

Main Project - COMP 353

Nematollaah Shiri - Section X

April 14th, 2023

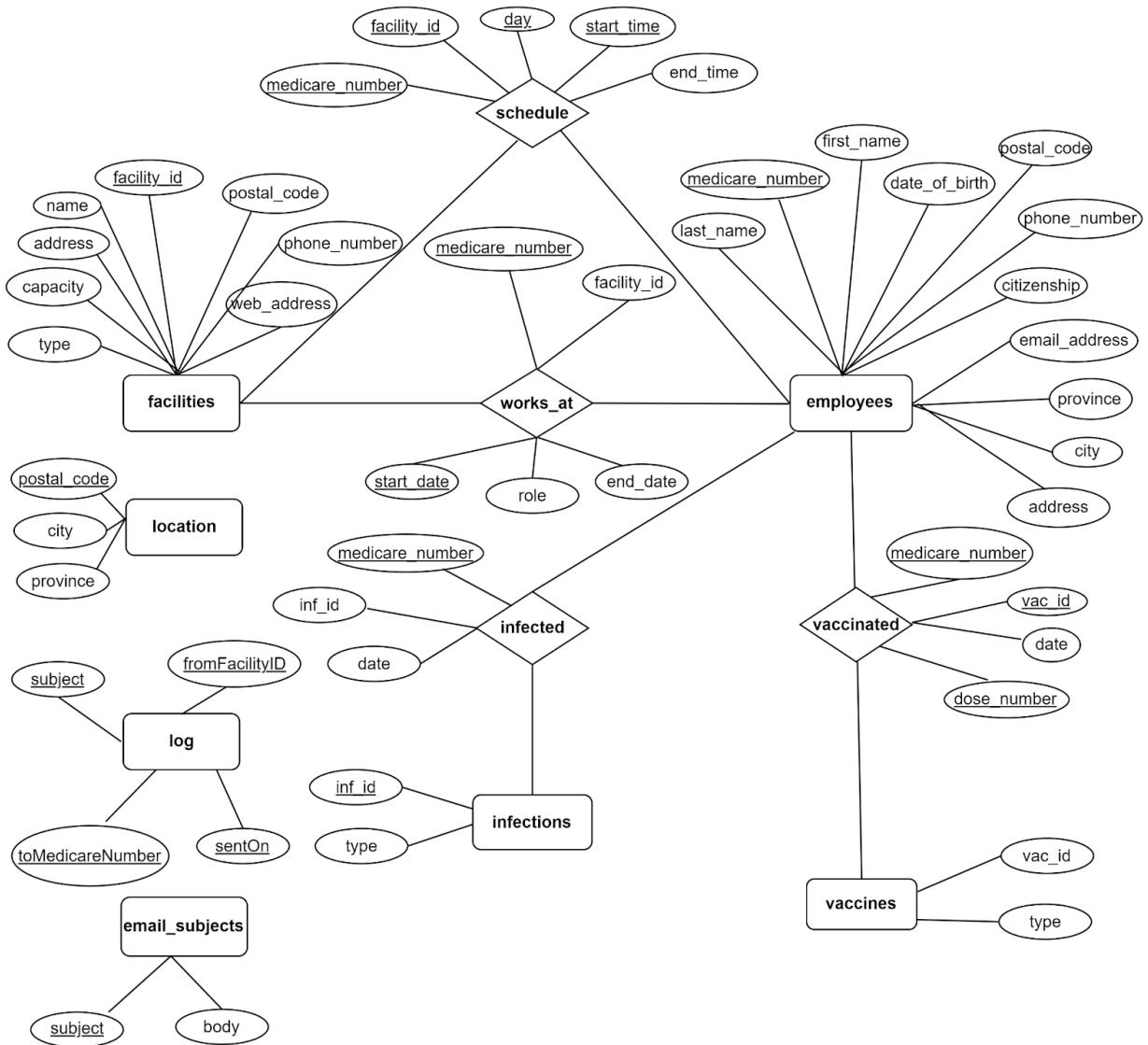
Winter 2023



CONTRIBUTION TABLE

Member	Contribution
Harmanvir Singh 	GUI, Triggers, Relation implementation
Sarabpreet Singh Rekhi 	E/R diagram
Zita Torocsik 	Triggers, Constraints, Queries, Normalization
Natasha Uwase 	Triggers, Constraints, Queries

HFESTS E/R model and design



RELATIONS

facilities(facility_id, name, address, postal_code, phone, web_address, type, capacity)

facility_id	name	address	postal_code	phone_number	web_address	type	capacity
F001	General Hospital	123 Main St	M1B 2K5	4444444444	www.generalHospital.com	Hospital	500
F002	CLSC Saint-Laurent	456 Saint-Laurent Blvd	H2T 2T3	5555555556	www.CLSCSaintLaurent.com	CLSC	2
F003	Family Clinic	789 John St	V6E 2E9	3333333333	www.FamilyClinic.com	Clinic	100
F004	Rexall Pharmacy	111 King St	K1P 5P5	8888888888	www.RexallPharmacy.com	Pharmacy	50
F005	Special Installation	222 Queen St	T2P 2P2	7777777777	www.SpecialInstallation.com	Special Installment	75
F006	Hospital Maisonneuve Rosemont	333 Sher St	H3H 1XY	9999999999	rosemont.com	Hospital	200
F007	Harman Hospital	Harman Road	H2H 1C3	1123409281	harman.com	Clinic	30
F008	Zitta Hospital	Zitta Belle Vue	H2H 1C3	9876543210	zitta4u@zittahospital.com	Hospital	100
F009	Soucy Clinic	28 rue Soucy	J4T 3K5	5674635645	www.clinicsoucy.com	Clinic	50

