilities

-- trigger 2) An employee cannot be scheduled at two different  
-- conflicting times neither at the same facility nor at different facilities

DELIMITER //

CREATE TRIGGER CheckEmployeeConflictingSchedule

BEFORE INSERT ON Schedule

FOR EACH ROW

BEGIN

DECLARE conflictingCount INT;

SET conflictingCount = (

SELECT COUNT(\*)

FROM Schedule

WHERE employeeID = NEW.employeeID

AND scheduleDate = NEW.scheduleDate

AND ((startTime <= NEW.startTime AND endTime > NEW.startTime)

OR (startTime < NEW.endTime AND endTime >= NEW.endTime)

OR (NEW.startTime <= startTime AND NEW.endTime > startTime)

OR (NEW.startTime < endTime AND NEW.endTime >= endTime)

)

);

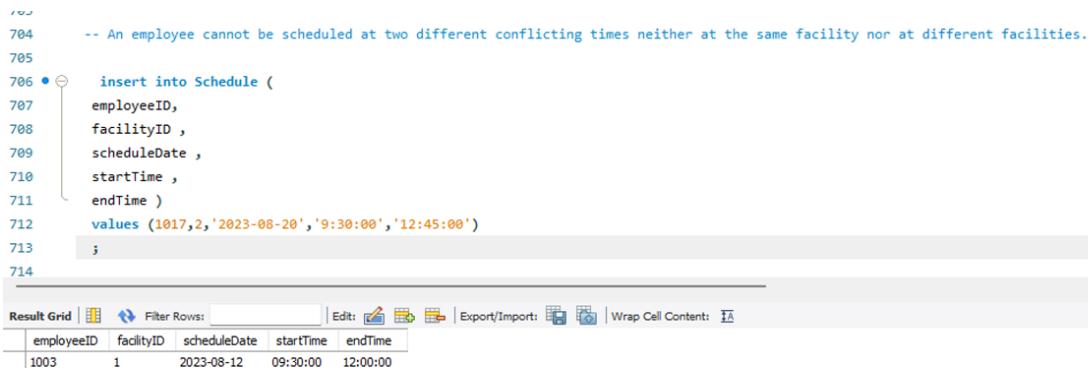
```

IF conflictingCount > 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Conflicting schedule for the employee';
END IF;

END;
//  

DELIMITER ;

```



The screenshot shows a MySQL Workbench interface. In the top-left pane, there is a code editor with the following SQL code:

```

-- An employee cannot be scheduled at two different conflicting times neither at the same facility nor at different facilities.
insert into Schedule (
    employeeID,
    facilityID ,
    scheduleDate ,
    startTime ,
    endTime )
values (1017,2,'2023-08-20','9:30:00','12:45:00')
;
```

In the bottom-left pane, there is a results grid titled "Result Grid" showing the following data:

employeeID	facilityID	scheduleDate	startTime	endTime
1003	1	2023-08-12	09:30:00	12:00:00
1013	1	2023-08-11	04:30:00	05:00:00
<b>1017</b>	<b>2</b>	<b>2023-08-20</b>	<b>08:00:00</b>	<b>11:00:00</b>
*	HULL	HULL	HULL	HULL

In the bottom-right pane, there is an "Output" window titled "Schedule 27" showing the following message:

```

Action Output
# Time Action
1 19:27:30 insert into Schedule ( employeeID, facilityID , scheduleDate , start... Error Code: 1644. Conflicting schedule for the employee

```

### 3) CheckOneHourBetweenSchedule

If an employee is scheduled for two different periods on the same day either at the same facility or at different facilities, then at least one hour should be the duration between the first schedule and the second one.

```
-- trigger 3) If an employee is scheduled for two different periods on the same day  
-- either at the same facility or at different facilities, then at least one hour should be the  
duration between the first schedule and the second one.
```

```
DELIMITER //
```

```
CREATE TRIGGER CheckOneHourBetweenSchedule
```

```
BEFORE INSERT ON Schedule
```

```
FOR EACH ROW
```

```
BEGIN
```

```
DECLARE count INT;
```

```
SET count = (SELECT COUNT(timeDiff) FROM  
(SELECT
```

```
TIMESTAMPDIFF(MINUTE, endTime, NEW.startTime) AS timeDiff
```

```
FROM Schedule s
```

```
WHERE employeeID = NEW.employeeID
```

```
AND scheduleDate = NEW.scheduleDate
```

```
AND startTime < new.startTime) AS T
```

```
WHERE timeDiff < 60);
```

```
SET count = count + (SELECT COUNT(timeDiff) FROM
```

```
(SELECT
```

```
TIMESTAMPDIFF(MINUTE, NEW.endTime,startTime) AS timeDiff  
FROM Schedule s  
WHERE employeeID = NEW.employeeID  
AND scheduleDate = NEW.scheduleDate  
AND new.startTime<startTime) AS T  
WHERE timeDiff <60);  
  
IF count > 0 THEN  
    SIGNAL SQLSTATE '45000'  
    SET MESSAGE_TEXT = 'At least one hour is needed between two periods';  
END IF;  
  
END;  
//  
DELIMITER ;
```

```

705
706    -- If an employee is scheduled for two different periods on the same day either at the same facility or at different facilities,
707    -- then at least one hour should be the duration between the first schedule and the second one.
708
709 • ↴ insert into Schedule (
710     employeeID,
711     facilityID,
712     scheduleDate,
713     startTime,
714     endTime
715     values (1017,2,'2023-08-20','11:30:00','13:30:00');
716
717

```

Result Grid | Filter Rows:  Edit: Export/Import: Wrap Cell Content:

	employeeID	facilityID	scheduleDate	startTime	endTime
▶	1003	1	2023-08-12	09:30:00	12:00:00
	1013	1	2023-08-11	04:30:00	05:00:00
●	1017	2	2023-08-20	08:00:00	11:00:00
*	NULL	NULL	NULL	NULL	NULL

Schedule 30 ×

Output Action Output # Time Action Message

1 19:36:27 insert into Schedule ( employeeID, facilityID, scheduleDate, sta... Error Code: 1644. At least one hour is needed between two periods

#### 4) CheckInfectionStatusBeforeScheduling

If a teacher is infected by COVID-19, then he/she cannot be scheduled to work for at least two weeks from the date of infection.

-- trigger 4) If a teacher is infected by COVID-19, then he/she cannot be scheduled to work for at least two weeks from the date of infection.

DELIMITER //

CREATE TRIGGER CheckInfectionStatusBeforeScheduling

BEFORE INSERT ON Schedule

FOR EACH ROW

BEGIN

DECLARE countConflictingInfection INT;

```
SET countConflictingInfection=
(SELECT COUNT(date)
     FROM Infections INNER JOIN Employees ON
Infections.medicareNumber=Employees.medicareNumber
INNER JOIN Teachers ON Teachers.employeeID=Employees.employeeID
WHERE Teachers.employeeID = NEW.employeeID
AND date + INTERVAL 2 WEEK >= NEW.scheduleDate );
```

```
IF countConflictingInfection > 0 THEN
  SIGNAL SQLSTATE '45000'
  SET MESSAGE_TEXT = 'Teacher cannot be scheduled to work for at least two weeks from the
date of infection';
END IF;
END;
//  
DELIMITER ;
```

```

690
691
692 -- If a teacher is infected by COVID-19, then he/she cannot be scheduled to work for at least two weeks from the date of infection.
693
694
695 • insert into Schedule (employeeID,facilityID ,scheduleDate ,startTime ,endTime )
696 values(1002,1,'2023-08-11','4:30:00','5:00:00');
697
698
699
700
701 -----

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

infectionNumber	type	dateOfInfection	medicareNumber	employeeID	teacherID
1	Other	2023-04-13	1132	1001	1
2	SARS-CoV-2 Variant	2023-06-12	1132	1001	1
1	SARS-CoV-2 Variant	2022-07-09	1556	1002	2
2	SARS-CoV-2 Variant	2023-08-07	1556	1002	2
1	SARS-CoV-2 Variant	2022-09-18	1799	1003	3

Result 18    Result 21 ×

Output :::::

Action Output

#	Time	Action	Message
1	19:05:56	insert into Schedule (employeeID,facilityID ,scheduleDate ,startTime ...	Error Code: 1644. Teacher cannot be scheduled to work for at least two weeks from the date of infection

## 5) CheckVaccinationStatusBeforeScheduling

An employee cannot be scheduled if she/he is not vaccinated, at least one vaccine for COVID-19 in the past six months prior to the date of the new schedule.

-- trigger 5) An employee cannot be scheduled if she/he is not vaccinated, at least one

-- vaccine for COVID-19 in the past six months prior to the date of the new schedule.

DELIMITER //

CREATE TRIGGER CheckVaccinationStatusBeforeScheduling

BEFORE INSERT ON Schedule

FOR EACH ROW

BEGIN

DECLARE countValidVaccines INT;

SET countValidVaccines=

```

(SELECT COUNT(*)

        FROM      Vaccinations     INNER     JOIN      Employees      ON
Vaccinations.medicareNumber=Employees.medicareNumber

        WHERE Employees.employeeID = NEW.employeeID

        AND date + INTERVAL 6 MONTH >= NEW.scheduleDate );

```

```

IF countValidVaccines = 0 THEN

    SIGNAL SQLSTATE '45000'

    SET MESSAGE_TEXT = 'An employee cannot be scheduled if she/he is not vaccinated in the
past six months prior the scheduled date';

END IF;

END;

//



DELIMITER ;

```

```

753
754 •   select * from Schedule;
755
756
757
758
759
760
761
762

```

employeeID	facilityID	scheduleDate	startTime	endTime
1003	1	2023-08-12	09:30:00	12:00:00
1013	1	2023-08-11	04:30:00	05:00:00
1017	2	2023-08-20	08:00:00	11:00:00
*				

```

Result 32  Result 41  Schedule 43 x
Output:
Action Output
#  Time  Action
1  21:23:59  select * from Schedule LIMIT 0, 1000
Message
3 row(s) returned

```

## 6) CancelAssignmentsOnInfection

If a teacher gets infected by COVID-19, then the system should automatically cancel all the assignments for the infected teacher for two weeks from the date of infection.

-- trigger 6) If a teacher gets infected by COVID-19, then the system should automatically cancel  
-- all the assignments for the infected teacher for two weeks from the date of infection.

DELIMITER //

CREATE TRIGGER CancelAssignmentsOnInfection

AFTER INSERT ON Infections

FOR EACH ROW

BEGIN

DELETE FROM Schedule WHERE scheduleDate IN

(SELECT scheduleDate

                  FROM       Infections      INNER      JOIN       Employees      ON  
Infections.medicareNumber=Employees.medicareNumber

                  INNER JOIN Teachers ON Teachers.employeeID=Employees.employeeID

                  INNER JOIN Schedule ON Schedule.employeeID=Teachers.employeeID

                  WHERE new.date + INTERVAL 2 WEEK >= scheduleDate );

END;

//

DELIMITER ;

- schedule table before insertion of infection

