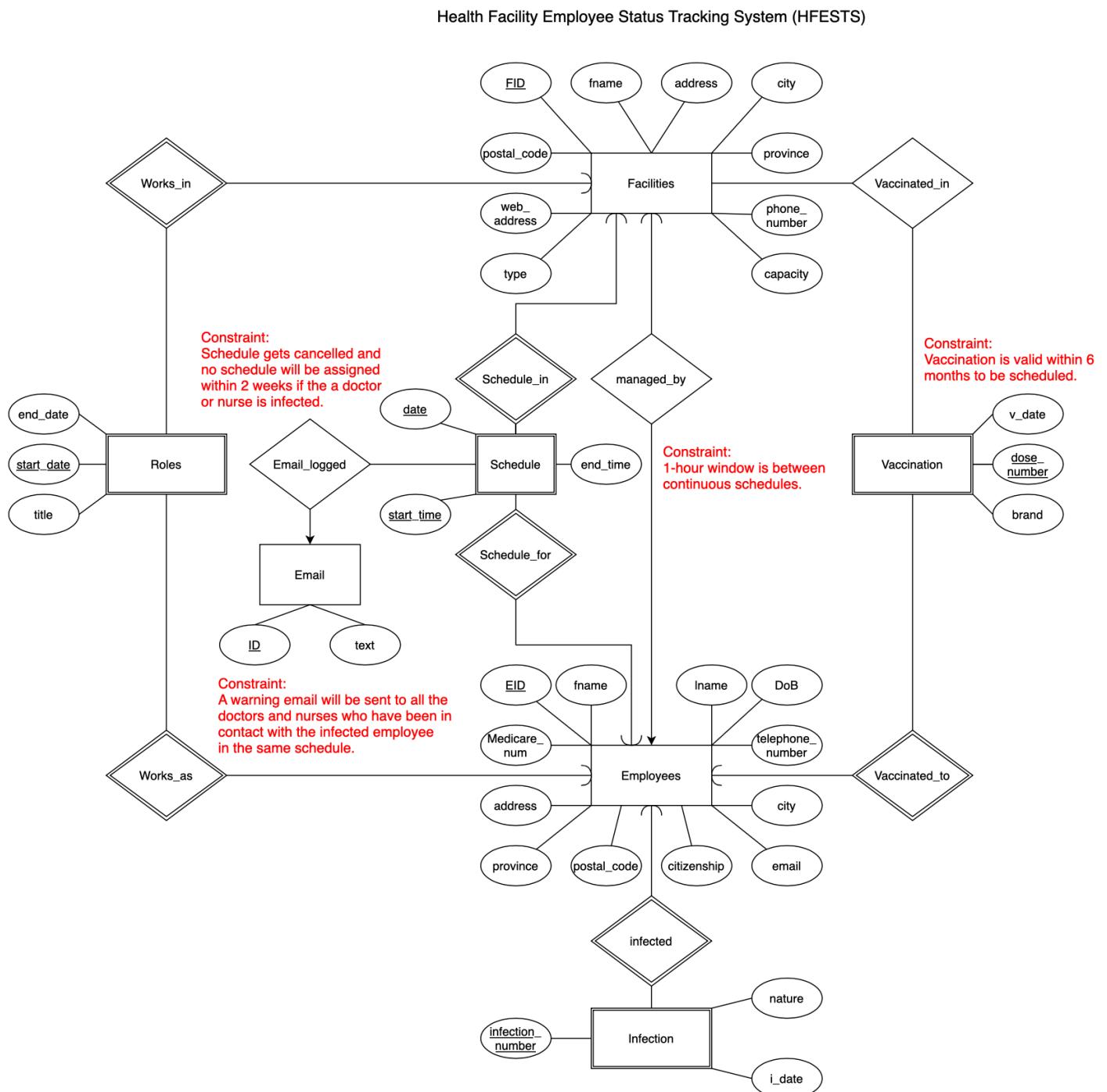


COMP 5531 Final Project Report

1. E/R diagram



2. Constraints

In the diagram, mark or express various constraints (keys, functional dependencies, cardinalities of the relationships, etc.) Identify any constraints that are not captured by the E/R diagram.

All major constraints are displayed on E/R diagram.

3. Relational Schemas

Convert your E/R diagram into a relational database schema. Make refinements to the DB schema if necessary. Identify various integrity constraints such as primary keys, foreign keys, functional dependencies, and referential constraints. Make sure that your database schema is at least in 3NF.

Facilities(FID, FName, Address, City, Province, Postal_Code, Phone_Number, Web_Address, FType, Capacity)

- Primary key: FID
- FID → FName, Address, City, Province, Postal_Code, Phone_Number, Web_Address, FType, Capacity

Employees(EID, FName, LName, DoB, Medicare_Number, Telephone_Number, address, city, Province, Postal_code, Citizenship, Email)

- Primary key: EID
- Unique Constraint: Medicare_Number
- EID → FName, LName, DoB, Medicare_Number, Telephone_Number, address, city, Province, Postal_code, Citizenship, Email

Vaccination(EID, Dose_Number, Brand, V_date)

- Primary key: EID, Dose_Number
- Foreign Key: EID references Employees(EID)
- Foreign Key: FID references Facilities(FID)
- EID, Dose_Number → Brand, V_date

Vaccinated_in(FID, EID, Dose_Number)

- Primary key: FID, EID, Dose_Number
- Foreign Key: FID references Facilities(FID)
- Foreign Key: Dose_Number references Vaccination(EID, Dose_Number)
- No applicable FD

Infection(EID, Infection_Number, Nature, i_date)

- Primary key: EID, FID, Infection_Number
- Foreign Key: EID references Employees(EID)
- Foreign Key: FID references Facilities(FID)

- EID, Infection_Number → Nature, i_date

Roles(EID, FID, title, Start_Date, end_date)

- Primary key: EID, FID, Start_Date
- Foreign Key: EID references Employees(EID)
- Foreign Key: FID references Facilities(FID)
- EID, FID, Start_Date → title, end_date

Managed_by(FID, EID)

- Primary key:FID
- Foreign Key: FID references Facilities(FID)
- Foreign Key: EID references Employees(EID)
- FID → EID

Schedule(FID, EID, date, start_time, end_time)

- Primary key: EID, FID, date, start_time
- Foreign Key: EID references Employees(EID)
- Foreign Key: FID references Facilities(FID)
- FID, EID, date, start_time → end_time

Email(ID, text)

- Primary key: ID
- ID → text

Email_logged(FID, EID, date, start_time, ID)

- Primary key: EID, FID, date, start_time
- Foreign Key: EID, FID, date, start_time references Schedule(EID, FID, date, start_time)
- Foreign Key: ID references Email(ID)
- FID, EID, date, start_time → ID

4. BCNF

Is all your relations in the database in BCNF? (Explain which ones and why not).

Yes. All the relations in our schema are in BCNF, as there are no functional dependencies where the left-hand side is not a superkey.

5. Non-BCNF

For any relation in your database, if it is not in BCNF, then show that it is in 3NF.

All relationships are BCNF as explained above.

6. Trigger

Create at least one trigger to execute some of the requirements specified in the description above.

Constraint: 1-hour window is set between continuous schedules.

```
DELIMITER //

CREATE TRIGGER check_schedule_gap
BEFORE INSERT ON Schedule
FOR EACH ROW
BEGIN
    DECLARE previous_end_time TIME;

    SELECT MAX(end_time) INTO previous_end_time
    FROM Schedule
    WHERE EID = NEW.EID AND date = NEW.date;

    IF previous_end_time IS NOT NULL THEN
        IF NEW.start_time < ADDTIME(previous_end_time, '01:00:00') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'at least one hour should be the duration between
the first schedule and the second one.';
        END IF;
    END IF;
END; //

DELIMITER ;
```

Constraint: Schedule gets cancelled and no schedule will be assigned within 2 weeks if the doctor or nurse is infected.

To implement this requirement, we need 2 triggers. The 1st trigger will prevent scheduling of infected doctors and nurses within 2 weeks. The 2nd trigger will cancel the existing schedules for infected doctors and nurses within 2 weeks.

Trigger #1: prevent scheduling infected doctors and nurses

```
DELIMITER //
CREATE TRIGGER prevent_infected_scheduling
BEFORE INSERT ON Schedule
FOR EACH ROW
BEGIN
    DECLARE infection_date DATE;
    DECLARE employee_role VARCHAR(255);
```

```

SELECT i_date INTO infection_date
FROM Infection
WHERE EID = NEW.EID;

SELECT title INTO employee_role
FROM Roles
WHERE EID = NEW.EID AND FID = NEW.FID;

IF infection_date IS NOT NULL AND (employee_role = 'Doctor' OR
employee_role = 'Nurse') THEN
    IF NEW.date BETWEEN infection_date AND DATE_ADD(infection_date, INTERVAL 2
WEEK) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Infected doctors and nurses cannot be assigned to a
schedule within the next two weeks.';
    END IF;
END IF;
END;
//  

DELIMITER ;

```

Trigger #2: cancel existing schedules for infected doctors and nurses

```

DELIMITER //
CREATE TRIGGER cancel_infected_schedules
AFTER UPDATE ON Infection
FOR EACH ROW
BEGIN
    IF OLD.i_date IS NULL AND NEW.i_date IS NOT NULL THEN

        DELETE S
        FROM Schedule S
        JOIN Role R ON S.EID = R.EID AND S.FID = R.FID
        WHERE S.EID = NEW.EID AND S.date BETWEEN NEW.i_date AND
DATE_ADD(NEW.i_date, INTERVAL 2 WEEK)
            AND (R.title = 'Doctor' OR R.title = 'Nurse');

    END IF;
END;
//  

DELIMITER ;

```

Constraint: Vaccination is valid within 6 months to be scheduled.

Only the employees who have been vaccinated within the last 6 months can be scheduled.

```

DELIMITER //
CREATE TRIGGER prevent_unvaccinated_scheduling
BEFORE INSERT ON Schedule
FOR EACH ROW
BEGIN
    DECLARE latest_vaccine_date DATE;

    SELECT MAX(V_date) INTO latest_vaccine_date
    FROM Vaccination
    WHERE EID = NEW.EID;

    IF latest_vaccine_date IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Employees must be vaccinated to be scheduled for work.';

    ELSEIF NEW.date > DATE_ADD(latest_vaccine_date, INTERVAL 6 MONTH) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Employees must be vaccinated within the past 6 months to be scheduled for work.';

    END IF;
END;
//
DELIMITER ;

```

Constraint: A warning email will be sent to all the doctors and nurses who have been in contact with the infected employee in the same schedule.

```

DELIMITER //
CREATE TRIGGER notify_contacted_employees
AFTER INSERT ON Infection
FOR EACH ROW
BEGIN
    -- Declare variables to hold the EID and email address of the contacted employees
    DECLARE contacted_eid INT;
    DECLARE contacted_email VARCHAR(255);

    -- Initialize the finished variable
    DECLARE finished INT DEFAULT 0;

    -- Declare a cursor to fetch doctors and nurses who have been in contact with the infected employee
    DECLARE contacted_employee_cursor CURSOR FOR
        SELECT DISTINCT E.EID, E.Email
        FROM Employees E
        JOIN Schedule S1 ON E.EID = S1.EID

```

```

JOIN Schedule S2 ON S1.date = S2.date AND S1.FID = S2.FID
JOIN Role R ON E.EID = R.EID AND S1.FID = R.FID
WHERE S2.EID = NEW.EID AND E.EID <> NEW.EID
      AND (R.title = 'Doctor' OR R.title = 'Nurse');

-- Declare a handler to exit the loop when the cursor runs out of rows
DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1;

-- If the employee is updated to be infected (i_date was null and is now
not null)
IF OLD.i_date IS NULL AND NEW.i_date IS NOT NULL THEN
    -- Open the cursor
    OPEN contacted_employee_cursor;

    -- Loop through the contacted employees
    contact_loop: LOOP
        -- Fetch the next contacted employee
        FETCH contacted_employee_cursor INTO contacted_eid, contacted_email;

        -- Exit the loop if no more rows are available
        IF finished = 1 THEN
            LEAVE contact_loop;
        END IF;

        -- Call the send_warning_email procedure with the contacted employee's
        EID and email text
        CALL send_warning_email(contacted_eid, 'Warning: You have been in
contact with an infected employee.);

        END LOOP contact_loop;

        -- Close the cursor
        CLOSE contacted_employee_cursor;
    END IF;
END;
// 
DELIMITER ;