location(postal_code, city, province)

postal_code	city	province
A1B 2C3	Toronto	ON
H1Z 8G1	Kool	BC
H2H 1C3	Montreal	QC
H2T 2T3	Montreal	QC
H3H 1XY	Montreal	QC
J2G 2A3	Saint Julie	QC
J4T 3K5	Saint Hubert	QC
K1P 5P5	Ottawa	ON
M1B 2K5	Toronto	ON
M1K 2L5	York	ON
T2P 2P2	Calgary	AB
V1E 2C9	Vancouver	BC
V6E 2E9	Vancouver	BC
X1Y 2Z3	Vancouver	BC

employees(first_name, last_name, dob, medicare_number, phone_number, address, postal_code, citizenship, email_address)

medicare_number	first_name	last_name	date_of_birth	phone_number	address	postal_code	citizenship	email_address
M001	John	Doe	1980-01-01	4385559711	123 Main St	M1B 2K5	Canadian	johndoe@email.com
M002	Jane	Doe	1985-01-01	5145555512	456 Saint-Laurent Blvd	H2T 2T3	Canadian	janedoe@email.com
M003	Jim	Smith	1990-01-01	5285858553	789 John St	V6E 2E9	Canadian	jimsmith@email.com
M004	Juan	Pedro	1968-01-01	4388123331	93 Alps St	M1K 2L5	Canadian	juanpedro@email.com
M005	Xiung	Cheng	1983-01-01	7058573332	3 Deuxieme Rang	J2G 2A3	Canadian	xiungcheng@email.com
M006	Fair	Olade	1997-01-01	9614672349	45 Parc Ave	V1E 2C9	Canadian	fairolade@email.com
M007	John	Doe	1990-01-01	1234567890	123 Main St	A1B 2C3	Canadian	johndoe@example.com
M008	James	Mulligan	2000-10-20	1223412331	Concordia Rue	H1Z 8G1	German	jamesy@gmail.com
M012	Jamesy	PJ	1972-10-20	1223412341	11 Palace Road	H1Z 8G1	Lithuanian	jamesy@gmail.com

schedule(medicare_number, facility_id, day, start_time, end_time)

medicare_number	facility_id	day	start_time	end_time
M002	F001	2023-03-16	08:00:00	09:00:00
M002	F002	2023-04-10	08:00:00	17:00:00
M003	F001	2023-03-16	08:00:00	09:00:00
M003	F002	2023-04-10	08:00:00	17:00:00
M004	F001	2023-03-14	08:00:00	09:00:00
M004	F002	2023-04-10	08:00:00	17:00:00

works_at(facility_id, medicare_number, start_date, end_date, role)

```
mysql> select * from works_at;
+-----+-----+-----+-----+-----+
| facility_id | medicare_number | start_date | end_date | role |
+-----+-----+-----+-----+-----+
| F001 | M001 | 2020-01-01 | NULL | Doctor |
| F002 | M003 | 2021-01-01 | NULL | Receptionist |
| F002 | M007 | 2022-01-01 | NULL | Doctor |
| F003 | M004 | 2020-01-01 | 2021-12-31 | Doctor |
| F004 | M005 | 2022-01-01 | NULL | Pharmacist |
| F004 | M012 | 2023-02-02 | NULL | general manager |
| F005 | M006 | 2021-01-01 | 2022-12-31 | Security |
| F006 | M002 | 2020-01-01 | 2022-12-31 | Nurse |
| F006 | M002 | 2021-03-02 | NULL | Nurse |
| F006 | M006 | 2023-01-05 | NULL | Security |
| F006 | M007 | 2020-01-01 | 2021-12-31 | Doctor |
| F006 | M008 | 2022-01-01 | NULL | Pharmacist |
+-----+-----+-----+-----+-----+
12 rows in set (0.01 sec)
```

vaccines(type, vac_id)

```
mysql> select * from vaccines;
+-----+-----+
| vac_id | type |
+-----+-----+
| A1B2C3 | COVID-19 |
| D4E5F6 | Influenza |
| G7H8I9 | Measles |
| J0K1L2 | COVID-19 |
+-----+-----+
4 rows in set (0.00 sec)
```