```

703
704      -- An employee cannot be scheduled if she/he is not vaccinated, at least one vaccine for COVID-19 in the past six months
705 • insert into Schedule (
706     employeeID,
707     facilityID ,
708     scheduleDate ,
709     startTime ,
710     endTime )
711     values (1016,1,'2023-08-11','4:30:00','5:00:00')
712 ;
713
714

```

The screenshot shows a database interface with a result grid and an output log.

**Result Grid:**

doseNumber	medicareNumber	date	type	employeeID	medicareNumber
1	4233	2022-03-02	AstraZeneca	1016	4233
1	4628	2023-10-18	Sinovac	1017	4628
1	4874	2023-02-27	Moderna	1018	4874
1	4997	2023-10-09	Sinovac	1019	4997
1	5048	2022-11-03	Novavax	1020	5048

**Output:**

#	Time	Action	Message
1	18:40:17	insert into Schedule ( employeeID, facilityID , scheduleDate , start... )	Error Code: 1644. An employee cannot be scheduled if she/he is not vaccinated in the past six months prior the scheduled date

- Updated schedule after a teacher got infected

```

777
778      -- If a teacher gets infected by COVID-19, then the system should automatically cancel
779      -- all the assignments for the infected teacher for two weeks from the date of infection.
780
781 • INSERT INTO Infections (infectionNumber, medicareNumber, date, type)
782   VALUES
783   (3, 1799, '2023-08-07', 'Other');
784
785 • select * from Schedule;

```

The screenshot shows a database interface with a result grid and an output log.

**Result Grid:**

employeeID	facilityID	scheduleDate	startTime	endTime
1013	1	2023-08-11	04:30:00	05:00:00
1017	2	2023-08-20	08:00:00	11:00:00
*	HULL	HULL	HULL	HULL

**Output:**

#	Time	Action	Message
1	21:39:11	INSERT INTO Infections (infectionNumber, medicareNumber, date...)	1 row(s) affected
2	21:40:19	select *from Schedule LIMIT 0, 1000	2 row(s) returned

## 7) CheckIfContractValidBeforeScheduling

-- trigger 7) Ensure that a employee have a valid contract with a Facility before scheduling

DELIMITER //

CREATE TRIGGER CheckIfContractValidBeforeScheduling

BEFORE INSERT ON Schedule

FOR EACH ROW

BEGIN

DECLARE countValidContract INT;

SET countValidContract=

(SELECT COUNT(\*)

FROM Contract c where c.employeeID>New.employeeID and c.endDate is null and facilityID=new.facilityID);

IF countValidContract = 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'This employee cannot be scheduled.No valid contract with this Facility';

END IF;

END;

//

DELIMITER ;

```

283
284 • Ⓜ INSERT INTO Schedule (
285     employeeID,
286     facilityID,
287     scheduleDate ,
288     startTime ,
289     endTime )
290     VALUES
291     (1002,8,'2023-08-21','17:00:00','21:00:00');
292

```

Output :::::

#	Time	Action	Message
1	14:31:57	INSERT INTO Schedule ( employeeID, facilityID, scheduleDate , startTime , endTime ) VALUES (1002,8,'2023-08-21','17:00:00','21:00:00');	Error Code: 1644. This employee cannot be scheduled.No valid contract with this Facility

## 8) SendEmailOnInfection

If a teacher gets infected by COVID-19, then the system sends an email to inform the principal of the school where the teacher has been infected.

-- trigger 8) checks if a teacher is infected

DELIMITER //

CREATE TRIGGER CheckIfTeacherInfected

AFTER INSERT ON Infections

FOR EACH ROW

BEGIN

DECLARE isTeacher INT;

SET isTeacher = (

SELECT I.medicareNumber

FROM Infections I

```
        INNER JOIN Employees E ON I.medicareNumber = E.medicareNumber
        LEFT JOIN Teachers T ON E.employeeID = T.employeeID
        WHERE infectionID = (SELECT MAX(infectionID) FROM Infections)
    );
```

```
IF isTeacher IS NOT NULL THEN
    CALL InfectedTeacherEmail (isTeacher);
END IF;
```

```
END;
```

```
//
```

```
DELIMITER ;
```

```
-- procedure sends email if is a teacher
```

```
DELIMITER //
```

```
CREATE PROCEDURE InfectedTeacherEmail (
    medicareNumber INT
)
BEGIN
    DECLARE foundTeacherFName VARCHAR(255);
    DECLARE foundTeacherLName VARCHAR(255);
    DECLARE foundTeacherEmail VARCHAR(255);
```

```
DECLARE foundPrincipalFName VARCHAR(255);

DECLARE foundPrincipalLName VARCHAR(255);

DECLARE foundPrincipalEmail VARCHAR(255);

DECLARE foundfacilityID INT;

DECLARE foundEmployeeID INT;

DECLARE foundInfectionDate DATE;

DECLARE foundInfectionType VARCHAR(255);

DECLARE foundFacilityName VARCHAR(255);

DECLARE emailBody TEXT;

DECLARE first80char TEXT;

DECLARE currentDate DATE;
```

```
SELECT P1.firstName AS teacherFName, P1.lastName AS teacherLName, P1.email AS
teacherEmail, P2.firstName AS principalFName, P2.lastName AS principallName, P2.email AS
principalEmail, F.facilityID, E1.employeeID, F.name, I.date, I.type

INTO      foundTeacherFName,      foundTeacherLName,      foundTeacherEmail,
foundPrincipalFName,      foundPrincipalLName,      foundPrincipalEmail,      foundfacilityID,
foundEmployeeID, foundFacilityName, foundInfectionDate, foundInfectionType

FROM Infections I

INNER JOIN Employees E1 ON I.medicareNumber = E1.medicareNumber

INNER JOIN Persons P1 ON E1.medicareNumber = P1.medicareNumber

INNER JOIN Teachers T ON E1.employeeID = T.employeeID

INNER JOIN EducationalFacility EF ON T.educationalFacilityID = EF.educationalFacilityID

INNER JOIN Facilities F ON EF.facilityID = F.facilityID

LEFT JOIN Contract C ON F.facilityID = C.facilityID
```

```

LEFT JOIN Employees E2 ON C.employeeID = E2.employeeID
LEFT JOIN Persons P2 ON E2.medicareNumber = P2.medicareNumber
GROUP BY I.infectionID, I.medicareNumber, P1.firstName, P1.lastName, P1.email,
P2.firstName, P2.lastName, P2.email, F.facilityID, E1.employeeID, F.name, I.date, I.type, role
HAVING infectionID = (SELECT MAX(infectionID) FROM Infections) AND role = 'Principal';

```

```

SET emailBody = CONCAT(foundTeacherFName, ' ', foundTeacherLName, ' who teaches in
your school ', foundFacilityName, ' has been infected with ', foundInfectionType, ' on ',
foundInfectionDate);

```

```
SET first80char = LEFT(emailBody, 80);
```

```
SET currentDate = CURDATE();
```

```

INSERT INTO EmailLog (date, facilityID, employeeID, sender, receiverFirstName,
receiverLastName, subject, body) VALUES (currentDate, foundFacilityID, foundEmployeeID,
foundFacilityName, foundPrincipalFName, foundPrincipalLName, 'Warning', first80char);

```

```
END;
```

```
//
```

```
DELIMITER ;
```

	emailID	date	facilityID	employeeID	sender	receiverFirstName	receiverLastName	subject	body
▶	1	2023-08-10	2	1002	Head	Natali	White	Warning	Mary Rogers who teaches in your school Head has been infected with COVID-19 on 2

## Queries

### -- Query 1

```
SELECT * FROM Facilities;
```

	facilityID	name	phone	webAddress	capacity
▶	1	Base	930727010	www.facility1.com	152
	2	Head	542731214	www.facility2.com	108
	3	Knox	699068911	www.facility3.com	185
	4	Jackson	306677544	www.facility4.com	462
	5	Levi	510938734	www.facility5.com	72
	6	Home	456290238	www.facility6.com	38
	7	Tree	497364493	www.facility7.com	49
	8	Sea	585972116	www.facility8.com	310
	9	Ship	513477902	www.facility9.com	163

DELETE FROM Facilities WHERE facilityID = 9;

	facilityID	name	phone	webAddress	capacity
▶	1	Base	930727010	www.facility1.com	152
	2	Head	542731214	www.facility2.com	108
	3	Knox	699068911	www.facility3.com	185
	4	Jackson	306677544	www.facility4.com	462
	5	Levi	510938734	www.facility5.com	72
	6	Home	456290238	www.facility6.com	38
	7	Tree	497364493	www.facility7.com	49
	8	Sea	585972116	www.facility8.com	310

INSERT INTO Facilities (facilityID, name, phone, webAddress, capacity)

VALUES (9, 'Ship', 513477902, 'www.facility9.com', 0);

	facilityID	name	phone	webAddress	capacity
▶	1	Base	930727010	www.facility1.com	152
	2	Head	542731214	www.facility2.com	108
	3	Knox	699068911	www.facility3.com	185
	4	Jackson	306677544	www.facility4.com	462
	5	Levi	510938734	www.facility5.com	72
	6	Home	456290238	www.facility6.com	38
	7	Tree	497364493	www.facility7.com	49
	8	Sea	585972116	www.facility8.com	310
	9	Ship	513477902	www.facility9.com	0

UPDATE Facilities SET capacity = 163 WHERE facilityID = 9;

	facilityID	name	phone	webAddress	capacity
▶	1	Base	930727010	www.facility1.com	152
	2	Head	542731214	www.facility2.com	108
	3	Knox	699068911	www.facility3.com	185
	4	Jackson	306677544	www.facility4.com	462
	5	Levi	510938734	www.facility5.com	72
	6	Home	456290238	www.facility6.com	38
	7	Tree	497364493	www.facility7.com	49
	8	Sea	585972116	www.facility8.com	310
	9	Ship	513477902	www.facility9.com	163

## -- Query 2

SELECT \* FROM Students;

	studentID	medicareNumber
▶	2018	3123
	2001	6349
	2002	6560
	2003	6640
	2004	6661
	2005	6706
	2006	6736
	2007	6760
	2008	7633
	2009	8179
	2010	8280
	2011	8456
	2012	9388
	2013	9508
	2014	9620
	2015	9671
	2016	9801
	2017	9830

DELETE FROM Students WHERE studentID = 2017;

	studentID	medicareNumber
▶	2018	3123
	2001	6349
	2002	6560
	2003	6640
	2004	6661
	2005	6706
	2006	6736
	2007	6760
	2008	7633
	2009	8179
	2010	8280
	2011	8456
	2012	9388
	2013	9508
	2014	9620
	2015	9671
	2016	9801

INSERT INTO Students (studentID, medicareNumber)

VALUES (2222, 9830);

	studentID	medicareNumber
▶	2018	3123
	2001	6349
	2002	6560
	2003	6640
	2004	6661
	2005	6706
	2006	6736
	2007	6760
	2008	7633
	2009	8179
	2010	8280
	2011	8456
	2012	9388
	2013	9508
	2014	9620
	2015	9671
	2016	9801
	2222	9830