```

Constraint: Log the schedules of employees and send them via email.

```

DELIMITER //
CREATE TRIGGER `log_schedule_email` AFTER INSERT ON `Schedule` FOR EACH ROW
BEGIN
    -- Declare necessary variables
    DECLARE employee_fname VARCHAR(255);
    DECLARE employee_lname VARCHAR(255);
    DECLARE employee_email VARCHAR(255);
    DECLARE email_subject VARCHAR(255);

```

```

DECLARE email_body TEXT;
DECLARE inserted_email_id INT;
DECLARE start_of_week DATE;
DECLARE end_of_week DATE;

-- Fetch employee
SELECT E.FName, E.LName, E.Email
INTO employee_fname, employee_lname, employee_email
FROM Employees E
WHERE E.EID = NEW.EID;

-- Set start and end of the week
SET start_of_week = NEW.date - INTERVAL WEEKDAY(NEW.date) DAY;
SET end_of_week = NEW.date + INTERVAL (6 - WEEKDAY(NEW.date)) DAY;

-- Check if an email for the given employee and week already exists
SELECT E.ID INTO inserted_email_id
FROM Email E
JOIN Email_logged EL ON E.ID = EL.ID
WHERE EL.EID = NEW.EID AND EL.date BETWEEN start_of_week AND end_of_week
ORDER BY E.ID DESC LIMIT 1;

-- Construct the email subject
SET email_subject = CONCAT('Subject: Schedule for Mon-',
DATE_FORMAT(start_of_week, '%Y-%m-%d'), ' to Sun-', DATE_FORMAT(end_of_week,
'%Y-%m-%d'));

-- Fetch schedules for the week and construct the email body
SET email_body = CONCAT('Name: ', employee_fname, ' ', employee_lname,
'\nEmail: ', employee_email, '\n\n');

SELECT
    GROUP_CONCAT(CONCAT('Facility: ', F.FName, '\n(', DAYNAME(S.date), ')',
Start: ', S.start_time, ' - End: ', S.end_time, '\n\n') ORDER BY S.date,
S.start_time SEPARATOR '')
    INTO @schedule_details
FROM Schedule S
JOIN Facilities F ON S.FID = F.FID
WHERE S.EID = NEW.EID AND S.date BETWEEN start_of_week AND end_of_week;

SET email_body = CONCAT(email_subject, '\n\n', email_body,
@schedule_details);

IF inserted_email_id IS NULL THEN
    -- If no email exists for the week, create a new one

    -- Insert the email into the Email table
    INSERT INTO Email (text) VALUES (email_body);
    SET inserted_email_id = (SELECT MAX(ID) FROM Email);
ELSE

```

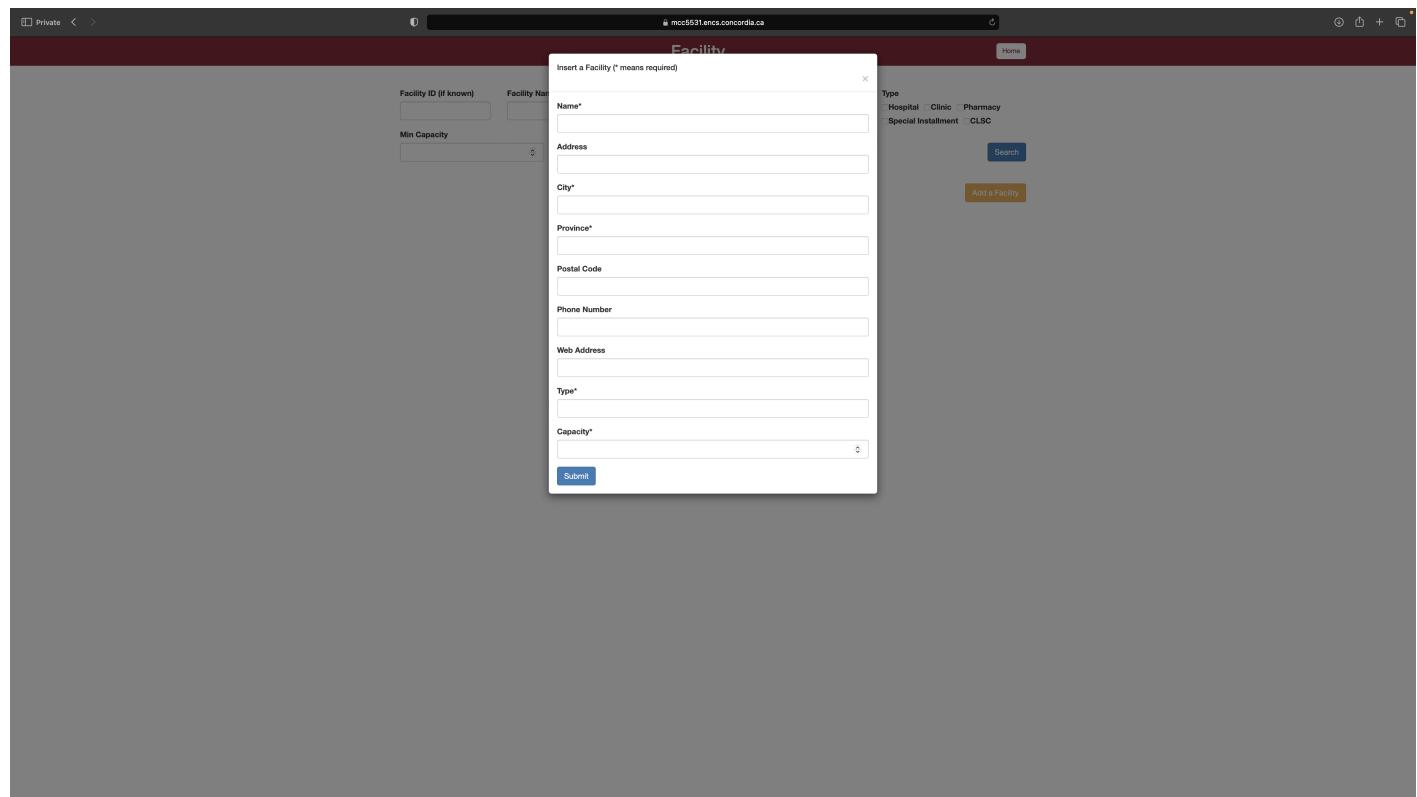
```
-- If an email for the week already exists, update the email body with  
the new schedule  
  
-- Update email with new schedule  
UPDATE Email SET text = email_body WHERE ID = inserted_email_id;  
END if;  
  
-- Insert the record into the Email_logged table  
INSERT INTO Email_logged (FID, EID, date, start_time, ID) VALUES (NEW.FID,  
NEW.EID, NEW.date, NEW.start_time, inserted_email_id);  
END; //  
DELIMITER ;
```

Formulate and evaluate SQL DDL and DML commands (Q1-20)

1. Create/Delete/Edit/Display a Facility (using SQL or web interface).

Create a facility:

```
INSERT INTO Facilities (FID, FName, Address, City, Province, Postal_Code,  
Phone_Number, Web_Address, FType, Capacity)  
VALUES (1, 'Facility 1', '123 Main St', 'Toronto', 'Ontario', 'M1M 1M1',  
'555-123-5678', 'www.facility1.com', 'Type A', 100);
```



Delete a facility:

```
DELETE FROM Facilities WHERE FID = 1;
```

The screenshot shows a web-based facility management system. At the top, there are search fields for Facility ID (1003) and Facility Name (Hospital Maisonneuve Rosemont). Below these are dropdown menus for City (Toronto, Montreal, Vancouver), Province (ON, BC, QC), and Type (Hospital, Clinic, Pharmacy, Special Installment, CISSC). There are also input fields for Min Capacity and Max Capacity. A 'Search' button and an 'Add a Facility' link are visible. The main content area displays a table of facilities with columns: FID, Name, Address, City, Province, Postal_Code, Phone_Number, Web, Type, Capacity, and Action. One row is selected for Hospital Maisonneuve Rosemont. The 'Action' column contains 'Delete' and 'Update' buttons. A modal dialog box is overlaid on the page, asking 'Are you sure you want to delete the facility with FID: 1003?' with 'Cancel' and 'OK' buttons.

FID	Name	Address	City	Province	Postal_Code	Phone_Number	Web	Type	Capacity	Action
1003	Hospital Maisonneuve Rosemont	7085 Hutchison St	Montreal	QC	H3N 1Y9	514-273-9591	www.mrshospital.com	hospital	400	<button>Delete</button> <button>Update</button>

Edit a facility:

```
UPDATE Facilities
SET Name = 'Updated Facility Name', Address = '456 Updated St', Phone_Number
= '555-567-1234', Capacity = 200
WHERE FID = 1;
```

The screenshot shows a web browser window with a modal dialog titled "Update Facility". The modal contains fields for Facility ID (FID), Facility Name, Address, City, Province, Postal Code, Phone Number, Web Address, Type, and Capacity. The FID field is set to 1003. The Facility Name is "Hospital Maisonneuve Rosemont". The Address is "7085 Hutchison St". The City is "Montreal". The Province is "QC". The Postal Code is "H3N 1Y9". The Phone Number is "514-273-8591". The Web Address is "www.mrhospital.com". The Type is "hospital". The Capacity is "400". There are "Save changes" and "Close" buttons at the bottom of the modal.

Display a facility:

```
SELECT * FROM Facilities WHERE FID = 1;
```

The screenshot shows a web browser window with a table displaying facility information. The columns include FID, Name, Address, City, Province, Postal Code, Phone Number, Web, Type, Capacity, and Action. One row is selected, showing FID 1003, Name "Hospital Maisonneuve Rosemont", Address "7085 Hutchison St", City "Montreal", Province "QC", Postal Code "H3N 1Y9", Phone Number "514-273-8591", Web "www.mrhospital.com", Type "hospital", Capacity "400", and Action buttons for "Delete" and "Update". At the top of the page, there is a search bar and a "Facility" button. Below the table, there is a "Add a Facility" button.

2. Create/Delete/Edit/Display an employee (using SQL or web interface).

Create an employee:

```
INSERT INTO Employees (EID, FName, LName, DoB, Medicare_Number,  
Telephone_Number, address, city, Province, Postal_code, Citizenship, Email)  
VALUES (1, 'John', 'Doe', '1990-01-01', '1234567890', '555-000-1111', '123  
Main St', 'Toronto', 'Ontario', 'M1M 1M1', 'Canada', 'johndoe@email.com');
```

Delete an employee:

```
DELETE FROM Employees WHERE EID = 1;
```

The screenshot shows a web application titled "Employee" with a search interface. At the top, there are input fields for "Employee ID (if known)", "Employee First Name (if known)", "Employee Last Name (if known)", "Facility ID (if known)", and "Facility Name (if known)". Below these are dropdown menus for "Role" (Nurse, Doctor, Pharmacist) and "Status" (Receptionist, Administrative Personnel, Security Personnel, Regular Employee). There are also date pickers for "Start Date" (04/11/2023) and "End Date" (04/11/2023), and a "Search" button. A "Add an Employee" button is located at the bottom right of the search area. Below the search area is a table with columns: ID, Name, LName, DOb, Medicare_Number, Telephone_Number, Address, City, Province, Postal_Code, Citizenship, Email, RID, Facility_Name, Title, Start_Date, End_Date, and Action. One row is visible, showing an employee named "Test" with ID 12345, born 1990-04-11, with various contact and employment details. To the right of the table are "Delete" and "Update" buttons. A modal dialog box is displayed in the center, stating "Employee deleted successfully!" with a "Close" button.