```
UPDATE Students SET studentID = 2017 WHERE studentID = 2222;
```

	studentID	medicareNumber
▶	2018	3123
	2001	6349
	2002	6560
	2003	6640
	2004	6661
	2005	6706
	2006	6736
	2007	6760
	2008	7633
	2009	8179
	2010	8280
	2011	8456
	2012	9388
	2013	9508
	2014	9620
	2015	9671
	2016	9801
	2017	9830

### -- Query 3

```
SELECT * FROM Employees;
```

	employeeID	medicareNumber
▶	1001	1132
	1002	1556
	1003	1799
	1004	1805
	1005	1947
	1006	2068
	1007	2165
	1008	2506
	1009	2515
	1010	2941
	1011	3295
	1012	3667
	1013	3682
	1014	3792
	1015	3837

```
DELETE FROM Employees WHERE employeeID = 1001;
```

	employeeID	medicareNumber
▶	1002	1556
	1003	1799
	1004	1805
	1005	1947
	1006	2068
	1007	2165
	1008	2506
	1009	2515
	1010	2941
	1011	3295
	1012	3667
	1013	3682
	1014	3792
	1015	3837

```
INSERT INTO Employees (employeeID, medicareNumber)
```

```
VALUES (10001, 1132);
```

	employeeID	medicareNumber
▶	10001	1132
	1002	1556
	1003	1799
	1004	1805
	1005	1947
	1006	2068
	1007	2165
	1008	2506
	1009	2515
	1010	2941
	1011	3295
	1012	3667
	1013	3682
	1014	3792
	1015	3837
	1016	4233
	1017	4628

```
UPDATE Employees SET employeeID = 1001 WHERE employeeID = 10001;
```

employeeID	medicareNumber
1001	1132
1002	1556
1003	1799
1004	1805
1005	1947
1006	2068
1007	2165
1008	2506
1009	2515
1010	2941
1011	3295
1012	3667
1013	3682
1014	3792
1015	3837
1016	4233

#### -- Query 4

SELECT \* FROM Vaccinations;

	doseNumber	medicareNumber	date	type
▶	1	1556	2023-07-13	Sinovac
	1	1799	2023-07-15	Johnson & Johnson
	1	1805	2022-10-08	Moderna
	1	2068	2022-09-03	Novavax
	1	2506	2022-11-10	Sinovac
	1	2515	2021-09-13	Johnson & Johnson
	1	2941	2021-01-03	Novavax
	1	3295	2021-03-12	Moderna
	1	3667	2022-02-18	Sinovac
	1	3682	2021-04-22	Novavax
	1	3837	2022-10-17	AstraZeneca
	1	4233	2022-03-02	AstraZeneca
	1	4628	2023-05-18	Sinovac
	1	4874	2023-02-27	Moderna
	1	4997	2022-10-09	Sinovac
	1	5048	2022-11-03	Novavax
	1	5721	2022-09-27	Moderna
	1	5728	2022-09-24	Moderna
	1	5905	2022-01-17	Sinovac
	1	6011	2022-02-12	Johnson & Johnson
	1	6013	2021-10-14	AstraZeneca
	1	6185	2022-04-03	Johnson & Johnson
	1	6349	2021-12-03	Johnson & Johnson
	1	6560	2022-05-22	Novavax

```
DELETE FROM Vaccinations WHERE medicareNumber = 1556 AND doseNumber = 1;
```

	doseNumber	medicareNumber	date	type
▶	1	1799	2023-07-15	Johnson & Johnson
	1	1805	2022-10-08	Moderna
	1	2068	2022-09-03	Novavax
	1	2506	2022-11-10	Sinovac
	1	2515	2021-09-13	Johnson & Johnson
	1	2941	2021-01-03	Novavax
	1	3295	2021-03-12	Moderna
	1	3667	2022-02-18	Sinovac
	1	3682	2021-04-22	Novavax
	1	3837	2022-10-17	AstraZeneca
	1	4233	2022-03-02	AstraZeneca
	1	4628	2023-05-18	Sinovac
	1	4874	2023-02-27	Moderna
	1	4997	2022-10-09	Sinovac
	1	5048	2022-11-03	Novavax
	1	5721	2022-09-27	Moderna
	1	5728	2022-09-24	Moderna
	1	5905	2022-01-17	Sinovac
	1	6011	2022-02-12	Johnson & Johnson
	1	6013	2021-10-14	AstraZeneca
	1	6185	2022-04-03	Johnson & Johnson
	1	6349	2021-12-03	Johnson & Johnson
	1	6560	2022-05-22	Novavax
	1	6640	2022-07-11	Johnson & Johnson

```
INSERT INTO Vaccinations (doseNumber, medicareNumber, date, type)
```

```
VALUES (1, 1556, '2023-01-28', 'Sinovac');
```

	doseNumber	medicareNumber	date	type
▶	1	1556	2023-01-28	Sinovac
	1	1799	2023-07-15	Johnson & Johnson
	1	1805	2022-10-08	Moderna
	1	2068	2022-09-03	Novavax
	1	2506	2022-11-10	Sinovac
	1	2515	2021-09-13	Johnson & Johnson
	1	2941	2021-01-03	Novavax
	1	3295	2021-03-12	Moderna
	1	3667	2022-02-18	Sinovac
	1	3682	2021-04-22	Novavax
	1	3837	2022-10-17	AstraZeneca
	1	4233	2022-03-02	AstraZeneca
	1	4628	2023-05-18	Sinovac
	1	4874	2023-02-27	Moderna
	1	4997	2022-10-09	Sinovac
	1	5048	2022-11-03	Novavax
	1	5721	2022-09-27	Moderna
	1	5728	2022-09-24	Moderna
	1	5905	2022-01-17	Sinovac
	1	6011	2022-02-12	Johnson & Johnson
	1	6013	2021-10-14	AstraZeneca
	1	6185	2022-04-03	Johnson & Johnson
	1	6349	2021-12-03	Johnson & Johnson
	1	6560	2022-05-22	Novavax

UPDATE Vaccinations SET date = '2023-05-05' WHERE doseNumber = 1 AND medicareNumber = 1556;

	doseNumber	medicareNumber	date	type
▶	1	1556	2023-05-05	Sinovac
	1	1799	2023-07-15	Johnson & Johnson
	1	1805	2022-10-08	Moderna
	1	2068	2022-09-03	Novavax
	1	2506	2022-11-10	Sinovac
	1	2515	2021-09-13	Johnson & Johnson
	1	2941	2021-01-03	Novavax
	1	3295	2021-03-12	Moderna
	1	3667	2022-02-18	Sinovac
	1	3682	2021-04-22	Novavax
	1	3837	2022-10-17	AstraZeneca
	1	4233	2022-03-02	AstraZeneca
	1	4628	2023-05-18	Sinovac
	1	4874	2023-02-27	Moderna
	1	4997	2022-10-09	Sinovac
	1	5048	2022-11-03	Novavax
	1	5721	2022-09-27	Moderna
	1	5728	2022-09-24	Moderna
	1	5905	2022-01-17	Sinovac
	1	6011	2022-02-12	Johnson & Johnson
	1	6013	2021-10-14	AstraZeneca
	1	6185	2022-04-03	Johnson & Johnson
	1	6349	2021-12-03	Johnson & Johnson
	1	6560	2022-05-22	Novavax

-- Query 5

SELECT \* FROM Infections;

	infectionID	infectionNumber	medicareNumber	date	type
▶	1	1	9830	2022-12-17	Other
	2	1	9801	2023-01-28	SARS-CoV-2 Variant
	3	1	5400	2021-11-08	Other
	4	1	5860	2023-03-26	Other
	5	3	4874	2023-07-14	COVID-19
	6	1	6349	2023-03-09	COVID-19
	7	1	1947	2023-01-04	COVID-19
	8	1	4233	2022-11-17	COVID-19
	9	1	6706	2022-03-22	COVID-19
	10	1	6736	2022-08-15	SARS-CoV-2 Variant
	11	3	6011	2023-06-24	Other
	12	2	8179	2023-01-03	SARS-CoV-2 Variant
	13	1	4874	2022-05-03	COVID-19
	14	2	1947	2023-05-26	SARS-CoV-2 Variant
	15	1	8179	2022-08-22	SARS-CoV-2 Variant
	16	2	1799	2023-05-26	Other
	17	1	8280	2022-06-19	SARS-CoV-2 Variant
	18	1	9508	2021-03-04	COVID-19
	19	2	2506	2023-06-10	SARS-CoV-2 Variant
	20	2	2515	2023-01-22	COVID-19
	21	1	2506	2022-08-14	SARS-CoV-2 Variant
	22	1	1805	2022-05-21	SARS-CoV-2 Variant
	23	3	8280	2023-07-17	SARS-CoV-2 Variant
	24	1	3295	2023-01-21	SARS-CoV-2 Variant
	25	1	4628	2023-02-17	SARS-CoV-2 Variant
	26	1	6011	2022-01-13	Other
	27	2	4874	2023-06-19	COVID-19
	28	1	3667	2023-03-02	SARS-CoV-2 Variant
	29	1	9671	2022-08-05	Other

DELETE FROM Infections WHERE infectionID = 1;

	infectionID	infectionNumber	medicareNumber	date	type
▶	2	1	9801	2023-01-28	SARS-CoV-2 Variant
	3	1	5400	2021-11-08	Other
	4	1	5860	2023-03-26	Other
	5	3	4874	2023-07-14	COVID-19
	6	1	6349	2023-03-09	COVID-19
	7	1	1947	2023-01-04	COVID-19
	8	1	4233	2022-11-17	COVID-19
	9	1	6706	2022-03-22	COVID-19
	10	1	6736	2022-08-15	SARS-CoV-2 Variant
	11	3	6011	2023-06-24	Other
	12	2	8179	2023-01-03	SARS-CoV-2 Variant
	13	1	4874	2022-05-03	COVID-19
	14	2	1947	2023-05-26	SARS-CoV-2 Variant
	15	1	8179	2022-08-22	SARS-CoV-2 Variant
	16	2	1799	2023-05-26	Other
	17	1	8280	2022-06-19	SARS-CoV-2 Variant
	18	1	9508	2021-03-04	COVID-19
	19	2	2506	2023-06-10	SARS-CoV-2 Variant
	20	2	2515	2023-01-22	COVID-19
	21	1	2506	2022-08-14	SARS-CoV-2 Variant
	22	1	1805	2022-05-21	SARS-CoV-2 Variant
	23	3	8280	2023-07-17	SARS-CoV-2 Variant
	24	1	3295	2023-01-21	SARS-CoV-2 Variant
	25	1	4628	2023-02-17	SARS-CoV-2 Variant

```
INSERT INTO Infections (infectionID, infectionNumber, medicareNumber, date, type)
VALUES (1, 1, 9801, '2023-01-28', 'COVID-19');
```

	infectionID	infectionNumber	medicareNumber	date	type
▶	1	1	9801	2023-01-28	COVID-19
	2	1	9801	2023-01-28	SARS-Cov-2 Variant
	3	1	5400	2021-11-08	Other
	4	1	5860	2023-03-26	Other
	5	3	4874	2023-07-14	COVID-19
	6	1	6349	2023-03-09	COVID-19
	7	1	1947	2023-01-04	COVID-19
	8	1	4233	2022-11-17	COVID-19
	9	1	6706	2022-03-22	COVID-19
	10	1	6736	2022-08-15	SARS-Cov-2 Variant
	11	3	6011	2023-06-24	Other
	12	2	8179	2023-01-03	SARS-Cov-2 Variant
	13	1	4874	2022-05-03	COVID-19
	14	2	1947	2023-05-26	SARS-Cov-2 Variant
	15	1	8179	2022-08-22	SARS-Cov-2 Variant
	16	2	1799	2023-05-26	Other
	17	1	8280	2022-06-19	SARS-Cov-2 Variant
	18	1	9508	2021-03-04	COVID-19
	19	2	2506	2023-06-10	SARS-Cov-2 Variant
	20	2	2515	2023-01-22	COVID-19
	21	1	2506	2022-08-14	SARS-Cov-2 Variant
	22	1	1805	2022-05-21	SARS-Cov-2 Variant
	23	3	8280	2023-07-17	SARS-Cov-2 Variant
	24	1	2005	2023-01-21	SARS-Cov-2 Variant

UPDATE Infections SET date = '2023-05-05' WHERE infectionID = 1;

	infectionID	infectionNumber	medicareNumber	date	type
▶	1	1	9801	2023-05-05	COVID-19
	2	1	9801	2023-01-28	SARS-Cov-2 Variant
	3	1	5400	2021-11-08	Other
	4	1	5860	2023-03-26	Other
	5	3	4874	2023-07-14	COVID-19
	6	1	6349	2023-03-09	COVID-19
	7	1	1947	2023-01-04	COVID-19
	8	1	4233	2022-11-17	COVID-19
	9	1	6706	2022-03-22	COVID-19
	10	1	6736	2022-08-15	SARS-Cov-2 Variant
	11	3	6011	2023-06-24	Other
	12	2	8179	2023-01-03	SARS-Cov-2 Variant
	13	1	4874	2022-05-03	COVID-19
	14	2	1947	2023-05-26	SARS-Cov-2 Variant
	15	1	8179	2022-08-22	SARS-Cov-2 Variant
	16	2	1799	2023-05-26	Other
	17	1	8280	2022-06-19	SARS-Cov-2 Variant
	18	1	9508	2021-03-04	COVID-19
	19	2	2506	2023-06-10	SARS-Cov-2 Variant
	20	2	2515	2023-01-22	COVID-19
	21	1	2506	2022-08-14	SARS-Cov-2 Variant
	22	1	1805	2022-05-21	SARS-Cov-2 Variant
	23	3	8280	2023-07-17	SARS-Cov-2 Variant
	24	1	3295	2023-01-21	SARS-Cov-2 Variant

## -- Query 6

SELECT \* FROM Registration;

	studentID	facilityID	startDate	endDate	level
▶	2001	1	2023-11-23	2024-12-15	5
	2002	1	2022-07-16	NULL	12
	2003	3	2022-12-28	NULL	6
	2004	1	2023-05-14	NULL	10
	2005	3	2022-10-06	2024-01-27	2
	2006	3	2022-01-17	2024-04-23	8
	2007	1	2022-02-14	2023-05-01	6
	2008	1	2022-04-08	2024-09-15	9
	2009	2	2022-08-21	2023-05-05	5
	2010	2	2023-08-06	2023-07-24	2
	2011	2	2023-09-09	2023-05-16	12
	2012	3	2023-03-17	2023-04-23	1
	2013	2	2023-06-03	2024-07-11	8
	2014	1	2022-11-03	2023-08-01	1
	2015	2	2022-11-21	2024-06-06	4
	2016	1	2022-04-05	NULL	1
	2017	1	2022-11-10	2024-05-05	12

```
DELETE FROM Registration WHERE studentID = 2017;
```

	studentID	facilityID	startDate	endDate	level
▶	2001	1	2023-11-23	2024-12-15	5
	2002	1	2022-07-16	NULL	12
	2003	3	2022-12-28	NULL	6
	2004	1	2023-05-14	NULL	10
	2005	3	2022-10-06	2024-01-27	2
	2006	3	2022-01-17	2024-04-23	8
	2007	1	2022-02-14	2023-05-01	6
	2008	1	2022-04-08	2024-09-15	9
	2009	2	2022-08-21	2023-05-05	5
	2010	2	2023-08-06	2023-07-24	2
	2011	2	2023-09-09	2023-05-16	12
	2012	3	2023-03-17	2023-04-23	1
	2013	2	2023-06-03	2024-07-11	8
	2014	1	2022-11-03	2023-08-01	1
	2015	2	2022-11-21	2024-06-06	4
	2016	1	2022-04-05	NULL	1

```
INSERT INTO Registration (studentID, facilityID, startDate, endDate, level)
```

```
VALUES (2017, 1, '2022-11-11', '2024-05-05', 12);
```

	studentID	facilityID	startDate	endDate	level
▶	2001	1	2023-11-23	2024-12-15	5
	2002	1	2022-07-16	NULL	12
	2003	3	2022-12-28	NULL	6
	2004	1	2023-05-14	NULL	10
	2005	3	2022-10-06	2024-01-27	2
	2006	3	2022-01-17	2024-04-23	8
	2007	1	2022-02-14	2023-05-01	6
	2008	1	2022-04-08	2024-09-15	9
	2009	2	2022-08-21	2023-05-05	5
	2010	2	2023-08-06	2023-07-24	2
	2011	2	2023-09-09	2023-05-16	12
	2012	3	2023-03-17	2023-04-23	1
	2013	2	2023-06-03	2024-07-11	8
	2014	1	2022-11-03	2023-08-01	1
	2015	2	2022-11-21	2024-06-06	4
	2016	1	2022-04-05	NULL	1
	2017	1	2022-11-11	2024-05-05	12