Edit an employee:

```
UPDATE Employees
SET FName = 'Jane', LName = 'Doe', Telephone_Number = '555-000-222',
Citizenship = 'USA'
WHERE EID = 1;
```

Employee

Update Employee

Employee ID (if known)	Employee F
Role	Nurse Doctor Pharmacist Receptionist Administrative Personnel Security Personnel Regular Employee
First Name	Lee
Last Name	Kim
Date of Birth	1982-05-22
Medicare Number	814927360
Telephone Number	604-924-7731
Address	2390 West 10th Ave
City	Employee updated successfully!
Province	BC
Postal Code	V6K 2J5
Citizenship	Canada
Email	larkim82@gmail.com

Add an Employee

ID	First Name	Last Name	Start Date	End Date	Action
123010	Randall	Terry	1970-01-01	2020-06-30	Delete Update
123009	Lindsey	Karen	1980-01-01	2020-06-30	Delete Update
123008	Diane	Perry	1981-01-01	2020-06-30	Delete Update
123007	Avery	Lee	1980-01-01	2020-06-30	Delete Update
123006	Diane	Perry	1981-01-01	2020-06-30	Delete Update
123005	Jessica	Kim	1984-01-01	2020-06-30	Delete Update
123004	Sophie	Lee	1985-01-01	2020-06-30	Delete Update
123003	Noelle	Perry	1981-01-01	2020-06-30	Delete Update
123002	Ava	Lee	1987-01-01	2020-06-30	Delete Update
123001	Ella	Kim	1980-01-01	2020-06-30	Delete Update
123000	Ella	Kim	1980-01-01	2020-06-30	Delete Update

Display an employee:

```
SELECT * FROM Employees WHERE EID = 1;
```

Employee

Employee ID (if known)	Employee First Name (if known)	Employee Last Name (if known)	Facility ID (if known)	Facility Name (if known)
123001	Eddy	Wong		
Role	Start Date	End Date		
Nurse Doctor Pharmacist Receptionist Administrative Personnel Security Personnel Regular Employee	04/17/2023	04/11/2023		
Address				
City				
Province				
Postal Code				
Citizenship				
Email				
FID	Facility Name	Title	Start Date	End Date
123001	Toronto General Hospital	Administrative personnel	2010-10-01	2020-01-31

3. Create /Delete/Edit/Display a vaccination

Create a vaccination:

```
INSERT INTO Vaccination (EID, Dose_Number, Brand, V_date)
VALUES (1, 1, 'Pfizer', '2022-01-01');
```

Delete a vaccination:

```
DELETE FROM Vaccination WHERE EID = 1 AND Dose_Number = 1;
```

Edit a vaccination:

```
UPDATE Vaccination
SET Brand = 'Moderna', V_date = '2022-02-01'
WHERE EID = 1 AND Dose_Number = 1;
```

Display a vaccination:

```
SELECT * FROM Vaccination WHERE EID = 1 AND Dose_Number = 1;
```

4. Create/Delete/Edit/Display an infection

Create an infection:

```
INSERT INTO Infection (EID, Infection_Number, Nature, I_date)
VALUES (1, 1, 'COVID-19', '2022-03-01');
```

Delete an infection:

```
DELETE FROM Infection WHERE EID = 1 AND Infection_Number = 1;
```

Edit an infection:

```
UPDATE Infection
SET Nature = 'Flu', I_date = '2022-04-01'
WHERE EID = 1 AND Infection_Number = 1;
```

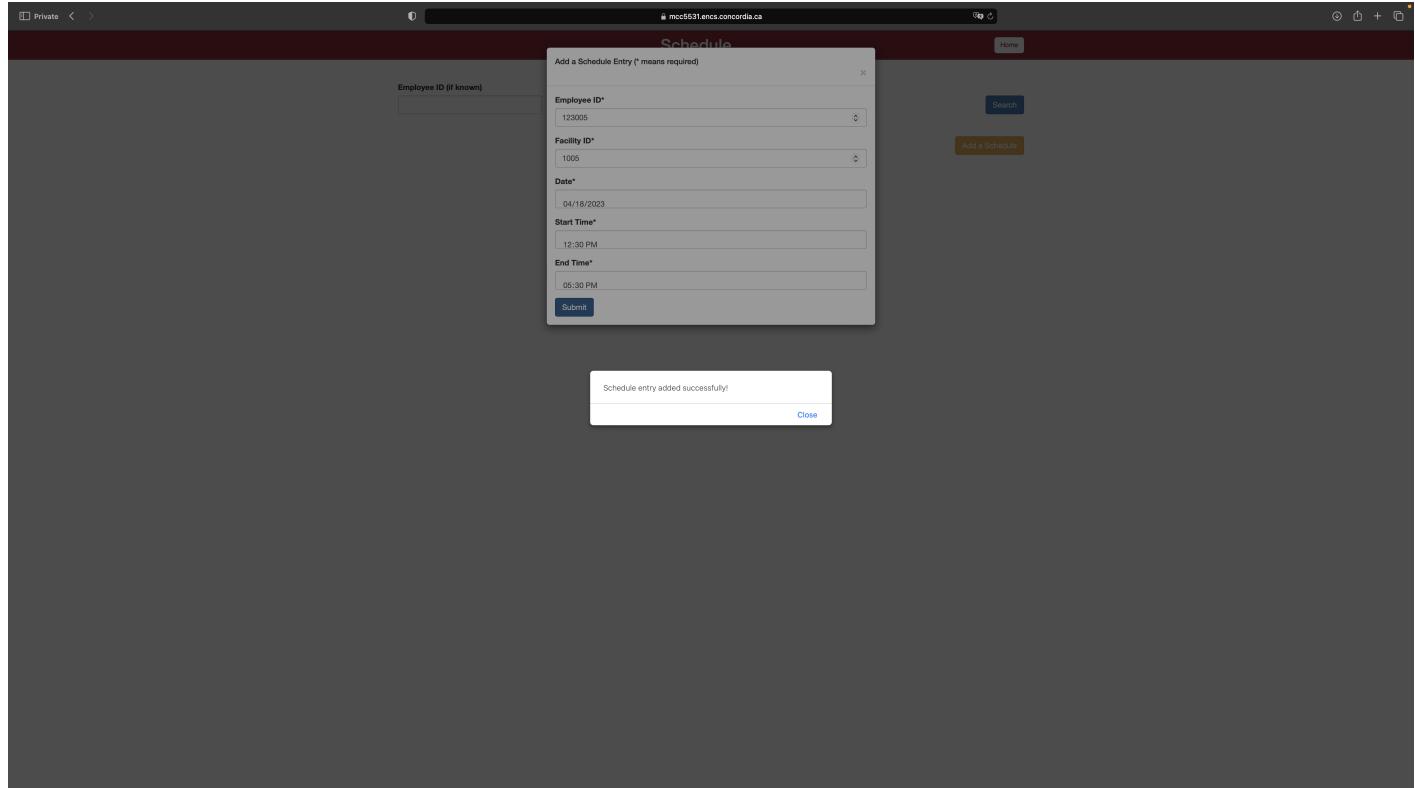
Display an infection:

```
SELECT * FROM Infection WHERE EID = 1 AND Infection_Number = 1;
```

5. Assign/Delete/Edit schedule for an employee (attempt to schedule a conflicting assignment for an employee)

Assign schedule for an employee:

```
INSERT INTO Schedule (FID, EID, date, start_time, end_time)
VALUES (1, 1, '2023-04-10', '09:00:00', '17:00:00');
```



Delete schedule for an employee:

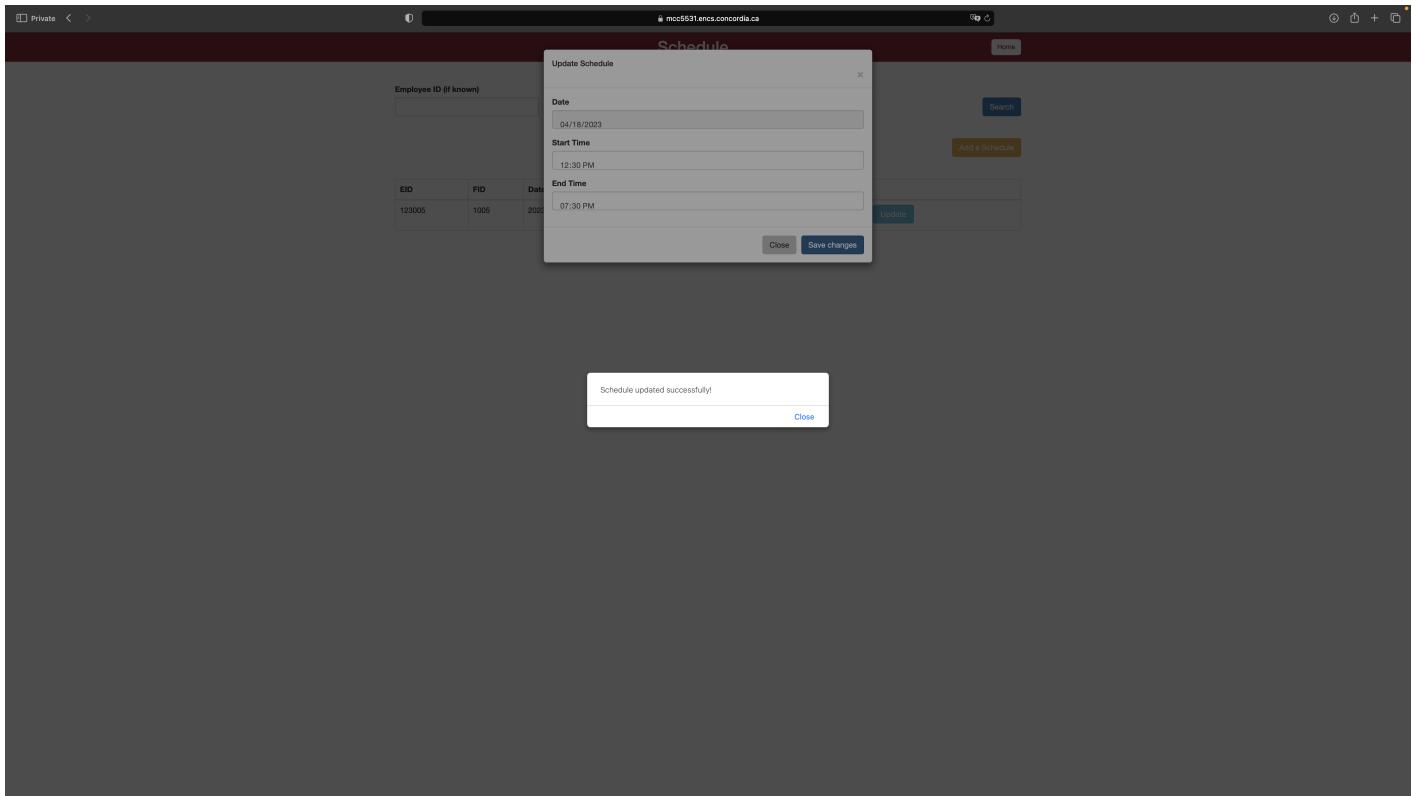
```
DELETE FROM Schedule WHERE FID = 1 AND EID = 1 AND date = '2023-04-10' AND start_time = '09:00:00';
```

The screenshot shows a web browser window titled "Schedule". At the top, there are input fields for "Employee ID (if known)" (123005), "Facility ID (if known)", and "Date (if known)" (04/11/2023). A "Search" button is to the right of the date field. Below the search bar is a small "Add a Schedule" button. The main content area is a table with columns: EID, FID, Date, Start_Time, End_Time, and Action. The table contains several rows of data. A modal dialog box is overlaid on the table, containing the following text: "Are you sure you want to delete the schedule for EID: 123005, FID: 1005, Date: 2023-04-18, Start Time: 12:30:00, End Time: 19:30:00?". It includes "Cancel" and "OK" buttons.