```
UPDATE Registration SET endDate = '2023-05-05' WHERE studentID = 2017;
```

	studentID	facilityID	startDate	endDate	level
▶	2001	1	2023-11-23	2024-12-15	5
	2002	1	2022-07-16	NULL	12
	2003	3	2022-12-28	NULL	6
	2004	1	2023-05-14	NULL	10
	2005	3	2022-10-06	2024-01-27	2
	2006	3	2022-01-17	2024-04-23	8
	2007	1	2022-02-14	2023-05-01	6
	2008	1	2022-04-08	2024-09-15	9
	2009	2	2022-08-21	2023-05-05	5
	2010	2	2023-08-06	2023-07-24	2
	2011	2	2023-09-09	2023-05-16	12
	2012	3	2023-03-17	2023-04-23	1
	2013	2	2023-06-03	2024-07-11	8
	2014	1	2022-11-03	2023-08-01	1
	2015	2	2022-11-21	2024-06-06	4
	2016	1	2022-04-05	NULL	1
	2017	1	2022-11-11	2023-05-05	12

## -- Query 7

See trigger 3) CheckOneHourBeetweenSchedule

## -- Query 8

SELECT

F.name AS facilityName,

L.address,

L.city,

L.province,

L.postalCode,

F.phone,

F.webAddress,

CASE

WHEN EF.educationalFacilityID IS NOT NULL THEN EF.schoolType

WHEN MF.managementFacilityID IS NOT NULL THEN 'Management'

ELSE 'Unknown'  
END AS type,  
F.capacity,  
Per.firstName AS presidentFirstName,  
Per.lastName AS presidentLastName,  
COALESCE(EmpCount.numberOfEmployees, 0) AS numberOfEmployees  
FROM  
Facilities F  
LEFT JOIN  
Situated S ON F.facilityID = S.facilityID  
LEFT JOIN  
Location L ON S.addressID = L.addressID  
LEFT JOIN  
ManagementFacility MF ON F.facilityID = MF.facilityID  
LEFT JOIN  
Employees Emp ON MF.president = Emp.employeeID  
LEFT JOIN  
Persons Per ON Emp.medicareNumber = Per.medicareNumber  
LEFT JOIN  
EducationalFacility EF ON F.facilityID = EF.facilityID  
LEFT JOIN  
(SELECT facilityID, COUNT(employeeID) AS numberOfEmployees FROM Contract GROUP BY facilityID) AS EmpCount  
ON F.facilityID = EmpCount.facilityID  
ORDER BY

L.province ASC, L.city ASC, type ASC, EmpCount.numberOfEmployees ASC;

facilityName	address	city	province	postalCode	phone	webAddress	type	capacity	presidentFirstName	presidentLastName	numberOfEmployees
Ship	216	Ottawa	Ontario	H7K6L9	513477902	www.facility9.com	Management	163	Nathan	Kelly	5
Knox	937	Toronto	Ontario	H2K8L7	699068911	www.facility3.com	High	185	NULL	NULL	3
Tree	429	Toronto	Ontario	H0K0L2	497364493	www.facility7.com	Management	49	Jack	Roberts	3
Jackson	118	Laval	Quebec	H1K5L1	306677544	www.facility4.com	Management	462	Maria	Evans	4
Levi	253	Laval	Quebec	H1K3L1	510938734	www.facility5.com	Management	72	Natalie	Williams	4
Head	714	Laval	Quebec	H2K5L1	542731214	www.facility2.com	Middle	108	NULL	NULL	2
Base	462	Laval	Quebec	H3K9L4	930727010	www.facility1.com	Primary	152	NULL	NULL	3
Home	424	Montreal	Quebec	H9K9L2	456290238	www.facility6.com	Management	38	Natalya	Baker	3
Sea	391	Montreal	Quebec	H5K5L9	585972116	www.facility8.com	Management	310	Nathaly	Howard	3

-- Query 9 (managementFacility is chosen)

SELECT

P.firstName,

P.lastName,

Em.startDate AS startDateOfWork,

P.dateOfBirth,

P.medicareNumber,

P.phone AS telephoneNumber,

L.address,

L.city,

L.province,

L.postalCode,

P.citizenship,

P.email,

Em.role

FROM

Employees Emp

JOIN

Persons P ON Emp.medicareNumber = P.medicareNumber

JOIN

Contract Em ON Emp.employeeID = Em.employeeID

JOIN

Facilities F ON Em.facilityID = F.facilityID

JOIN

ManagementFacility MF ON F.facilityID = MF.facilityID

LEFT JOIN

Residence R ON P.medicareNumber = R.medicareNumber

LEFT JOIN

Location L ON R.addressID = L.addressID

ORDER BY

Em.role ASC, P.firstName ASC, P.lastName ASC;

firstName	lastName	startDateOfWork	dateOfBirth	medicareNumber	telephoneNumber	address	city	province	postalCode	citizenship	email	role
Joe	Miller	2023-01-01	1988-09-16	6013	295279935	228	Toronto	Ontario	H7KG5L	Canadian	person9@example.com	other
Wei	Su	2023-01-01	2023-08-16	6185	5142946093	444	Pointe-Claire	Quebec	H9R0G8	Canadian	hi.wei.su@gmail.com	other
Jack	Roberts	2023-01-01	1998-04-14	2165	648481342	587	Toronto	Ontario	H1KG1L	American	person18@example.com	President
Maria	Evans	2023-01-01	1962-06-19	1805	958736466	611	Montreal	Quebec	H1KG3L	French	person6@example.com	President
Natalie	Williams	2023-01-01	1978-04-25	1947	821922687	673	Laval	Quebec	H4KG3L	Australian	person40@example.com	President
Natalya	Baker	2023-01-01	1967-07-05	2068	962355955	973	Toronto	Ontario	H4KG8L	German	person8@example.com	President
Nathaly	Howard	2023-01-01	1977-02-19	2506	945519013	588	Laval	Quebec	H1KG4L	Canadian	person23@example.com	President
Nathan	Kelly	2023-01-01	1976-03-09	2515	278047016	284	Toronto	Ontario	H2KG4L	American	person24@example.com	President
Jack	Robinson	2023-01-01	1992-01-08	4997	581147180	411	Laval	Quebec	H8KG3L	French	person17@example.com	Secretary
Maria	Russell	2023-01-01	1996-08-23	3837	957211208	959	Montreal	Quebec	H1KG9L	Canadian	person16@example.com	Secretary
Nancy	Rogers	2023-01-01	1984-06-22	4628	720769265	736	Laval	Quebec	H4KG6L	French	person25@example.com	Secretary
Nathalie	Walker	2023-01-01	1998-06-18	4874	783500106	920	Montreal	Quebec	H7KG2L	German	person21@example.com	Secretary
Nevaeh	Johnson	2023-01-01	1995-06-11	3682	938775297	707	Montreal	Quebec	H7KG1L	Canadian	person35@example.com	Secretary
Nora	White	2023-01-01	1994-01-09	4233	785019153	299	Laval	Quebec	H3KG3L	British	person22@example.com	Secretary
Steven	James	2023-01-01	1995-10-23	3792	417204837	772	Laval	Quebec	H4KG6L	American	person3@example.com	Secretary
Tom	Phillips	2023-01-01	2002-11-04	5048	218038875	301	Montreal	Quebec	H0KG0L	Canadian	person1@example.com	Secretary
Nathalie	Cooper	2023-01-01	1964-02-18	5905	974005556	452	Montreal	Quebec	H6KG5L	Canadian	person45@example.com	Security
Nathan	Carte	2023-01-01	1996-05-12	5400	781192683	943	Montreal	Quebec	H9KG5L	Canadian	person42@example.com	Security
Nathan	Diaz	2023-01-01	2003-08-12	5860	692850501	904	Montreal	Quebec	H9KG8L	British	person36@example.com	Security
Nia	Cook	2023-01-01	1975-04-26	5728	681659378	246	Ottawa	Ontario	H3KG7L	American	person34@example.com	Security
Tom	Martinez	2023-01-01	1974-09-01	5721	639881652	717	Laval	Quebec	H8KG0L	German	person11@example.com	Security
Tom	Price	2023-01-01	2000-11-27	6011	596009224	829	Ottawa	Ontario	H7KG4L	Australian	person43@example.com	Security

-- Query 10 (employeeID is chosen)

```
SELECT f.name,
       s.scheduleDate,
       s.startTime,
       s.endTime
  FROM Facilities f
 INNER JOIN Schedule s WHERE employeeID=1002 AND f.facilityID=s.facilityID
 ORDER BY
       f.name,
       s.scheduleDate,
       s.startTime,
       s.endTime ASC;
```

```
268
269 •  SELECT f.name, s.scheduleDate, s.startTime, s.endTime
270   FROM Facilities f INNER JOIN Schedule s WHERE employeeID=1002 AND f.facilityID=s.facilityID
271   ORDER BY f.name, s.scheduleDate,s.startTime, s.endTime ASC;
272
273
```

Result Grid				
	name	scheduleDate	startTime	endTime
▶	Base	2023-08-10	11:00:00	13:00:00
	Knox	2023-08-13	08:00:00	09:00:00
	Knox	2023-08-14	11:00:00	13:00:00

-- Query 11

```
SELECT
    P.firstName,
    P.lastName,
    Inf.date AS dateOfInfection,
    F.name AS facilityName
FROM
    Teachers T
JOIN
    Employees E ON T.employeeID = E.employeeID
JOIN
    Persons P ON E.medicareNumber = P.medicareNumber
JOIN
    Infections Inf ON P.medicareNumber = Inf.medicareNumber
JOIN
    Contract Emp ON E.employeeID = Emp.employeeID
JOIN
    Facilities F ON Emp.facilityID = F.facilityID
ORDER BY
    F.name ASC,
    P.firstName ASC,
    P.lastName ASC
LIMIT 0, 1000;
```

firstName	lastName	dateOfInfection	facilityName
Jack	Alexander	2023-04-13	Base
Jack	Alexander	2023-08-01	Base
Jack	Alexander	2023-04-13	Base
Jack	Alexander	2023-08-01	Base
Mary	Rogers	2022-07-09	Head
Mary	Rogers	2023-01-01	Head
Mary	Rogers	2023-03-09	Head
Mary	Rogers	2022-07-09	Knox
Mary	Rogers	2023-01-01	Knox
Mary	Rogers	2023-03-09	Knox
Natalya	Carter	2023-08-01	Knox
Natalya	Carter	2023-05-26	Knox

– Query 12 (chooseFacilityID of user's choice)

```

SELECT *
FROM EmailLog
-- WHERE facilityID = chosenFacilityID
ORDER BY facilityID ASC, date ASC;

```

	logID	date	facilityID	employeeID	sender	receiverFirstName	receiverLastName	subject	body
▶	1	2023-08-10 00:00:00	3	1002	Knox	Mary	Rogers	Knox Schedule for 10-Aug-2023 to 13-Aug-2023	Facility: Knox Address: 937 Employee: Mary Rogers (person33@example.com) Thurs
	2	2023-08-10 00:00:00	4	1013	Jackson	Nevaeh	Johnson	Jackson Schedule for 10-Aug-2023 to 13-Aug-2023	Facility: Jackson Address: 118 Employee: Nevaeh Johnson (person35@example.com)

firstName	lastName	dateOfInfection	facilityName
Jack	Alexander	2023-04-13	Base
Jack	Alexander	2023-08-01	Base
Jack	Alexander	2023-04-13	Base
Jack	Alexander	2023-08-01	Base
Mary	Rogers	2022-07-09	Head
Mary	Rogers	2023-01-01	Head
Mary	Rogers	2023-03-09	Head
Mary	Rogers	2022-07-09	Knox
Mary	Rogers	2023-01-01	Knox
Mary	Rogers	2023-03-09	Knox
Natalya	Carter	2023-08-01	Knox
Natalya	Carter	2023-05-26	Knox

– **Query 13**

```

SELECT DISTINCT

P.firstName,
P.lastName,
T.teacherID,
T.educationalFacilityID

FROM

Teachers T

JOIN

Employees E ON T.employeeID = E.employeeID

JOIN

Persons P ON E.medicareNumber = P.medicareNumber

JOIN

```

Schedule S ON E.employeeID = S.employeeID

JOIN

Contract C ON E.employeeID = C.employeeID

JOIN

Facilities F ON C.facilityID = F.facilityID

WHERE

S.scheduleDate >= DATE\_SUB(CURRENT\_DATE, INTERVAL 2 WEEK)

AND F.facilityID = 1

ORDER BY

T.educationalFacilityID ASC,

P.firstName ASC;

```
18 WHERE
19     S.scheduleDate >= DATE_SUB(CURRENT_DATE, INTERVAL 2 WEEK)
20     AND F.facilityID = 1
21 ORDER BY
22     T.educationalFacilityID ASC,
23     P.firstName ASC;
24
```

The screenshot shows a database query results grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below is a table with four columns: 'firstName', 'lastName', 'teacherID', and 'educationalFacilityID'. A single row of data is shown: Mary, Rogers, 2, 2.

	firstName	lastName	teacherID	educationalFacilityID
▶	Mary	Rogers	2	2

-- Query 14 (facilityID is chosen)

SELECT

p.firstName,

p.lastName ,

```

SUM( TIME_TO_SEC(TIMEDIFF(endTime, startTime)) / 3600) AS NumberOfhours,f.name AS
facilityName

FROM Persons p

INNER JOIN Employees e ON p.medicareNumber=e.medicareNumber

INNER JOIN Schedule s ON s.employeeID=e.employeeID

INNER JOIN Facilities f ON f.facilityID=s.facilityID

WHERE f.facilityID=3

GROUP BY e.employeeID

ORDER BY p.firstName,p.lastName ASC;

```

```

280 • SELECT p.firstName,p.lastName , SUM( TIME_TO_SEC(TIMEDIFF(endTime, startTime)) / 3600) AS NumberOfhours,f.name AS facilityName
281   FROM Persons p INNER JOIN Employees e ON p.medicareNumber=e.medicareNumber
282   INNER JOIN Schedule s ON s.employeeID=e.employeeID
283   INNER JOIN Facilities f ON f.facilityID=s.facilityID
284   WHERE f.facilityID=3
285   GROUP BY e.employeeID
286   ORDER BY p.firstName,p.lastName ASC;
287
288

```

Result Grid			
	firstName	lastName	NumberOfhours
▶	Mary	Rogers	3.0000
	Knox		

### - Query 15

```

CREATE TABLE high_schools (
    school_id INT PRIMARY KEY,
    province VARCHAR(50),
    school_name VARCHAR(100),
    capacity INT

```

);

-- Insert data into high\_schools table

INSERT INTO high\_schools (school\_id, province, school\_name, capacity)

VALUES

(1, 'QC', 'High School 1', 500),

(2, 'Ontario', 'High School 2', 700),

(3, 'Alberta', 'High School 3', 530),

(4, 'QC', 'High School 4', 987),

(5, 'Ontario', 'High School 5', 300),

(6, 'Alberta', 'High School 6', 860),

(7, 'QC', 'High School 7', 700),

(8, 'Ontario', 'High School 8', 500);

SELECT

HS.province,

HS.school\_name AS schoolName,

HS.capacity,

COUNT(DISTINCT CASE WHEN TInf.date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 2 WEEK) THEN T.employeeID END) AS teachersInfected,

COUNT(DISTINCT CASE WHEN SInf.date >= DATE\_SUB(CURRENT\_DATE, INTERVAL 2 WEEK) THEN S.studentID END) AS studentsInfected

FROM

high\_schools HS

LEFT JOIN

Teachers T ON HS.school\_id = T.teacherID

LEFT JOIN

Schedule Sch ON T.employeeID = Sch.employeeID

LEFT JOIN

Infections TInf ON T.employeeID = TInf.medicareNumber

LEFT JOIN

Students S ON HS.school\_id = S.studentID

LEFT JOIN

Infections SInf ON S.studentID = SInf.medicareNumber

GROUP BY

HS.province,

HS.school\_name,

HS.capacity

ORDER BY

HS.province ASC,

teachersInfected ASC;

Province	School Name	Capacity	Teachers Infected	Students Infected
Alberta	High School 3	530	0	0
Alberta	High School 6	860	0	0
Ontario	High School 2	700	0	0
▶ Ontario	High School 5	300	0	0
Ontario	High School 8	500	0	0
QC	High School 1	500	0	0
QC	High School 4	987	0	0
QC	High School 7	700	0	0

## – Query 16

SELECT

M.ministryID,  
M.sector,  
SQ.ministerFirstName,  
SQ.ministerLastName,  
SQ.city as ministerCity,  
COUNT(DISTINCT MF.managementFacilityID) AS totalManagementFacilities,  
COUNT(DISTINCT EF.educationalFacilityID) AS totalEducationalFacilities  
FROM  
Ministries M  
INNER JOIN Operation O ON M.ministryID = O.ministryID  
INNER JOIN Facilities F ON O.facilityID = F.facilityID  
LEFT JOIN ManagementFacility MF ON O.facilityID = MF.facilityID  
LEFT JOIN EducationalFacility EF ON F.facilityID = EF.facilityID  
INNER JOIN (SELECT  
M.ministryID,  
P.firstName AS ministerFirstName,  
P.lastName AS ministerLastName,  
L.city AS city,  
HO.headOfficeID  
FROM  
Ministries M  
INNER JOIN Operation O ON M.ministryID = O.ministryID  
INNER JOIN Facilities F ON O.facilityID = F.facilityID  
LEFT JOIN ManagementFacility MF ON O.facilityID = MF.facilityID

```

        LEFT JOIN HeadOffice HO ON MF.managementFacilityID =
HO.managementFacilityID

        LEFT JOIN Employees E ON MF.president = E.employeeID

        LEFT JOIN Contract C ON E.employeeID = C.employeeID

        LEFT JOIN Persons P ON E.medicareNumber = P.medicareNumber

        LEFT JOIN Residence R ON P.medicareNumber = R.medicareNumber

        LEFT JOIN Location L ON R.addressID = L.addressID

        GROUP BY M.ministryID, P.firstName, P.lastName, L.city, HO.headOfficeID

        HAVING headOfficeID IS NOT NULL

    ) AS SQ ON M.ministryID = SQ.ministryID

GROUP BY

M.ministryID, SQ.ministerFirstName, SQ.ministerLastName, SQ.city

```

	ministryID	sector	ministerFirstName	ministerLastName	ministerCity	totalManagementFacilities	totalEducationalFacilities
▶	1	Agriculture	Maria	Evans	Montreal	2	1
	2	Health	Natalya	Baker	Toronto	2	1
	3	Energy	Nathaly	Howard	Laval	2	1

## – Query 17