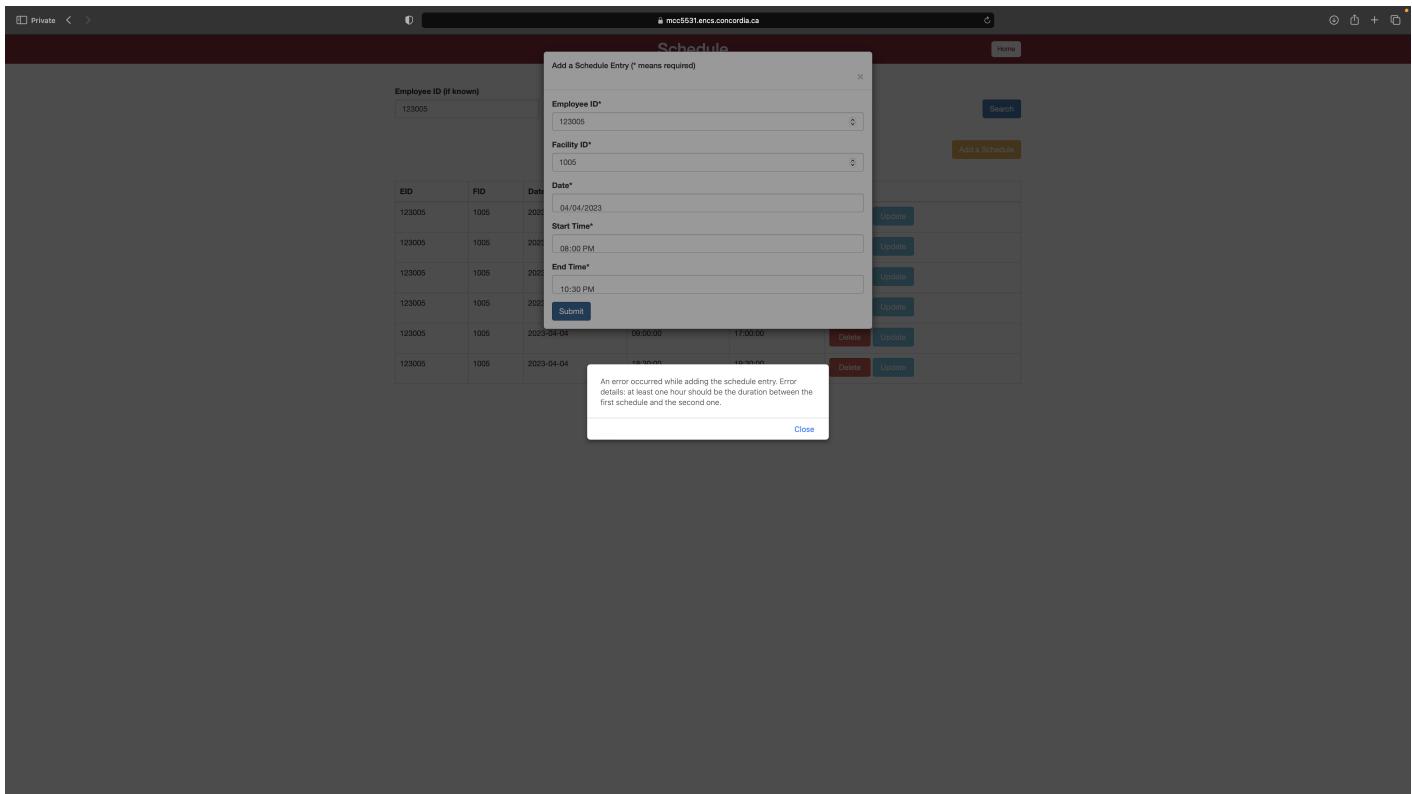
EID	FID	Date	Start_Time	End_Time	Action
123005	1005	2023-03-09	10:14:00	12:00:00	<button>Delete</button> <button>Update</button>
123005	1005	2023-03-14	11:00:00	13:00:00	<button>Delete</button> <button>Update</button>
123005	1005	2023-03-15	11:00:00	13:00:00	<button>Delete</button> <button>Update</button>
123005	1005	2023-04-03	09:00:00	17:00:00	<button>Delete</button> <button>Update</button>
123005	1005	2023-04-04	09:00:00	17:00:00	<button>Delete</button> <button>Update</button>
123005	1005	2023-04-04	12:30:00	19:30:00	<button>Delete</button> <button>Update</button>
123005	1005	2023-04-18			<button>Delete</button> <button>Update</button>

Edit schedule for an employee:

```
SET start_time = '10:00:00', end_time = '18:00:00'  
WHERE FID = 1 AND EID = 1 AND date = '2023-04-10' AND start_time =  
'09:00:00';
```



Attempt to schedule a conflicting assignment:



6. Get all details of the facilities in the system. Details include the facility's name, address, city, province, postal-code, phone number, web address, types, capacity, general manager's name and number of employees currently working for the facility. Results should be displayed sorted in

ascending order by province, then by city, then by type, then by number of employees currently working for the facility.

```
SELECT
    f.FName,
    f.Address,
    f.City,
    f.Province,
    f.Postal_Code,
    f.Phone_Number,
    f.Web_Address,
    f.FType,
    f.Capacity,
    CONCAT(e.FName, ' ', e.LName) AS ManagerName,
    COUNT(DISTINCT r.EID) AS NumEmployees
FROM
    Facilities f
        LEFT JOIN
    Managed_by m ON f.FID = m.FID
        LEFT JOIN
    Employees e ON m.EID = e.EID
        LEFT JOIN
    Roles r ON f.FID = r.FID AND r.end_date IS NULL
GROUP BY f.FID , e.FName , e.LName
ORDER BY f.Province ASC , f.City ASC , f.FType ASC , NumEmployees ASC;
```

7. Get details of all the employees currently working in a specific facility. Details include employees's first-name, last-name, start date of work, date of birth, Medicare card number, telephone-number, address, city, province, postal-code, citizenship, and email address. Results should be displayed sorted in ascending order by role, then by first name, then by last name.

```
SELECT
    f.FID,
    e.FName,
    e.LName,
    r.Start_Date,
    e.DoB,
    e.Medicare_Number,
    e.Telephone_Number,
    e.Address,
    e.City,
    e.Province,
    e.Postal_code,
    e.Citizenship,
    e.Email
FROM
```

```

Employees e
    JOIN
Roles r ON e.EID = r.EID
    JOIN
Facilities f ON f.FID = r.FID
WHERE
    r.end_date IS NULL
GROUP BY f.FID , e.EID
ORDER BY f.FID ASC , r.title ASC , e.FName ASC , e.LName ASC;

```

8. For a given employee, get the details of all the schedules she/he has been scheduled during a specific period of time. Details include facility name, day of the year, start time and end time. Results should be displayed sorted in ascending order by facility name, then by day of the year, then by start time.

```

SELECT
    f.FName AS Facility,
    e.FName,
    e.LName,
    DAYOFYEAR(s.date) AS 'Day of the year',
    s.start_time,
    s.end_time
FROM
    Facilities f
        JOIN
    Schedule s ON f.FID = s.FID
        JOIN
    Employees e ON s.EID = e.EID
WHERE
    s.EID = 123021
        AND s.date BETWEEN '2023-01-01' AND '2023-12-31'
ORDER BY f.FName ASC , DAYOFYEAR(s.date) ASC , s.start_time ASC;

```

9. Get details of all the doctors who have been infected by COVID-19 in the past two weeks. Details include the doctor's first-name, last-name, date of infection, and the name of the facility that the doctor is currently working for. Results should be displayed sorted in ascending order by the facility name, then by the first-name of the doctor.

```

SELECT
    e.EID, e.FName, e.LName, i.I_date, f.FName
FROM
    Employees e
        LEFT JOIN
    Infection i ON e.EID = i.EID

```

```

        JOIN
    Roles r ON e.EID = r.EID AND r.end_date IS NULL
        JOIN
    Facilities f ON r.FID = f.FID
WHERE
    Nature = 'COVID-19'
        AND r.Title = 'Doctor'
        AND i.I_date BETWEEN DATE_SUB(NOW(), INTERVAL 2 WEEK) AND NOW()
ORDER BY f.FName ASC , e.FName ASC;

```

10. List the emails generated by a given facility. The result should be displayed in ascending order by the date of the emails.

```

SELECT Facilities.FID, Email.ID AS Email_ID, Email_logged.EID AS Receiver,
Email.text
FROM Email_logged
JOIN Facilities ON Email_logged.FID = Facilities.FID
JOIN Email ON Email_logged.ID = Email.ID
WHERE Email_logged.FID = 1005
GROUP BY Email_logged.FID, Email_logged.EID, Email.ID
ORDER BY Email_logged.date ASC;

```

11. For a given facility, generate a list of all the doctors and nurses who have been on schedule to work in the last two weeks. The list should include first-name, last-name, and role. Results should be displayed in ascending order by role, then by first name. *FID=1005

```

SELECT DISTINCT E.FName, E.LName, R.Title
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE R.FID = 1005
AND R.title IN ('Doctor', 'Nurse')
AND S.date BETWEEN DATE_SUB(CURRENT_DATE, INTERVAL 2 WEEK) AND CURRENT_DATE
ORDER BY R.title ASC, E.FName ASC;

```

12. For a given facility, give the total hours scheduled for every role during a specific period. Results should be displayed in ascending order by role. *FID 1005 TIME 2023-03-01 TO 2023-04-01

```

SELECT R.title, SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS
Total_Hours
FROM Roles R
JOIN Schedule S ON R.EID = S.EID

```

```

WHERE R.FID = 1005
AND S.date BETWEEN '2023-03-01' AND '2023-04-01'
GROUP BY R.title
ORDER BY R.title ASC;

```

13. For every facility, provide the province where the facility is located, the facility name, the capacity of the facility, and the total number of employees in the facility who have been infected by COVID-19 in the past two weeks. The results should be displayed in ascending order by province, then by the total number of employees infected.

```

SELECT F.Province, F.FName, F.Capacity, COUNT(I.EID) AS Infected_Employees
FROM Facilities F
LEFT JOIN Roles R ON F.FID = R.FID
LEFT JOIN Infection I ON R.EID = I.EID
AND I.i_date BETWEEN DATE_SUB(CURRENT_DATE, INTERVAL 2 WEEK) AND CURRENT_DATE
AND I.Nature = 'COVID-19'
GROUP BY F.FID, F.Province, F.FName, F.Capacity
ORDER BY F.Province ASC, Infected_Employees ASC;

```

14. For every doctor who is currently working in the province of "Québec", provide the doctor's first-name, last-name, the city of residence of the doctor, and the total number of facilities the doctor is currently working for. Results should be displayed in ascending order by city, then in descending order by total number of facilities.

```

SELECT E.FName, E.LName, E.city, COUNT(R.FID) as Total_Facilities
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Facilities F ON R.FID = F.FID
WHERE R.title = 'Doctor'
AND F.Province = 'QC'
AND R.end_date IS NULL
GROUP BY E.EID, E.FName, E.LName, E.city
ORDER BY E.city ASC, Total_Facilities DESC;

```

15. Get details of the nurse(s) who is/are currently working and has the highest number of hours scheduled in the system since they started working as a nurse. Details include first-name, last-name, first day of work as a nurse, date of birth, email address, and total number of hours scheduled.

```

SELECT E.FName, E.LName, MIN(R.Start_Date) AS First_Day_Work, E.DoB, E.Email,
SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS Total_Hours

```

```

FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE R.title = 'Nurse'
AND R.end_date IS NULL
GROUP BY E.EID, E.FName, E.LName, E.DoB, E.Email
HAVING Total_Hours = (
    SELECT MAX(Sub.Total_Hours)
    FROM (
        SELECT E2.EID, SUM(TIMESTAMPDIFF(HOUR, S2.start_time, S2.end_time))
    AS Total_Hours
        FROM Employees E2
        JOIN Roles R2 ON E2.EID = R2.EID
        JOIN Schedule S2 ON E2.EID = S2.EID
        WHERE R2.title = 'Nurse'
        AND (R2.end_date IS NULL OR R2.end_date > CURRENT_DATE)
        GROUP BY E2.EID
    ) Sub
)
ORDER BY E.FName, E.LName;

```

16. Get details of the nurse(s) or the doctor(s) who are currently working and has been infected by COVID-19 at least three times. Details include first-name, last-name, first day of work as a nurse or as a doctor, role (nurse/doctor), date of birth, email address, and total number of hours scheduled. Results should be displayed sorted in ascending order by role, then by first name, then by last name.

```

SELECT E.FName, E.LName, MIN(R.Start_Date) AS First_Day_Work, R.Title as
Role, E.DoB, E.Email, SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS
Total_Hours
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE (R.Title = 'Nurse' OR R.Title = 'Doctor')
    AND R.End_Date IS NULL
    AND E.EID IN (SELECT I.EID
        FROM Infection I
        WHERE I.Nature = 'COVID-19'
        GROUP BY I.EID
        HAVING COUNT(*) >= 3)
GROUP BY E.EID
ORDER BY R.Title ASC, E.FName ASC, E.LName ASC;

```

17. Get details of the nurse(s) or doctor(s) who are currently working and has never been infected by COVID-19. Details include first-name, last-name, first day of work as a nurse or as a doctor, role

(nurse/doctor), date of birth, email address, and total number of hours scheduled. Results should be displayed sorted in ascending order by role, then by first name, then by last name.

```
SELECT E.FName, E.LName, MIN(R.Start_Date) AS First_Day_Work, R.Title as Role, E.DoB, E.Email, SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS Total_Hours
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE (R.Title = 'Nurse' OR R.Title = 'Doctor')
    AND R.End_Date IS NULL
    AND E.EID NOT IN (SELECT DISTINCT I.EID
        FROM Infection I
        WHERE I.Nature = 'COVID-19')
GROUP BY E.EID
ORDER BY R.Title ASC, E.FName ASC, E.LName ASC;
```

18. You should show the trigger(s) used by your system. Explain the trigger(s) used and their benefits.

Schedule

check_schedule_gap

The check_schedule_gap trigger activates **before a new record is inserted into the Schedule table**. Its purpose is to ensure that there is a minimum one-hour gap between any two consecutive schedules for an employee on the same date.

The trigger retrieves the maximum end time of the employee's previous schedules on the same date. If a previous schedule exists, the trigger checks if the start time of the new schedule is at least one hour later than the end time of the previous schedule. If this condition is not met, the trigger raises an error with a message stating that a one-hour gap is required between the first and second schedules.