```

SELECT

firstName,

lastName,

MIN(startDate) AS startDate,

schoolType AS role,

dateOfBirth,

email,

```

```
    TIME(hours.totalWorkedTime) AS totalWorkedTime
  FROM SecondaryTeacher ST
  LEFT JOIN Teachers T ON ST.teacherID = T.teacherID
  LEFT JOIN EducationalFacility EF ON T.educationalFacilityID = EF.educationalFacilityID
  LEFT JOIN Employees E ON T.employeeID = E.employeeID
  LEFT JOIN Contract C ON E.employeeID = C.employeeID
  LEFT JOIN Persons P ON E.medicareNumber = P.medicareNumber
  LEFT JOIN Infections I ON P.medicareNumber = I.medicareNumber
  LEFT JOIN (
    SELECT
      secondaryTeacherID,
      SUM(endTime - startTime) AS totalWorkedTime
    FROM SecondaryTeacher ST
    LEFT JOIN Teachers T ON ST.teacherID = T.teacherID
    LEFT JOIN Employees E ON T.employeeID = E.employeeID
    LEFT JOIN Schedule S ON E.employeeID = S.employeeID
    GROUP BY secondaryTeacherID, role
    HAVING role = 'Counselor'
  ) AS hours ON ST.secondaryTeacherID = hours.secondaryTeacherID
  WHERE ST.role = 'Counselor' AND infectionNumber >= 3 AND endDate IS NULL
  GROUP BY ST.secondaryTeacherID, firstName, lastName, role, dateOfBirth, email,
  totalWorkedTime
  ORDER BY role ASC, firstName ASC, lastName ASC;
```

	firstName	lastName	startDate	role	dateOfBirth	email	totalWorkedTime
▶	Mary	Rogers	2023-03-02	Middle	1985-05-14	person33@example.com	05:00:00

## User Interface

index.php

The screenshot shows a web browser window with the URL [jdc353.encs.concordia.ca/index.php](http://jdc353.encs.concordia.ca/index.php). The page title is "COMP 353 - Section CC Databases". Below the title, it says "Instructor: Prof. Khaled Jababo" and "Summer 2023". A section titled "Project done by:" lists five group members: Yujing Wang (40098623), Yifan Wu (40153584), Wei Su (27072473), and Mikelson Benjamin (40158499). The page also includes a navigation bar with links to Home, Facilities, Persons, Students, Employees, Vaccinations, and Infections.

facilities.php

The screenshot shows a web browser window with the URL [jdc353.encs.concordia.ca/facilities.php](http://jdc353.encs.concordia.ca/facilities.php). The page title is "Facilities". It displays a table with 10 rows of facility data, each with "Edit" and "Delete" buttons. The columns are Facility ID, Name, Address, City, Province, Postal Code, Phone, Web Address, and Capacity. The last row contains an "Add" button. The page also includes a navigation bar with links to Home, Facilities, Persons, Students, Employees, Vaccinations, and Infections.

Facility ID	Name	Address	City	Province	Postal Code	Phone	Web Address	Capacity	Edit	Delete
1	Base	462	Laval	Quebec	H3K9L4	930727010	www.facility1.com	152	<button>Edit</button>	<button>Delete</button>
2	Head	714	Laval	Quebec	H2K5L1	542731214	www.facility2.com	108	<button>Edit</button>	<button>Delete</button>
3	Knox	937	Toronto	Ontario	H2K8L7	699068911	www.facility3.com	185	<button>Edit</button>	<button>Delete</button>
4	Jackson	118	Laval	Quebec	H1K5L1	306677544	www.facility4.com	462	<button>Edit</button>	<button>Delete</button>
5	Levi	253	Laval	Quebec	H1K3L1	510938734	www.facility5.com	72	<button>Edit</button>	<button>Delete</button>
6	Home	424	Montreal	Quebec	H9K9L2	456290238	www.facility6.com	38	<button>Edit</button>	<button>Delete</button>
7	Tree	429	Toronto	Ontario	H0K0L2	497364493	www.facility7.com	49	<button>Edit</button>	<button>Delete</button>
8	Sea	391	Montreal	Quebec	H5K5L9	585972116	www.facility8.com	310	<button>Edit</button>	<button>Delete</button>
9	Ship	216	Ottawa	Ontario	H7K6L9	513477902	www.facility9.com	163	<button>Edit</button>	<button>Delete</button>
									<button>Add</button>	

## edit.php?facilityID=X

← → ⌂ 🔒 jdc353.encts.concordia.ca/edit.php?facilityID=1

Home Facilities Persons Students Employees Vaccinations Infections

### Edit Facility

Facility ID	Name	Address	City	Province	Postal Code	Phone	Web Address	Capacity	
1									<button>Update</button>

## persons.php

← → ⌂ 🔒 jdc353.encts.concordia.ca/persons.php

Home Facilities Persons Students Employees Vaccinations Infections

### Persons

Medicare Number	First Name	Last Name	Medicare Expiry Date	Date of Birth	Phone	Address	City	Province	Postal Code	Citizenship	Email
1132	Jack	Alexander	2024-10-27	2001-02-07	997905683	228	Ottawa	Ontario	H4KG8L	American	person32@example.com
1556	Mary	Rogers	2023-05-28	1985-05-14	581046349	514	Laval	Quebec	H0KG4L	American	person33@example.com
1799	Natalya	Carter	2025-11-19	1964-11-20	859210472	930	Montreal	Quebec	H1KG1L	German	person29@example.com
1805	Maria	Evans	2022-09-19	1962-06-19	958736466	611	Montreal	Quebec	H1KG3L	French	person6@example.com
1947	Natalie	Williams	2024-03-13	1978-04-25	821922687	673	Laval	Quebec	H4KG3L	Australian	person40@example.com
2068	Natalya	Baker	2023-03-24	1967-07-05	962355955	973	Toronto	Ontario	H4KG8L	German	person8@example.com
2165	Jack	Roberts	2024-08-14	1998-04-14	648481342	587	Toronto	Ontario	H1KG1L	American	person18@example.com
2506	Nathaly	Howard	2023-02-10	1977-02-19	945519013	588	Laval	Quebec	H1KG4L	Canadian	person23@example.com
2515	Nathan	Kelly	2022-03-05	1976-03-09	278047016	284	Toronto	Ontario	H2KG4L	American	person24@example.com
2941	Steven	Murphy	2023-09-28	1960-09-22	736149025	852	Ottawa	Ontario	H4KG4L	French	person12@example.com
3123	Wei	Su	2023-08-30	2023-08-30	5142946093	332	POINTE-CLAIRE	Quebec	H9R 1E8	Canadian	hi.wei.su@gmail.com
3295	Natali	White	2025-06-12	1990-10-16	303711351	374	Toronto	Ontario	H5KG2L	French	person4@example.com
3667	Julia	Kelly	2024-06-15	1979-07-19	122821961	233	Montreal	Quebec	H5KG8L	Australian	person30@example.com
3682	Nevaeh	Johnson	2024-11-13	1995-06-11	938775297	707	Montreal	Quebec	H7KG1L	Canadian	person35@example.com
3792	Steven	James	2022-01-26	1995-10-23	417204837	772	Laval	Quebec	H4KG6L	American	person3@example.com
3837	Maria	Russell	2023-07-03	1996-08-23	957211208	959	Montreal	Quebec	H1KG9L	Canadian	person16@example.com
4233	Nora	White	2025-09-09	1994-01-09	785019153	299	Laval	Quebec	H3KG3L	British	person22@example.com
4628	Nancy	Rogers	2023-02-02	1984-06-22	720769265	736	Laval	Quebec	H4KG6L	French	person25@example.com
4874	Nathalie	Walker	2024-07-21	1998-06-18	783500106	920	Montreal	Quebec	H7KG2L	German	person21@example.com
4997	Jack	Robinson	2023-06-26	1992-01-08	581147180	411	Laval	Quebec	H8KG3L	French	person17@example.com

## students.php

## Students

Student ID	First Name	Last Name	Medicare Number	Medicare Expiry Date	Date of Birth	Phone	Address	City	Province	Postal Code	Citizenship	Email		
2001	Nicole	Jackson	6349	2024-11-05	1988-09-14	777745522	124	Montreal	Quebec	H8KG0L	Canadian	person7@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2002	John	Gonzalez	6560	2023-10-08	1962-03-18	143748391	228	Toronto	Ontario	H4KG5L	Australian	person19@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2003	Natalia	Lewis	6640	2024-12-27	2003-04-05	373676914	327	Laval	Quebec	H3KG5L	French	person37@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2004	Nia	Murphy	6661	2025-04-17	1967-02-27	912851240	395	Toronto	Ontario	H5KG0L	German	person20@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2005	Nora	Scott	6706	2022-09-13	1971-07-06	426885138	825	Ottawa	Ontario	H8KG3L	German	person27@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2006	Joe	Flores	6736	2025-06-12	1991-07-11	626373293	462	Montreal	Quebec	H2KG2L	French	person13@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2007	Steven	Bell	6760	2022-06-04	2002-02-14	751558232	369	Montreal	Quebec	H1KG7L	Australian	person15@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2008	Nayeli	Harris	7633	2022-10-07	1976-05-07	914546857	659	Laval	Quebec	H2KG3L	German	person2@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2009	Noah	Powell	8179	2025-03-28	1961-09-18	341125934	876	Ottawa	Ontario	H9KG7L	Canadian	person39@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2010	Nina	Williams	8280	2023-10-11	1996-05-05	274731623	349	Montreal	Quebec	H0KG7L	British	person44@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2011	Joe	Phillips	8456	2025-12-14	1978-10-11	496423866	905	Laval	Quebec	H8KG7L	Canadian	person38@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2012	Steven	Washington	9388	2022-11-25	1998-09-11	127109103	585	Montreal	Quebec	H9KG8L	French	person26@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2013	Michael	Martin	9508	2022-07-15	2003-06-11	856895891	659	Toronto	Ontario	H3KG4L	Australian	person5@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2014	Natali	Carter	9620	2022-01-01	1986-04-23	101158407	970	Montreal	Quebec	H3KG8L	Australian	person41@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>

2002	John	Gonzalez	6560	2023-10-08	1962-03-18	143748391	228	Toronto	Ontario	H4KG5L	Australian	person19@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2003	Natalia	Lewis	6640	2024-12-27	2003-04-05	373676914	327	Laval	Quebec	H3KG5L	French	person37@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2004	Nia	Murphy	6661	2025-04-17	1967-02-27	912851240	395	Toronto	Ontario	H5KG0L	German	person20@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2005	Nora	Scott	6706	2022-09-13	1971-07-06	426885138	825	Ottawa	Ontario	H8KG3L	German	person27@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2006	Joe	Flores	6736	2025-06-12	1991-07-11	626373293	462	Montreal	Quebec	H2KG2L	French	person13@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2007	Steven	Bell	6760	2022-06-04	2002-02-14	751558232	369	Montreal	Quebec	H1KG7L	Australian	person15@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2008	Nayeli	Harris	7633	2022-10-07	1976-05-07	914546857	659	Laval	Quebec	H2KG3L	German	person2@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2009	Noah	Powell	8179	2025-03-28	1961-09-18	341125934	876	Ottawa	Ontario	H9KG7L	Canadian	person39@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2010	Nina	Williams	8280	2023-10-11	1996-05-05	274731623	349	Montreal	Quebec	H0KG7L	British	person44@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2011	Joe	Phillips	8456	2025-12-14	1978-10-11	496423866	905	Laval	Quebec	H8KG7L	Canadian	person38@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2012	Steven	Washington	9388	2022-11-25	1998-09-11	127109103	585	Montreal	Quebec	H9KG8L	French	person26@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2013	Michael	Martin	9508	2022-07-15	2003-06-11	856895891	659	Toronto	Ontario	H3KG4L	Australian	person5@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2014	Natali	Carter	9620	2022-01-01	1986-04-23	101158407	970	Montreal	Quebec	H3KG8L	Australian	person41@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2015	John	Wright	9671	2022-08-22	1976-01-11	164511484	842	Montreal	Quebec	H6KG8L	French	person10@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2016	Natalya	Adams	9801	2024-10-03	1972-12-20	100886725	106	Ottawa	Ontario	H3KG9L	French	person31@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2017	Nathalia	Sanders	9830	2024-03-06	1976-10-13	412807743	341	Montreal	Quebec	H8KG6L	Canadian	person28@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
2018	Wei	Su	3123	2023-08-30	2023-08-30	5142946093	332	POINTE-CLAIRES	Quebec	H9R 1E8	Canadian	hi.wei.su@gmail.com	<a href="#">Edit</a>	<a href="#">Delete</a>
				yyyy - mm - dd	yyyy - mm - dd								<a href="#">Add</a>	

edit.php?studentID=XXXX&medicareNumber=YYYY

## Edit Student

Student ID	First Name	Last Name	Medicare Number	Medicare Expiry Date	Date of Birth	Phone	Address	City	Province	Postal Code	Citizenship	Email
2001			6349	yyyy - mm - dd	yyyy - mm - dd							

## employees.php

← → ⌂ 🔒 jdc353.encts.concordia.ca/employees.php

Home | Facilities | Persons | Students | Employees | Vaccinations | Infections

### Employees

Employee ID	First Name	Last Name	Medicare Number	Medicare Expiry Date	Date of Birth	Phone	Address	City	Province	Postal Code	Citizenship	Email		