The benefits of this trigger include automatically enforcing a one-hour gap between consecutive schedules, minimizing manual checks, and ensuring employees have sufficient time to rest and prepare for their next shift, which ultimately contributes to a healthier work environment.

prevent_unvaccinated_schedule_insert

The prevent_unvaccinated_schedule_insert trigger activates **before a new record is inserted into the Schedule table**. Its purpose is to ensure that only employees who have been vaccinated within the last six months can be scheduled for work.

The trigger checks the date of the most recent vaccination for the employee being scheduled. If the employee has not received any vaccines, or if their last vaccination occurred more than six months before the new schedule date, an error is raised, preventing the insertion of the new schedule.

The benefits of this trigger include automatically enforcing vaccination requirements for scheduling, reducing manual checks, and maintaining a safe work environment by ensuring that employees scheduled for work have up-to-date vaccinations. This ultimately helps to protect the health and safety of both the employees and the individuals they interact with.

prevent_unvaccinated_schedule_update

The prevent_unvaccinated_schedule_update trigger activates **before an existing record in the Schedule table is updated**. It ensures that only employees who have been vaccinated within the last six months can have their schedules updated.

The trigger checks the date of the most recent vaccination for the employee being scheduled. If the employee has not received any vaccines, or if their last vaccination occurred more than six months before the new schedule date, an error is raised, preventing the update of the schedule.

The benefits of this trigger include automatically enforcing vaccination requirements for schedule updates, reducing manual checks, and maintaining a safe work environment by ensuring that employees with updated schedules have up-to-date vaccinations. This ultimately helps to protect the health and safety of both the employees and the individuals they interact with.

prevent_infected_scheduling

This trigger is used to prevent infected doctors and nurses from being assigned to a schedule within the next two weeks. The trigger checks the Infection and Roles tables to retrieve the infection date and employee role of the employee being scheduled. If the employee is a doctor or nurse and has an infection date within the last two weeks, the trigger will raise an error message and prevent the insertion of the new schedule record.

The benefit of this trigger is that it helps to prevent the spread of infection within the healthcare facility by ensuring that infected employees are not scheduled to work until they have fully recovered and are no longer a risk to others.

log_schedule_email

This trigger is designed to log the schedules of employees and send them via email. It is triggered **after the insertion of a new record into the Schedule table for each row**. The trigger retrieves the necessary details of the employee and the facility, including their first and last name, email address, and facility name and address. It also determines the start and end dates of the week based on the date of the schedule.

If an email for the given employee and week already exists, the trigger updates the email body with the new schedule. Otherwise, it creates a new email and inserts a record into the Email_Logged table.

The benefits of this trigger include efficient logging and management of employee schedules, as well as a more streamlined communication process between employees and their supervisors. This trigger can help ensure that employees are aware of their schedules in advance and can plan their work accordingly.

Infection

notify_contacted_employees

The notify_contacted_employees trigger is executed **after a new row is inserted into the Infection table**. Its purpose is to automatically notify employees (doctors and nurses) who have worked with an infected colleague in the past two weeks.

The trigger uses a cursor to fetch relevant employees, then loops through them, inserting a warning email into the Email table and logging this information in the Email_logged table. This trigger helps ensure timely communication, reduces manual work, and maintains a log of sent emails for tracking and analysis.

cancel_infected_schedules

The cancel_infected_schedules trigger is executed **after a new row is inserted into the Infection table**. Its purpose is to automatically cancel the schedules of infected doctors and nurses for a period of two weeks following their infection date.

When a new infection is recorded with a non-null infection date, the trigger identifies doctors and nurses with active roles (where End_Date is null) and removes their schedules within the two-week period after the infection date.

The benefits of this trigger include automated schedule adjustments, reduced manual work, and ensuring that infected employees are not scheduled to work during their recovery period, which helps maintain a safe work environment.

update_cancel_infected_schedule

The update_cancel_infected_schedules trigger activates **after an update occurs in the Infection table**. Its primary function is to automatically remove scheduled shifts for doctors and nurses who become infected, covering a two-week period following their infection date when the infection date is modified.

When the infection date in a record is updated to a non-null value, the trigger finds doctors and nurses with ongoing roles (where End_Date is null) and cancels their schedules within the specified two-week timeframe after the infection date.

The advantages of this trigger include automatic schedule changes upon updating infection dates, minimizing manual intervention, and ensuring that infected staff members are not scheduled to work during their recuperation period, thereby promoting a secure work environment.

19. You need to demonstrate the integrity of all the requirements provided in the description. Example, the system should not allow a user to schedule an employee on two different conflicting times.

Insert two conflicting time for a given employee:

Private < mc55531.encc.concordia.ca Schedule Home

Employee ID (if known)	Facility ID (if known)	Date (if known)			
123005		04/11/2023	Delete	Update	
Add a Schedule					
EID	FID	Date	Start_Time	End_Time	Action
123005	1005	2023-03-09	10:14:00	12:00:00	Delete Update
123005	1005	2023-03-14	11:00:00	13:00:00	Delete Update
123005	1005	2023-03-15	11:00:00	13:00:00	Delete Update
123005	1005	2023-04-03	09:00:00	17:00:00	Delete Update
123005	1005	2023-04-04	09:00:00	17:00:00	Delete Update
123005	1005	2023-04-04	18:30:00	19:30:00	Delete Update

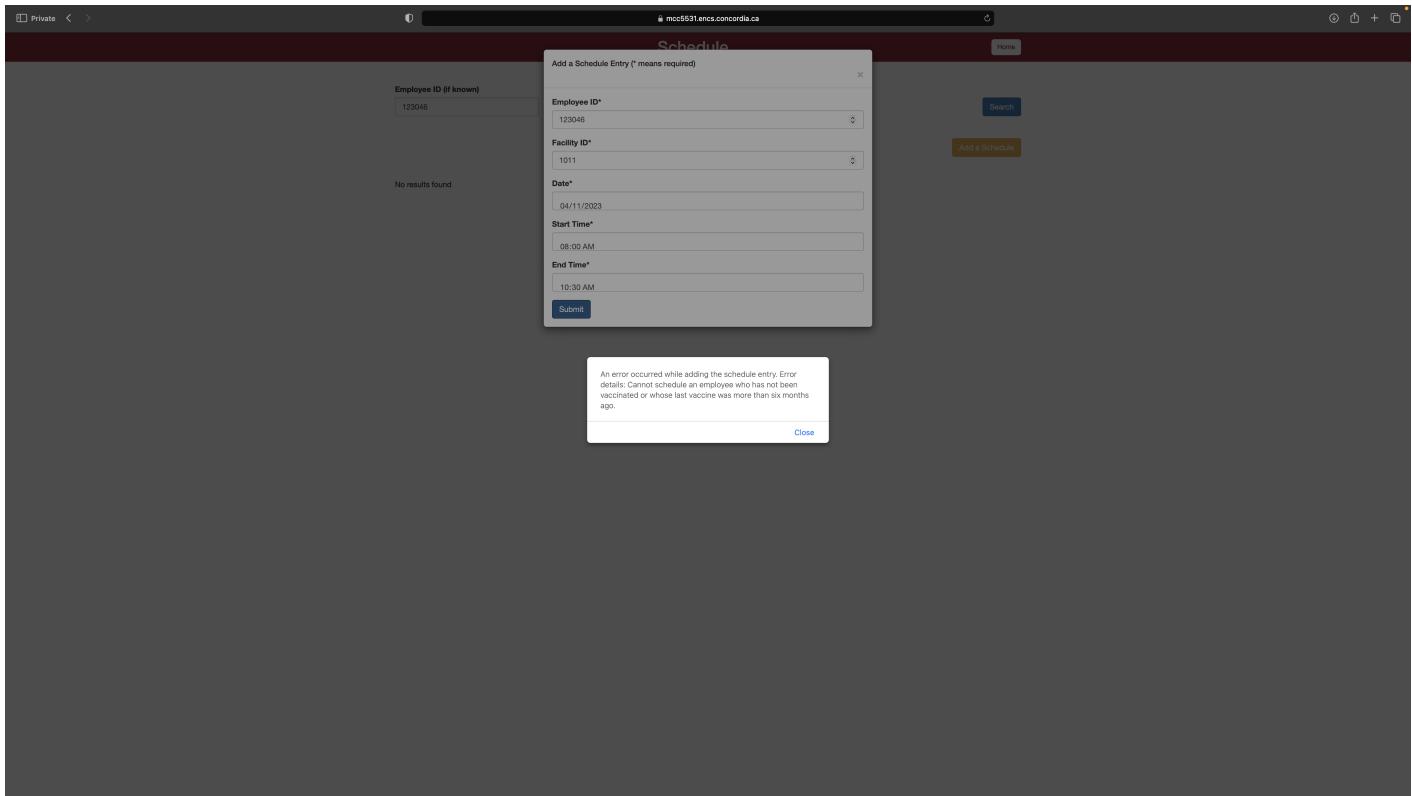
Private < mc55531.encc.concordia.ca Schedule Home

Employee ID (if known)	Facility ID (if known)	Date (if known)			
123005			Search	Add a Schedule	
Add a Schedule Entry (* means required)					
Employee ID*	123005	Facility ID*	1005		
Date*	2023-04-04	Start Time*	08:00 PM	Update	
		End Time*	10:30 PM	Update	
			Submit	Update	
123005	1005	2023-04-04	08:00:00	17:00:00	Delete Update
123005	1005	2023-04-04	18:30:00	19:30:00	Delete Update

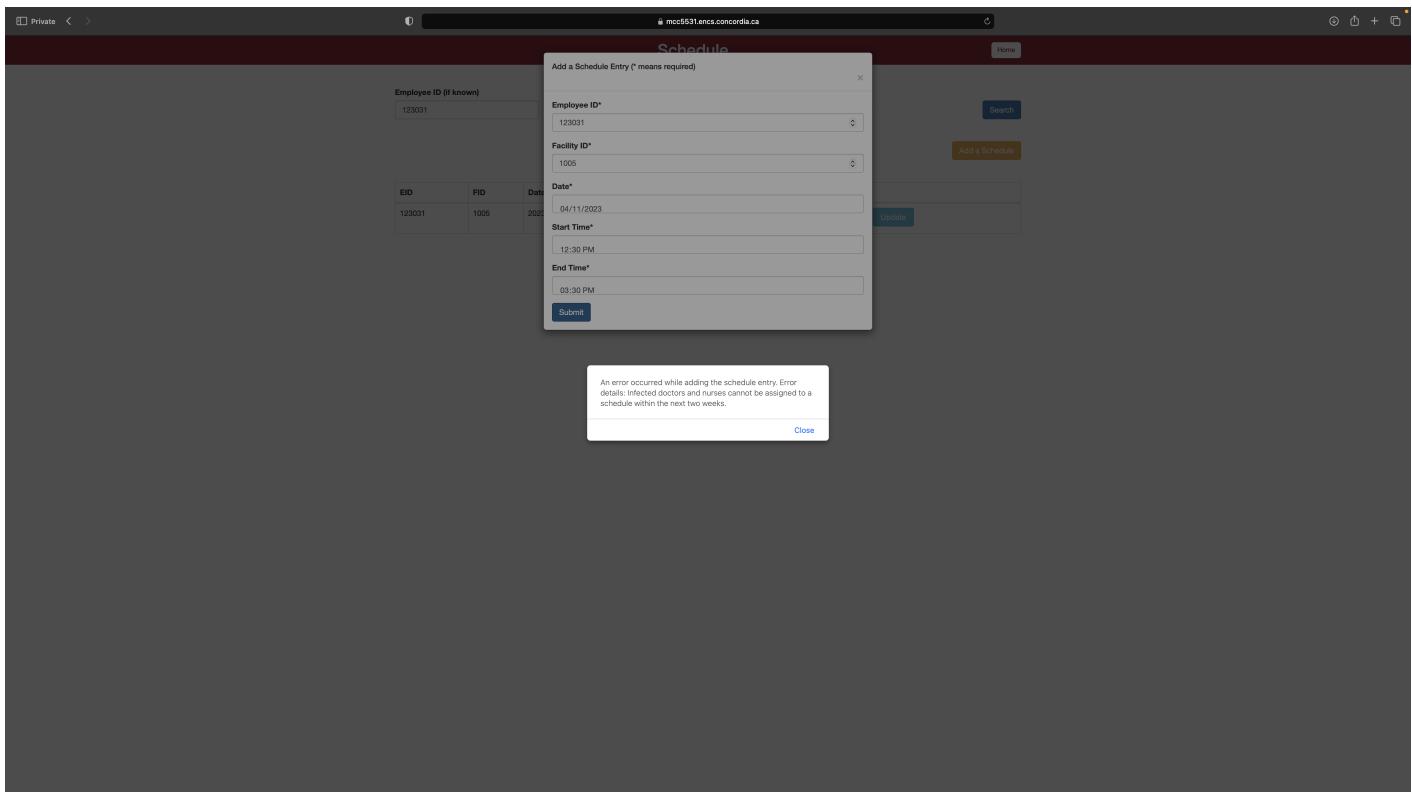
An error occurred while adding the schedule entry. Error details: at least one hour should be the duration between the first schedule and the second one.

[Close](#)

Insert a schedule for an employee who has not been vaccinated within last 6 months:



Insert a schedule for a doctor or nurse who has been infected within past 2 weeks:



Send warning email to the employees who have been in contact with an infected doctor or nurse within past 2 weeks:

17	Subject: Warning	Body: One of your colleagues that you have worked with in the past two weeks has been infected with COVID-19.
18	Subject: Warning	Body: One of your colleagues that you have worked with in the past two weeks has been infected with COVID-19.
19	Subject: Warning	Body: One of your colleagues that you have worked with in the past two weeks has been infected with COVID-19.
20	Subject: Warning	Body: One of your colleagues that you have worked with in the past two weeks has been infected with COVID-19.
21	Subject: Warning	Body: One of your colleagues that you have worked with in the past two weeks has been infected with COVID-19.
*	HULL	HULL

20. You need to demonstrate the generation of emails and the logs of the emails produced by the system.

In our trigger 'log_schedule_email', every time there is an insertion into the schedule, the system will generate an empty log in the Email table, and then grab the schedule to connect with the empty log.

If the new schedule is the 1st schedule of an employee in that week, it will be added to the log and generate an email in the format like 'subject - body - detailed schedule(s)'.

```
INSERT INTO Schedule (FID, EID, date, start_time, end_time) VALUES ('1003',
'123027', '2023-04-13', '8:00:00', '10:00:00');
```

	FID	EID	date	start_time	end_time
▶	1003	123027	2023-04-13	08:00:00	10:00:00

```
SELECT * FROM Email;
```

16	Subject: Schedule for Mon-2023-04-10 to Sun-2023-04-16 Name: William Roy Email: williamroy83@gmail.com Facility: Hospital Maisonneuve Rosemont (04-13-Thursday) Start: 08:00:00 - End: 10:00:00
▶*	Subject: Schedule for Mon-2023-04-10 to Sun-2023-04-16 Name: William Roy Email: williamroy83@gmail.com Facility: Hospital Maisonneuve Rosemont (04-13-Thursday) Start: 08:00:00 - End: 10:00:00

Later, if another new schedule is input for the same employee in the same week, the system will compare it with the existing schedule(s) by time and generate an updated email with all schedules in the correct order.

```
INSERT INTO Schedule (FID, EID, date, start_time, end_time) VALUES ('1003',
'123027', '2023-04-14', '8:00:00', '10:00:00'), ('1004', '123027', '2023-04-12', '8:00:00', '10:00:00');
```

	FID	EID	date	start_time	end_time
▶	1003	123027	2023-04-14	08:00:00	10:00:00
	1003	123027	2023-04-13	08:00:00	10:00:00
	1004	123027	2023-04-12	08:00:00	10:00:00

```
SELECT * FROM Email;
```

16	Subject: Schedule for Mon-2023-04-10 to Sun-2023-04-16 Name: William Roy Email: williamroy83@gmail.com
17	Subject: Warning Body: Subject: Schedule for Mon-2023-04-10 to Sun-2023-04-16
18	Subject: Warning Body: Name: William Roy
19	Subject: Warning Body: Email: williamroy83@gmail.com
20	Subject: Warning Body:
21	Subject: Warning Body:
22	Subject: Schedule for Mon-2023-04-10 to Sun-2023-04-16 Facility: Mount Sinai Hospital (04-12-Wednesday) Start: 08:00:00 - End: 10:00:00
Email 1	x
Output :	Facility: Hospital Maisonneuve Rosemont (04-13-Thursday) Start: 08:00:00 - End: 10:00:00
<input type="checkbox"/> Action Output	Facility: Hospital Maisonneuve Rosemont (04-14-Friday) Start: 08:00:00 - End: 10:00:00
#	Time
41	05:28:12
	DELETE FR

Additional features

1. An easy way to check answers for SQL DDL and DML commands (Q6-17) through the web interface (<https://mcc5531.encs.concordia.ca/query.html>)

Check the appendix for complete answers.

OB. Get details of all the facilities in the system. Details include facility's name, address, city, province, postal-code, phone number, web address, type, capacity, general manager's name and number of employees currently working for the facility. Results should be displayed sorted in ascending order by province, then by city, then by type, then by number of employees currently working for the facility.

```
SELECT
    f.FName,
    f.Address,
    f.City,
    f.Province,
    f.Postal_Code,
    f.Phone_Number,
    f.Web_Address,
    f.Type,
    f.Capacity,
    CONCAT(e.FName, ' ', e.LName) AS ManagerName,
    COUNT(DISTINCT r.EID) AS NumEmployees
FROM Facilities f
LEFT JOIN Managers m
    ON f.FID = m.FID
    LEFT JOIN Employees e
        ON m.EID = e.EID
    LEFT JOIN Roles r
        ON f.FID = r.FID AND r.end_date IS NULL
GROUP BY f.FID, e.FName, e.LName
ORDER BY f.Province ASC, f.City ASC, f.Type ASC, NumEmployees ASC;
```

Check

Facility ID	Name	Address	City	Province	Postal Code	Phone Number	Web Address	Type	Capacity	Manager Name	NumEmployees
1001 Burnet St	Vancouver General Hospital	888 W 12th Ave	Vancouver	BC	V6Z 1M8	604-875-4111	vch.ca	Hospital	600	Karen Green	12
805 W 12th St	Vancouver General Pharmacy	1081 Burnet St	Vancouver	BC	V6Z 1Y6	604-635-2860	www.vancouver.ca	Pharmacy	50	Oliver Cheng	11
3177 Kingsway Rd	RC Town Hall	805 W 12th St	Vancouver	BC	V6Z 1M8	604-675-4747	www.rctownhall.ca	Special Treatment	60	Jacobs Smith	11
2071 Bayview Ave	Sunnybrook Health Sciences Center	Toronto	ON	MAIN 1005	416-480-6100	www.sunnybrook.ca	Hospital	600	David Zhang	9	
204 Elizabeth St	Toronto General Hospital	Toronto	ON	M5G 1C4	416-348-3885	www.tgh.ca	Hospital	907	Isabella Ladee	2	
168 Bathurst St	Toronto Community Health Centre	Toronto	ON	M1V 2H4	416-783-6460	www.tchc.ca	Hospital	80	Peter Brown	1	
7075 Yonge St	Bell Clinic at Markham	Toronto	ON	M3J 1S2	416-323-6460	www.uhmc.ca	Hospital	70	John Anderson	6	
260 Augusta Ave	Toronto Community Pharmacy	Toronto	ON	M4T 1A4	416-688-4162	www.torontopharm.ca	Pharmacy	60	Ella Lee	4	
100 Queen St W	Rogers Stadium	Toronto	ON	M5J 1E2	416-867-1793	www.torontostadium.com	Special Treatment	20	Grace Chen	3	
150 Bloor St W	Crespi at St. Michael's Hospital	Toronto	ON	M5G 2W6	416-415-6880	www.crespiatsmh.com	Special Treatment	100	Ava Martin	8	
4011 Rue Ontario Est	Mount Sinai Hospital	Montreal	QC	H3T 1R2	514-351-6300	www.mshospital.ca	Hospital	400	Diane Gagnon	6	
7080 Hockenberry St	Hospital Maisonneuve-Rosemont	Montreal	QC	H3V 1Y8	514-275-6551	www.maisonneuve.ca	Hospital	400	Halley Nguyen	9	
3716 Chemin de la Côte-Garneau-Catherine	Jewish General Hospital	Montreal	QC	H3T 1S2	514-946-6222	www.jgh.ca	Hospital	600	Amy Kim	14	
1200 Rue Sainte-Justine	Montreal CLSC des Rive-Sud	Montreal	QC	H3V 3P7	514-921-2901	www.mclscdesrivesud.ca	CLSC	70	William Wong	12	
5700 Côte des Neiges Rd	Montreal CLSC Côte-des-Niges	Montreal	QC	H3T 1A2	514-731-6551	www.mclscdesniges.ca	CLSC	150	Stephen Xu	15	
800 Beaumont Ave	Gatineau Clinic	Montreal	QC	H3V 1R2	514-731-6643	www.gatineauclinic.ca	Pharmacy	80	Stephen Lu	11	
1000 Avenue A	Montreal Royal Pharmacy	Montreal	QC	H3Z 1B2	514-931-0310	www.montrealroyal.ca	Pharmacy	20	Anneke Wong	7	
1625 Chemin de la Côte-Garneau-Catherine	Concordia University	Montreal	QC	H3V 1L7	514-498-5511	www.concordia.ca	Special Treatment	60	Louise Gagnon	2	

Q7. Get details of all the employees currently working in a specific facility. Details include employee's first-name, last-name, start date of work, date of birth, Medicare card number, telephone one-number, address, city, province, postal-code, citizenship, and email address. Results should be displayed sorted in ascending order by role, then by first name, then by last name.

Facility ID: 1005

2. Google Map for facilities by clicking on Address

The screenshot shows a web-based facility management system. At the top, there is a search bar with the URL "mcc6531.encs.concordia.ca". Below the search bar, there is a "Facility Location" section containing a map and a table of facilities.

Facility Location:

- Map: Shows the location of Jewish General Hospital on a street map of Montreal, Quebec, Canada.
- Satellite: Shows the same location from a satellite perspective.
- Search: A text input field for searching facilities.
- Type: Buttons for Hospital, Clinic, Pharmacy, Special Installment, and CLSC.
- Add a Facility: A button to add a new facility.

Facilities Table:

FID	Name	Address	City	Prov	Postal	Phone	Website	Type	Capacity	Action
1001	Toronto General Hospital	200 Elizabeth St						hospital	507	<button>Delete</button> <button>Update</button>
1002	Sunnybrook Health Sciences Center	2075 Bayview Ave						hospital	899	<button>Delete</button> <button>Update</button>
1003	Hospital Maisonneuve Rive-Mont	7085 Hutchison St	Montreal	QC	H3N 1Y9	514-273-9591	www.mnhospital.com	hospital	400	<button>Delete</button> <button>Update</button>
1004	Mount Sinai Hospital	4201 Rue Ontario Est	Montreal	QC	H1W 1T2	514-251-8383	www.mountsinai.on.ca	hospital	450	<button>Delete</button> <button>Update</button>
1005	Jewish General Hospital	3755 Chemin de la Côte-Sainte-Catherine	Montreal	QC	H3T 1E2	514-340-8232	www.jgh.ca	hospital	500	<button>Delete</button> <button>Update</button>
1006	Vancouver General Hospital	889 W 12th Ave	Vancouver	BC	V5Z 1M9	604-875-4111	vch.ca	hospital	800	<button>Delete</button> <button>Update</button>
1007	Montreal CLSC Côte-des-Neiges	5700 Côte-des-Neiges Rd	Montreal	QC	H3T 2A2	514-731-8531	www.csssdcdn.qc.ca	CLSC	150	<button>Delete</button> <button>Update</button>
1008	Montreal CLSC des Faubourgs	1250 Rue Sanguinet	Montreal	QC	H2X 3E7	514-527-2361	www.cessdelamontagne.qc.ca	CLSC	75	<button>Delete</button> <button>Update</button>
1009	Toronto Community Health Centre	168 Bathurst St	Toronto	ON	M5V 2R4	416-703-8480	www.torontocho.ca	clinic	60	<button>Delete</button> <button>Update</button>
1010	Best Clinic at Markham	7678 Grenville St	Toronto	ON	M0S 1B2	416-323-6400	www.usfchs.ca	clinic	70	<button>Delete</button> <button>Update</button>
1011	Vancouver Community Health Centre	1081 Burrard St	Vancouver	BC	V6Z 1Y6	604-633-2600	www.vancouverhc.ca	clinic	50	<button>Delete</button> <button>Update</button>

CONTRIBUTIONS

All group members contributed equally.