1001	Jack	Alexander	1132	2024-10-27	2001-02-07	997905683	228	Ottawa	Ontario	H4KG8L	American	person32@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1002	Mary	Rogers	1556	2023-05-28	1985-05-14	581046349	514	Laval	Quebec	H0KG4L	American	person33@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1003	Natalya	Carter	1799	2025-11-19	1964-11-20	859210472	930	Montreal	Quebec	H1KG1L	German	person29@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1004	Maria	Evans	1805	2022-09-19	1962-06-19	958736466	611	Montreal	Quebec	H1KG3L	French	person6@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1005	Natalie	Williams	1947	2024-03-13	1978-04-25	821922687	673	Laval	Quebec	H4KG3L	Australian	person40@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1006	Natalya	Baker	2068	2023-03-24	1967-07-05	962355955	973	Toronto	Ontario	H4KG8L	German	person8@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1007	Jack	Roberts	2165	2024-08-14	1998-04-14	648481342	587	Toronto	Ontario	H1KG1L	American	person18@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1008	Nathaly	Howard	2506	2023-02-10	1977-02-19	945519013	588	Laval	Quebec	H1KG4L	Canadian	person23@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1009	Nathan	Kelly	2515	2022-03-05	1976-03-09	278047016	284	Toronto	Ontario	H2KG4L	American	person24@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1010	Steven	Murphy	2941	2023-09-28	1960-09-22	736149025	852	Ottawa	Ontario	H4KG4L	French	person12@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1011	Natali	White	3295	2025-06-12	1990-10-16	303711351	374	Toronto	Ontario	H5KG2L	French	person4@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1012	Julia	Kelly	3667	2024-06-15	1979-07-19	122821961	233	Montreal	Quebec	H5KG8L	Australian	person30@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1013	Nevaeh	Johnson	3682	2024-11-13	1995-06-11	938775297	707	Montreal	Quebec	H7KG1L	Canadian	person35@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1014	Steven	James	3792	2022-01-26	1995-10-23	417204837	772	Laval	Quebec	H4KG6L	American	person3@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>

← → ⌂ 🔒 jdc353.encts.concordia.ca/employees.php

Employee ID	Name	White	2023	2023-06-12	1990-10-10	2023/11/01	2024	TORONTO	Ontario	H4KG8L	French	person4@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1012	Julia	Kelly	3667	2024-06-15	1979-07-19	122821961	233	Montreal	Quebec	H5KG8L	Australian	person30@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1013	Nevaeh	Johnson	3682	2024-11-13	1995-06-11	938775297	707	Montreal	Quebec	H7KG1L	Canadian	person35@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1014	Steven	James	3792	2022-01-26	1995-10-23	417204837	772	Laval	Quebec	H4KG6L	American	person3@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1015	Maria	Russell	3837	2023-07-03	1996-08-23	957211208	959	Montreal	Quebec	H1KG9L	Canadian	person16@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1016	Nora	White	4233	2025-09-09	1994-01-09	785019153	299	Laval	Quebec	H3KG3L	British	person22@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1017	Nancy	Rogers	4628	2023-02-02	1984-06-22	720769265	736	Laval	Quebec	H4KG6L	French	person25@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1018	Nathalie	Walker	4874	2024-07-21	1998-06-18	783500106	920	Montreal	Quebec	H7KG2L	German	person21@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1019	Jack	Robinson	4997	2023-06-26	1992-01-08	581147180	411	Laval	Quebec	H8KG3L	French	person17@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1020	Tom	Phillips	5048	2023-09-26	2002-11-04	218038875	301	Montreal	Quebec	H0KG0L	Canadian	person11@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1021	Nathan	Carter	5400	2024-02-20	1996-05-12	781192683	943	Montreal	Quebec	H9KG5L	Canadian	person42@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1022	Tom	Martinez	5721	2023-03-23	1974-09-01	639881652	717	Laval	Quebec	H8KG0L	German	person11@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1023	Nia	Cook	5728	2024-07-11	1975-04-26	681659378	246	Ottawa	Ontario	H3KG7L	American	person34@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1024	Nathan	Diaz	5860	2022-03-07	2003-08-12	692850501	904	Montreal	Quebec	H9KG8L	British	person36@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1025	Nathalie	Cooper	5905	2023-04-10	1964-02-18	974005556	452	Montreal	Quebec	H6KG5L	Canadian	person45@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1026	Tom	Price	6011	2025-11-24	2000-11-27	596009224	829	Ottawa	Ontario	H7KG4L	Australian	person43@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1027	Joe	Miller	6013	2024-08-22	1988-09-16	295279935	228	Toronto	Ontario	H7KG5L	Canadian	person9@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
1028	Nancy	Cook	6185	2025-07-08	1998-04-11	335217668	753	Montreal	Quebec	H5KG6L	Australian	person14@example.com	<a href="#">Edit</a>	<a href="#">Delete</a>
				yyyy - mm - dd	yyyy - mm - dd								<a href="#">Add</a>	

edit.php?employeeID=XXXX&medicareNumber=YYYY

[Home](#) [Facilities](#) [Persons](#) [Students](#) [Employees](#) [Vaccinations](#) [Infections](#)

## Edit Employee

Employee ID	First Name	Last Name	Medicare Number	Medicare Expiry Date	Date of Birth	Phone	Address	City	Province	Postal Code	Citizenship	Email
1001			1132	yyyy - mm - dd	yyyy - mm - dd							

## vaccinations.php

[Home](#) [Facilities](#) [Persons](#) [Students](#) [Employees](#) [Vaccinations](#) [Infections](#)

## Vaccinations

Medicare Number	First Name	Last Name	Dose Number	Date	Type	Edit	Delete
1556	Mary	Rogers	1	2023-07-13	Sinovac	<a href="#">Edit</a>	<a href="#">Delete</a>
1556	Mary	Rogers	2	2023-08-23	astra	<a href="#">Edit</a>	<a href="#">Delete</a>
1799	Natalya	Carter	1	2023-07-15	Johnson & Johnson	<a href="#">Edit</a>	<a href="#">Delete</a>
1805	Maria	Evans	1	2022-10-08	Moderna	<a href="#">Edit</a>	<a href="#">Delete</a>
2068	Natalya	Baker	1	2022-09-03	Novavax	<a href="#">Edit</a>	<a href="#">Delete</a>
2068	Natalya	Baker	2	2023-05-06	Moderna	<a href="#">Edit</a>	<a href="#">Delete</a>
2506	Nathaly	Howard	1	2022-11-10	Sinovac	<a href="#">Edit</a>	<a href="#">Delete</a>
2515	Nathan	Kelly	1	2021-09-13	Johnson & Johnson	<a href="#">Edit</a>	<a href="#">Delete</a>
2515	Nathan	Kelly	2	2022-07-07	Novavax	<a href="#">Edit</a>	<a href="#">Delete</a>
2941	Steven	Murphy	1	2021-01-03	Novavax	<a href="#">Edit</a>	<a href="#">Delete</a>
3295	Natali	White	1	2021-03-12	Moderna	<a href="#">Edit</a>	<a href="#">Delete</a>
3295	Natali	White	2	2023-07-24	Sinovac	<a href="#">Edit</a>	<a href="#">Delete</a>
3667	Julia	Kelly	1	2022-02-18	Sinovac	<a href="#">Edit</a>	<a href="#">Delete</a>

← → ⌂ jdc353.encs.concordia.ca/vaccinations.php

6640	Natalia	Lewis	2	2023-07-24	Pfizer	<a href="#">Edit</a>	<a href="#">Delete</a>
6661	Nia	Murphy	1	2022-09-17	Novavax	<a href="#">Edit</a>	<a href="#">Delete</a>
6736	Joe	Flores	1	2022-11-03	Moderna	<a href="#">Edit</a>	<a href="#">Delete</a>
6760	Steven	Bell	1	2021-02-03	Novavax	<a href="#">Edit</a>	<a href="#">Delete</a>
6760	Steven	Bell	2	2023-02-10	Novavax	<a href="#">Edit</a>	<a href="#">Delete</a>
7633	Nayeli	Harris	1	2022-01-07	Johnson & Johnson	<a href="#">Edit</a>	<a href="#">Delete</a>
8179	Noah	Powell	1	2023-08-02	Johnson & Johnson	<a href="#">Edit</a>	<a href="#">Delete</a>
8179	Noah	Powell	2	2023-02-26	AstraZeneca	<a href="#">Edit</a>	<a href="#">Delete</a>
8280	Nina	Williams	1	2022-04-02	Pfizer	<a href="#">Edit</a>	<a href="#">Delete</a>
8456	Joe	Phillips	1	2022-03-26	Pfizer	<a href="#">Edit</a>	<a href="#">Delete</a>
9508	Michael	Martin	1	2022-02-26	Johnson & Johnson	<a href="#">Edit</a>	<a href="#">Delete</a>
9671	John	Wright	1	2022-08-05	Novavax	<a href="#">Edit</a>	<a href="#">Delete</a>
9671	John	Wright	2	2022-05-03	AstraZeneca	<a href="#">Edit</a>	<a href="#">Delete</a>
9801	Natalya	Adams	1	2022-04-01	Sinovac	<a href="#">Edit</a>	<a href="#">Delete</a>
9801	Natalya	Adams	2	2023-06-28	AstraZeneca	<a href="#">Edit</a>	<a href="#">Delete</a>
9830	Nathalia	Sanders	1	2022-10-28	AstraZeneca	<a href="#">Edit</a>	<a href="#">Delete</a>
				yyyy - mm - dd <input type="text"/>		<a href="#">Add</a>	

edit.php?medicareNumber=XXXX&doseNumber=Y

← → ⌂ jdc353.encs.concordia.ca/edit.php?medicareNumber=1556&doseNumber=1

Home Facilities Persons Students Employees Vaccinations Infections

## Edit Vaccination

Medicare Number	First Name	Last Name	Dose Number	Date	Type	
1556	Mary	Rogers	1	yyyy - mm - dd <input type="text"/>		<a href="#">Update</a>

infections.php

<a href="#">Home</a>	<a href="#">Facilities</a>	<a href="#">Persons</a>	<a href="#">Students</a>	<a href="#">Employees</a>	<a href="#">Vaccinations</a>	<a href="#">Infections</a>						
----------------------	----------------------------	-------------------------	--------------------------	---------------------------	------------------------------	----------------------------	--	--	--	--	--	--

## Infections

Medicare Number	First Name	Last Name	Infection Number	Date	Type	<a href="#">Edit</a>	<a href="#">Delete</a>
1132	Jack	Alexander	1	2023-04-13	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
1132	Jack	Alexander	2	2023-08-01	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1556	Mary	Rogers	1	2022-07-09	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1556	Mary	Rogers	2	2023-01-01	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1556	Mary	Rogers	3	2023-03-09	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1556	Mary	Rogers	4	2023-08-10	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1556	Mary	Rogers	4	2023-08-23	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1556	Mary	Rogers	5	2023-08-23	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1799	Natalya	Carter	1	2023-08-01	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1799	Natalya	Carter	2	2023-05-26	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
1805	Maria	Evans	1	2022-05-21	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
1947	Natalie	Williams	1	2023-01-04	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
1947	Natalie	Williams	2	2023-05-26	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>

Medicare Number	First Name	Last Name	Infection Number	Date	Type	<a href="#">Edit</a>	<a href="#">Delete</a>
7633	Nayeli	Harris	1	2022-09-02	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
8179	Noah	Powell	1	2022-08-22	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
8179	Noah	Powell	2	2023-01-03	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
8280	Nina	Williams	1	2022-06-19	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
8280	Nina	Williams	2	2023-02-03	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
8280	Nina	Williams	3	2023-07-17	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
8456	Joe	Phillips	1	2023-03-19	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
9508	Michael	Martin	1	2021-03-04	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
9508	Michael	Martin	2	2022-01-27	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
9508	Michael	Martin	3	2023-01-12	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
9620	Natali	Carter	1	2023-07-13	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
9671	John	Wright	1	2022-08-05	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
9671	John	Wright	2	2022-12-22	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
9671	John	Wright	3	2023-07-25	COVID-19	<a href="#">Edit</a>	<a href="#">Delete</a>
9801	Natalya	Adams	1	2023-01-28	SARS-Cov-2 Variant	<a href="#">Edit</a>	<a href="#">Delete</a>
9830	Nathalia	Sanders	1	2022-12-17	Other	<a href="#">Edit</a>	<a href="#">Delete</a>
				yyyy - mm - dd		<input type="button" value="Add"/>	

<edit.php?medicareNumber=XXXX&infectionNumber=Y>

← → ⌂

jd353.encts.concordia.ca/edit.php?medicareNumber=1132&infectionNumber=1

Home Facilities Persons Students Employees Vaccinations Infections

## Edit Infection

Medicare Number	First Name	Last Name	Infection Number	Date	Type	
1132	Jack	Alexander	1	yyyy-mm-dd <input type="button" value="Calendar"/>		<input type="button" value="Update"/>

## Contributions

### **Yifan Wu**

Set up the website on the server

Design and Implement GUI with stylings

Design and implement the PHP files for some functions

Design and Implement the signup & login system

Implement the SQL queries

Provided some SQL data

Handle the normalization

Prepare the report

### **Wei Su**

Set up the website on the server

Design the web interface

Design and Implement GUI with stylings

Design and implement the PHP files for some functions,

Design and Implement the signup & login system

Implement the SQL queries

Provided some SQL data

Prepare the report

### **Mikelson Benjamin**

Design and Implement GUI with stylings

Design and implement the PHP files for some functions

Design and Implement the signup & login system

Implement the SQL queries

Provided some SQL data

Prepare the report

### **Yujing Wang**

Design the E/R Diagram

Do the Conversion and relevant FDs

Create tables and prepare necessary data for testing

Prepare the report

Implement the SQL queries

Provided some SQL data