Appendix

Advanced Query

[Home \(index.html\)](#)

Q6. Get details of all the facilities in the system. Details include facility's name, address, city, province, postal-code, phone number, web address, type, capacity, general manager's name and number of employees currently working for the facility. Results should be displayed sorted in ascending order by province, then by city, then by type, then by number of employees currently working for the facility.

```
SELECT
    f.FName,
    f.Address,
    f.City,
    f.Province,
    f.Postal_Code,
    f.Phone_Number,
    f.Web_Address,
    f.FType,
    f.Capacity,
    CONCAT(e.FName, ' ', e.LName) AS ManagerName,
    COUNT(DISTINCT r.EID) AS NumEmployees
FROM
    Facilities f
        LEFT JOIN
    Managed_by m ON f.FID = m.FID
        LEFT JOIN
    Employees e ON m.EID = e.EID
        LEFT JOIN
    Roles r ON f.FID = r.FID AND r.end_date IS NULL
GROUP BY f.FID , e.FName , e.LName
ORDER BY f.Province ASC , f.City ASC , f.FType ASC , NumEmployees ASC;
```

[Check](#)

FName	Address	City	Province	Postal_Code	Phone_Number	Web_Address	FType	Capacity	ManagerName	NumEmployees
Vancouver General Hospital	899 W 12th Ave	Vancouver	BC	V5Z 1M9	604-875-4111	vch.ca	hospital	600	Kevin Green	13
Vancouver Community Health Centre	1081 Burrard St	Vancouver	BC	V6Z 1Y6	604-633-2600	www.vancouverchc.ca	clinic	50	Oliver Chang	11
Vancouver General Pharmacy	855 W 12th Ave	Vancouver	BC	V5Z 1M9	604-875-4747	www.vgph.ca	pharmacy	30	Jacob Smith	11
BC Town Hall	3777 Kingsway W	Vancouver	BC	V5R 5M4	604-436-2211	www.metropolisatmetrotown.com	special installment	50	David Zhang	9
Sunnybrook Health Sciences Center	2075 Bayview Ave	Toronto	ON	M4N 3M5	416-480-6100	sunnybrook.ca	hospital	699	Nathan Chen	6
Toronto General Hospital	200 Elizabeth St	Toronto	ON	M5G 2C4	416-340-3935	www.uhn.ca/TGH	hospital	507	Isabella Lavoie	7
Toronto Community Health Centre	168 Bathurst St	Toronto	ON	M5V 2R4	416-703-8480	www.torontochc.org	clinic	60	Rachel Brown	1
Best Clinic at Markham	7678 Grenville St	Toronto	ON	M5S 1B2	416-323-6400	www.uoftchs.ca	clinic	70	John Anderson	6

FName	Address	City	Province	Postal_Code	Phone_Number	Web_Address	FType	Capacity	ManagerName	NurmEmployees
Toronto Community Pharmacy	260 Augusta Ave	Toronto	ON	M5T 2L9	416-599-9783	www.torontopharm.ca	pharmacy	40	Ella Lee	6
Rogers Stadium	100 Queen St W	Toronto	ON	M5H 2N2	416-867-7970	www.1queenwest.com	special installment	20	Grace Chen	7
Cineplex at Toronto City Center	150 Bloor St W	Toronto	ON	M5S 2X9	416-413-6868	www.cineplextoronto.com	special installment	100	Ava Martin	8
Mount Sinai Hospital	4201 Rue Ontario Est	Montreal	QC	H1W 1T2	514-251-8383	www.mountsinai.on.ca	hospital	450	Owen Gagne	6
Hospital Maisonneuve Rosemont	7085 Hutchison St	Montreal	QC	H3N 1Y9	514-273-9591	www.mrhsospital.com	hospital	400	Hailey Nguyen	9
Jewish General Hospital	3755 Chemin de la Côte-Sainte-Catherine	Montreal	QC	H3T 1E2	514-340-8222	www.jgh.ca	hospital	500	Amy Kim	14
Montreal CLSC des Faubourgs	1250 Rue Sanguinet	Montreal	QC	H2X 3E7	514-527-2361	www.csssdelamontagne.qc.ca	CLSC	75	William Wilson	12
Montreal CLSC Cote-des-Neiges	5700 Cote-des-Neiges Rd	Montreal	QC	H3T 2A2	514-731-8531	www.cssscdn.qc.ca	CLSC	150	Sophia Xu	15
Concordia Clinic	950 Beaumont Ave	Montreal	QC	H3N 1W3	514-731-5544	www.concordiaclinic.ca	clinic	80	Sophie Liu	11
Montreal Rexall Pharmacy	1500 Atwater Ave	Montreal	QC	H3Z 1X5	514-931-3316	www.rexall.ca/montreal	pharmacy	25	Andrew Wong	7
Concordia University	1535 Chemin de la Côte-Sainte-Catherine	Montreal	QC	H2V 2J7	514-345-3511	www.concordia.ca	special installment	60	Lucas Gagnon	2

Q7. Get details of all the employees currently working in a specific facility. Details include employee's first-name, last-name, start date of work, date of birth, Medicare card number, telephone-number, address, city, province, postal-code, citizenship, and email address. Results should be displayed sorted in ascending order by role, then by first name, then by last name.

Facility ID:

```

SELECT
    f.FID,
    e.FName,
    e.LName,
    r.Start_Date,
    e.DoB,
    e.Medicare_Number,
    e.Telephone_Number,
    e.Address,
    e.City,
    e.Province,
    e.Postal_code,
    e.Citizenship,
    e.Email
FROM
    Employees e
        JOIN
    Roles r ON e.EID = r.EID
        JOIN
    Facilities f ON f.FID = r.FID
WHERE
    r.end_date IS NULL
    AND f.FID = ?
GROUP BY f.FID , e.EID
ORDER BY f.FID ASC , r.title ASC , e.FName ASC , e.LName ASC;

```

Check

FID	FName	LName	Start_Date	DoB	Medicare_Number	Telephone_Number	Address	City	Province	Postal_code	Citizenship	Email
1005	Ava	Kim	2020-05-15	1984-09-07	518246703	416-593-2094	134 Belgravia Ave	Toronto	ON	M6E 2M2	Canada	avakim84@hotmail.com
1005	Ella	Kim	2019-06-15	1986-06-11	367509824	416-835-7291	75 Eastwood Ave	Toronto	ON	M1N 3A8	Canada	ellakim86@hotmail.com
1005	Jacob	Park	2022-09-30	1988-04-19	752148930	416-624-8190	205 Wynford Dr	Toronto	ON	M3C 1K1	Canada	jacob.park88@gmail.com
1005	Noah	Lee	2021-03-15	1981-02-12	489257631	416-328-5721	29 Kinghorn Ave	Toronto	ON	M4B 1K4	Canada	noahlee81@gmail.com
1005	Sophie	Liu	2020-04-01	1983-01-16	908267413	416-729-6875	2 Parnham Crt	Toronto	ON	M1B 4G4	Canada	sophie.liu83@hotmail.com
1005	Avery	Lavoie	2018-06-01	1984-09-05	240973186	514-957-8193	2555 Rue Des Ormeaux	Montreal	QC	H1L 4X1	Canada	avery.lavoie84@hotmail.com
1005	Chloe	Belanger	2019-06-01	1983-03-26	670491235	514-789-4215	1329 Rue Sherbrooke Ouest	Montreal	QC	H3G 1J4	Canada	chloe.belanger83@hotmail.com
1005	Ethan	Tran	2019-03-01	1980-06-27	921475863	514-631-7624	5223 Rue de la Roche	Montreal	QC	H2J 3J7	Canada	ethantran80@gmail.com
1005	Jacob	Tremblay	2022-02-01	1981-08-08	936481720	514-343-2312	1071 Rue Sainte-Catherine E	Montreal	QC	H2L 2G2	Canada	jacobtremblay81@gmail.com

FID	FName	LName	Start_Date	DoB	Medicare_Number	Telephone_Number	Address	City	Province	Postal_code	Citizenship	Email
1005	Liam	Boucher	2020-01-01	1986-02-18	185249036	514-215-3986	3593 Rue de la Montagne	Montreal	QC	H3G 2A7	Canada	liamboucher86@gmail.com
1005	Olivia	Kim	2020-12-01	1980-06-11	594827103	416-913-4830	27 Jackman Dr	Toronto	ON	M2R 2W1	Canada	oliviakim80@hotmail.com
1005	Amy	Kim	2018-02-01	1969-07-15	329408122	514-716-4592	8361 Rue Saint-Denis	Montreal	QC	H2P 2H8	Canada	amykim89@hotmail.com
1005	Jacob	Lee	2022-02-26	1984-02-23	625849137	604-958-4362	6015 Oak St	Vancouver	BC	V6M 2V5	Canada	jacoblee84@hotmail.com
1005	Avery	Lee	2019-09-12	1989-03-15	976345812	604-875-3210	2450 Cornwall Ave	Vancouver	BC	V6K 1B8	Canada	averylee89@gmail.com

Q8. For a given employee, get the details of all the schedules she/he has been scheduled during a specific period of time. Details include facility name, day of the year, start time and end time. Results should be displayed sorted in ascending order by facility name, then by day of the year, then by start time.

Employee ID:

Start Date: End Date:

```

SELECT
    f.FName AS Facility,
    e.FName,
    e.LName,
    s.date,
    DAYOFYEAR(s.date) AS 'Day of the year',
    s.start_time,
    s.end_time
FROM
    Facilities f
        JOIN
    Schedule s ON f.FID = s.FID
        JOIN
    Employees e ON s.EID = e.EID
WHERE
    s.EID = ?
        AND s.date BETWEEN ? AND ?
ORDER BY f.FName ASC , DAYOFYEAR(s.date) ASC , s.start_time ASC;

```

Facility	FName	LName	date	Day of the year	start_time	end_time
Sunnybrook Health Sciences Center	Nathan	Chen	2023-03-02	61	09:00:00	11:00:00
Sunnybrook Health Sciences Center	Nathan	Chen	2023-03-06	65	11:00:00	13:00:00
Sunnybrook Health Sciences Center	Nathan	Chen	2023-03-09	68	14:00:00	16:00:00
Sunnybrook Health Sciences Center	Nathan	Chen	2023-03-13	72	09:00:00	11:00:00
Sunnybrook Health Sciences Center	Nathan	Chen	2023-03-26	85	14:00:00	16:00:00
Sunnybrook Health Sciences Center	Nathan	Chen	2023-03-27	86	14:00:00	16:00:00

Q9. Get details of all the doctors who have been infected by COVID-19 in the past two weeks. Details include the doctor's first-name, last-name, date of infection, and the name of the facility that the doctor is currently working for. Results should be displayed sorted in ascending order by the facility name, then by the first-name of the doctor.

```

SELECT
    e.EID, e.FName AS First_Name, e.LName AS Last_Name, i.I_date, f.FName AS Facility_Name
FROM
    Employees e
        LEFT JOIN
    Infection i ON e.EID = i.EID
        JOIN
    Roles r ON e.EID = r.EID AND r.end_date IS NULL
        JOIN
    Facilities f ON r.FID = f.FID
WHERE
    Nature = 'COVID-19'
        AND r.Title = 'Doctor'
        AND i.I_date BETWEEN DATE_SUB(NOW(), INTERVAL 2 WEEK) AND NOW()
ORDER BY f.FName ASC , e.FName ASC;

```

EID	First_Name	Last_Name	I_date	Facility_Name
123028	Ella	Dubois	2023-03-29	Hospital Maisonneuve Rosemont
123026	Sophia	Guerin	2023-04-05	Hospital Maisonneuve Rosemont
123040	Ethan	Desjardins	2023-04-08	Montreal CLSC Cote-des-Neiges
123037	Sophie	Dubois	2023-04-07	Montreal CLSC Cote-des-Neiges
123024	Bella	Wu	2023-04-01	Mount Sinai Hospital
123024	Bella	Wu	2023-04-01	Sunnybrook Health Sciences Center
123024	Bella	Wu	2023-04-01	Toronto General Hospital

Q10. List the emails generated by a given facility. The result should be displayed in ascending order by the date of the emails.

Facility ID:

```

SELECT Facilities.FID, Email.ID AS Email_ID, Email_logged.EID AS Receiver, Email.text
FROM Email_logged
JOIN Facilities on Email_logged.FID = Facilities.FID
JOIN Email ON Email_logged.ID = Email.ID
WHERE Email_logged.FID = ?
GROUP BY Email_logged.FID, Email_logged.EID, Email.ID
ORDER BY Email_logged.date ASC;

```

FID	Email_ID	Receiver	text
1005	12	123005	Subject: Schedule for Mon-2023-04-03 to Sun-2023-04-09 Name: Amy Kim Email: amykim89@hotmail.com Facility: Jewish General Hospital (Monday) Start: 09:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Tuesday) Start: 09:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Tuesday) Start: 18:30:00 - End: 19:30:00
1005	3	123033	Subject: Schedule for Mon-2023-04-03 to Sun-2023-04-09 Name: Liam Boucher Email: liamboucher86@gmail.com Facility: Jewish General Hospital (Wednesday) Start: 09:00:00 - End: 12:00:00 Facility: Jewish General Hospital (Thursday) Start: 09:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Friday) Start: 09:00:00 - End: 17:00:00
1005	6	123054	Subject: Schedule for Mon-2023-04-03 to Sun-2023-04-09 Name: Noah Lee Email: noahlee81@gmail.com Facility: Jewish General Hospital (Monday) Start: 15:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Tuesday) Start: 15:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Wednesday) Start: 12:00:00 - End: 15:00:00 Facility: Jewish General Hospital (Thursday) Start: 09:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Friday) Start: 09:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Sunday) Start: 09:00:00 - End: 17:00:00
1005	4	123034	Subject: Schedule for Mon-2023-04-03 to Sun-2023-04-09 Name: Chloe Belanger Email: chloe.belanger83@hotmail.com Facility: Jewish General Hospital (Monday) Start: 15:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Tuesday) Start: 15:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Wednesday) Start: 09:00:00 - End: 12:00:00 Facility: Jewish General Hospital (Saturday) Start: 09:00:00 - End: 17:00:00
1005	5	123035	Subject: Schedule for Mon-2023-04-03 to Sun-2023-04-09 Name: Jacob Tremblay Email: jacqbremblay81@gmail.com Facility: Jewish General Hospital (Wednesday) Start: 09:00:00 - End: 12:00:00 Facility: Jewish General Hospital (Thursday) Start: 10:00:00 - End: 12:00:00 Facility: Jewish General Hospital (Friday) Start: 10:00:00 - End: 12:00:00 Facility: Jewish General Hospital (Saturday) Start: 09:00:00 - End: 17:00:00 Facility: Jewish General Hospital (Sunday) Start: 09:00:00 - End: 17:00:00

Q11. For a given facility, generate a list of all the doctors and nurses who have been on schedule to work in the last two weeks. The list should include first-name, last-name, and role. Results should be displayed in ascending order by role, then by first name.

Facility ID:

```
SELECT DISTINCT E.FName, E.LName, R.Title
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE R.FID = ?
AND R.title IN ('Doctor', 'Nurse')
AND S.date BETWEEN DATE_SUB(CURRENT_DATE, INTERVAL 2 WEEK) AND CURRENT_DATE
ORDER BY R.title ASC, E.FName ASC;
```

FName	LName	Title
Ava	Kim	nurse
Ella	Kim	nurse
Noah	Lee	nurse
Sophie	Liu	nurse
Chloe	Belanger	doctor
Ethan	Tran	doctor
Jacob	Tremblay	doctor
Liam	Boucher	doctor
Sophie	Dubois	doctor

Q12. For a given facility, give the total hours scheduled for every role during a specific period. Results should be displayed in ascending order by role.

Facility ID: 1005

Start Date: 2023-03-01

End Date: 2023-04-01

```
SELECT R.title, SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS Total_Hours
FROM Roles R
JOIN Schedule S ON R.EID = S.EID
WHERE R.FID = ?
AND S.date BETWEEN ? AND ?
GROUP BY R.title
ORDER BY R.title ASC;
```

Check

title	Total_Hours
nurse	6
doctor	10
receptionist	2
administrative personnel	5
regular employee	2

Q13. For every facility, provide the province where the facility is located, the facility name, the capacity of the facility, and the total number of employees in the facility who have been infected by COVID-19 in the past two weeks. The results should be displayed in ascending order by province, then by the total number of employees infected.

```
SELECT F.Province, F.FName, F.Capacity, COUNT(I.EID) AS Infected_Employees
FROM Facilities F
LEFT JOIN Roles R ON F.FID = R.FID
LEFT JOIN Infection I ON R.EID = I.EID
AND I.i_date BETWEEN DATE_SUB(CURRENT_DATE, INTERVAL 2 WEEK) AND CURRENT_DATE
AND I.Nature = 'COVID-19'
GROUP BY F.FID, F.Province, F.FName, F.Capacity
ORDER BY F.Province ASC, Infected_Employees ASC;
```

Check

Province	FName	Capacity	Infected_Employees
BC	BC Town Hall	50	0
BC	Vancouver General Pharmacy	30	0
BC	Vancouver Community Health Centre	50	0
BC	Vancouver General Hospital	600	2
ON	Cineplex at Toronto City Center	100	0
ON	Rogers Stadium	20	0
ON	Toronto Community Pharmacy	40	0

Province	FName	Capacity	Infected_Employees
ON	Toronto Community Health Centre	60	0
ON	Best Clinic at Markham	70	0
ON	Sunnybrook Health Sciences Center	699	1
ON	Toronto General Hospital	507	1
QC	Concordia Clinic	80	0
QC	Montreal Rexall Pharmacy	25	0
QC	Concordia University	60	0
QC	Montreal CLSC des Faubourgs	75	1
QC	Mount Sinai Hospital	450	1
QC	Montreal CLSC Cote-des-Neiges	150	2
QC	Jewish General Hospital	500	2
QC	Hospital Maisonneuve Rosemont	400	2

Q14. For every doctor who is currently working in the province of “Québec”, provide the doctor’s first-name, last-name, the city of residence of the doctor, and the total number of facilities the doctor is currently working for. Results should be displayed in ascending order by city, then in descending order by total number of facilities.

```

SELECT E.FName, E.LName, E.city, COUNT(R.FID) as Total_Facilities
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Facilities F ON R.FID = F.FID
WHERE R.title = 'Doctor'
AND F.Province = 'QC'
AND R.end_date IS NULL
GROUP BY E.EID, E.FName, E.LName, E.city
ORDER BY E.city ASC, Total_Facilities DESC;

```

Check

FName	LName	city	Total_Facilities
Emma	Martin	Montreal	2
Ethan	Tran	Montreal	2
Avery	Lavoie	Montreal	2
Liam	Boucher	Montreal	2
Jacob	Tremblay	Montreal	2
Chloe	Belanger	Montreal	1
Ella	Wong	Montreal	1
Caleb	Park	Montreal	1
Avery	Chen	Montreal	1
Aiden	Liu	Montreal	1
Ethan	Desjardins	Montreal	1

FName	LName	city	Total_Facilities
Olivia	Roy	Montreal	1
Noah	Gagne	Montreal	1
Sophie	Dubois	Montreal	1
Sophia	Guerin	Montreal	1
Jacob	Belanger	Montreal	1
Bella	Wu	Montreal	1
Daniel	Gauthier	Montreal	1
Ella	Dubois	Montreal	1
William	Roy	Montreal	1

Q15. Get details of the nurse(s) who is/are currently working and has the highest number of hours scheduled in the system since they started working as a nurse. Details include first-name, last-name, first day of work as a nurse, date of birth, email address, and total number of hours scheduled.

```

SELECT E.FName, E.LName, MIN(R.Start_Date) AS First_Day_Work, E.DoB, E.Email, SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS Total_Hours
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE R.title = 'Nurse'
AND R.end_date IS NULL
GROUP BY E.EID, E.FName, E.LName, E.DoB, E.Email
HAVING Total_Hours = (
    SELECT MAX(Sub.Total_Hours)
    FROM (
        SELECT E2.EID, SUM(TIMESTAMPDIFF(HOUR, S2.start_time, S2.end_time)) AS Total_Hours
        FROM Employees E2
        JOIN Roles R2 ON E2.EID = R2.EID
        JOIN Schedule S2 ON E2.EID = S2.EID
        WHERE R2.title = 'Nurse'
        AND (R2.end_date IS NULL OR R2.end_date > CURRENT_DATE)
        GROUP BY E2.EID
    ) Sub
)
ORDER BY E.FName, E.LName;

```

FName	LName	First_Day_Work	DoB	Email	Total_Hours
Noah	Lee	2021-03-15	1981-02-12	noahlee81@gmail.com	93

Q16. Get details of the nurse(s) or the doctor(s) who are currently working and has been infected by COVID-19 at least three times. Details include first-name, last-name, first day of work as a nurse or as a doctor, role (nurse/doctor), date of birth, email address, and total number of hours scheduled. Results should be displayed sorted in ascending order by role, then by first name, then by last name.

```

SELECT E.FName, E.LName, MIN(R.Start_Date) AS First_Day_Work, R.Title as Role, E.DoB, E.Email, SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS Total_Hours
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE (R.Title = 'Nurse' OR R.Title = 'Doctor')
AND R.End_Date IS NULL
AND E.EID IN (SELECT I.EID
FROM Infection I
WHERE I.Nature = 'COVID-19'
GROUP BY I.EID
HAVING COUNT(*) >= 3)
GROUP BY E.EID
ORDER BY R.Title ASC, E.FName ASC, E.LName ASC;

```

[Check](#)

FName	LName	First_Day_Work	Role	DoB	Email	Total_Hours
Ava	Kim	2020-05-15	nurse	1984-09-07	avakim84@hotmail.com	12
Ava	Park	2018-05-15	nurse	1985-09-17	avapark85@gmail.com	12
Ella	Kim	2019-06-15	nurse	1986-06-11	ellakim86@hotmail.com	12
Olivia	Lee	2018-08-01	nurse	1980-12-27	olivialeee80@gmail.com	8
Jacob	Belanger	2019-04-01	doctor	1989-05-09	jacob.belanger89@gmail.com	12

Q17. Get details of the nurse(s) or doctor(s) who are currently working and has never been infected by COVID-19. Details include first-name, last-name, first day of work as a nurse or as a doctor, role (nurse/doctor), date of birth, email address, and total number of hours scheduled. Results should be displayed sorted in ascending order by role, then by first name, then by last name.

```

SELECT E.FName, E.LName, MIN(R.Start_Date) AS First_Day_Work, R.Title as Role, E.DoB, E.Email, SUM(TIMESTAMPDIFF(HOUR, S.start_time, S.end_time)) AS Total_Hours
FROM Employees E
JOIN Roles R ON E.EID = R.EID
JOIN Schedule S ON E.EID = S.EID
WHERE (R.Title = 'Nurse' OR R.Title = 'Doctor')
AND R.End_Date IS NULL
AND E.EID NOT IN (SELECT DISTINCT I.EID
FROM Infection I
WHERE I.Nature = 'COVID-19')
GROUP BY E.EID
ORDER BY R.Title ASC, E.FName ASC, E.LName ASC;

```

[Check](#)

FName	LName	First_Day_Work	Role	DoB	Email	Total_Hours
Chloe	Lee	2019-09-30	nurse	1980-07-20	chloe.lee80@hotmail.com	2

FName	LName	First_Day_Work	Role	DoB	Email	Total_Hours
Daniel	Kim	2020-01-01	nurse	1965-10-22	danielkim78@gmail.com	2
Ethan	Kim	2020-09-01	nurse	1983-02-14	ethankim83@hotmail.com	8
Ethan	Lee	2021-01-01	nurse	1987-02-21	ethanlee87@hotmail.com	12
Isabella	Park	2018-08-01	nurse	1989-06-15	isabella.park89@hotmail.com	2
Jacob	Park	2018-09-30	nurse	1987-10-01	jacobpark87@hotmail.com	8
Liam	Park	2018-05-01	nurse	1982-03-10	liampark82@hotmail.com	12
Lucas	Lee	2022-06-01	nurse	1982-11-01	lucaslee82@gmail.com	2
Noah	Liu	2020-01-01	nurse	1981-11-02	noahliu81@hotmail.com	12
Olivia	Kim	2021-11-01	nurse	1981-12-29	oliviakim81@gmail.com	8
Sophia	Kim	2021-01-01	nurse	1980-08-11	sophiakim80@gmail.com	8
Sophie	Kim	2020-11-01	nurse	1984-06-03	sophiekim84@gmail.com	12
Aiden	Liu	2019-11-01	doctor	1982-04-05	aiden.liu82@hotmail.com	8
Caleb	Park	2018-06-01	doctor	1979-11-20	calebpark79@hotmail.com	8
Chloe	Belanger	2019-06-01	doctor	1983-03-26	chloe.belanger83@hotmail.com	30
Daniel	Gauthier	2019-07-01	doctor	1981-11-27	daniel.gauthier81@gmail.com	2
Ella	Wong	2018-06-01	doctor	1981-07-06	ellawong81@gmail.com	8
Emma	Martin	2018-06-01	doctor	1989-01-10	emmamartin89@hotmail.com	4
Ethan	Tran	2019-03-01	doctor	1980-06-27	ethantran80@gmail.com	18
Liam	Boucher	2020-01-01	doctor	1986-02-18	liamboucher86@gmail.com	75
William	Roy	2018-05-01	doctor	1983-08-14	williamroy83@gmail.com	2

Enter your SQL code:

Enter password

Check