vaccinated(vac_id, medicare_number, date, dose_number)

```
mysql> select * from vaccinated;
+-----+-----+-----+-----+
| vac_id | medicare_number | date | dose_number |
+-----+-----+-----+-----+
| A1B2C3 | M005 | 2020-12-08 | 1 |
| A1B2C3 | M005 | 2020-12-23 | 2 |
| A1B2C3 | M006 | 2022-03-14 | 2 |
| A1B2C3 | M006 | 2021-03-29 | 3 |
| A1B2C3 | M008 | 2020-11-04 | 1 |
| A1B2C3 | M008 | 2020-11-19 | 2 |
| D4E5F6 | M004 | 2022-02-21 | 1 |
| D4E5F6 | M006 | 2021-12-14 | 1 |
| D4E5F6 | M008 | 2022-05-04 | 1 |
| J0K1L2 | M002 | 2023-01-01 | 1 |
| J0K1L2 | M003 | 2021-01-01 | 1 |
| J0K1L2 | M003 | 2021-01-16 | 2 |
| J0K1L2 | M003 | 2022-02-16 | 3 |
| J0K1L2 | M003 | 2023-01-12 | 4 |
| J0K1L2 | M004 | 2022-12-31 | 1 |
| J0K1L2 | M004 | 2021-03-08 | 2 |
| J0K1L2 | M004 | 2021-03-24 | 3 |
| J0K1L2 | M004 | 2023-01-01 | 4 |
+-----+-----+-----+-----+
18 rows in set (0.00 sec)
```

infections(type, inf_id)

```
mysql> select * from infections;
+-----+-----+
| inf_id | type |
+-----+-----+
| M3N4O5 | COVID-19 |
| X1Y2Z3 | Flu |
+-----+-----+
2 rows in set (0.00 sec)
```

infected(inf_id, medicare_number, date)

```
[mysql> select * from infected;
+-----+-----+-----+
| inf_id | medicare_number | date      |
+-----+-----+-----+
| M3N405 | M001          | 2023-04-10 |
| M3N405 | M003          | 2022-11-25 |
| M3N405 | M003          | 2022-12-17 |
| M3N405 | M003          | 2023-03-17 |
| M3N405 | M004          | 2023-03-15 |
| M3N405 | M004          | 2023-04-12 |
| X1Y2Z3 | M004          | 2022-01-01 |
| X1Y2Z3 | M006          | 2022-08-14 |
| M3N405 | M007          | 2023-04-10 |
| X1Y2Z3 | M008          | 2022-04-04 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

log(fromFacilityID, toMedicareNumber, subject, sentOn)

```
[mysql> select * from log;
+-----+-----+-----+-----+
| fromFacilityID | toMedicareNumber | subject           | sentOn      |
+-----+-----+-----+-----+
| F001           | M002            | COVID-19 exposure | 2023-03-17 |
| F002           | M002            | COVID-19 exposure | 2023-04-12 |
| F002           | M003            | COVID-19 exposure | 2023-04-12 |
| F002           | M004            | COVID-19 exposure | 2023-03-10 |
| F002           | M005            | COVID-19 exposure | 2023-03-10 |
| F002           | M007            | COVID-19 exposure | 2023-03-10 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

emailSubjects(subject, body)

```
[mysql> select * from emailSubjects;
+-----+-----+
| subject      | body          |
+-----+-----+
| COVID-19 exposure | You have been in contact with someone who tested positive for COVID-19. Take steps to avoid infection. |
+-----+-----+
1 row in set (0.00 sec)
```

emailSubjects(subject,

Database Normalization

facilities(facility_id, name, address, postal_code, phone, web_address, type, capacity)

FDs: {facility_id → name, facility_id → address, facility_id → postal_code, facility_id → phone, facility_id → web_address, facility_id → type, facility_id → capacity, phone → facility_id, web_address → facility_id}

CKs: facility_id, phone, web_address.

All FDs are in the form X → Y where X is a superkey; therefore, facilities is in BCNF.

location(postal_code, city, province)

FDs: {postal_code → city, postal_code → province}

CKs: postal_code

All FDs are in the form X → Y where X is a superkey; therefore, location is in BCNF.

employees(first_name, last_name, dob, medicare_number, phone_number, address, postal_code, citizenship, email_address)

FDs: {medicare_number → (to all attributes), phone_number → medicare_number, email_address → medicare_number}

CKs: medicare_number, phone_number, email_address

All FDs are in the form X → Y where X is a superkey; therefore, employees is in BCNF.

schedule(medicare_number, facility_id, day, start_time, end_time)

FDs: {(medicare_number)(facility_id)(day)(start_time) → end_time}

CKs: (medicare_number)(facility_id)(day)(start_time)

All FDs are in the form X → Y where X is a superkey; therefore, schedule is in BCNF.

works_at(facility_id, medicare_number, start_date, end_date, role)

FDs: {(medicare_number)(facility_id)(day)(start_date) → end_date,

{medicare_number)(facility_id)(day)(start_date) → role}

CKs: (medicare_number)(facility_id)(day)(start_date)

All FDs are in the form X → Y where X is a superkey; therefore, works_at is in BCNF.

vaccines(type, vac_id)

CKs: vac_id

FDs: {vac_id → type}

All FDs are in the form X → Y where X is a superkey; therefore, vaccines is in BCNF.

vaccinated(vac_id, medicare_number, date, dose_number)

FDs: {(medicare_number)(vac_id)(dose_number) → date}

CKs: (medicare_number)(vac_id)(dose_number)

All FDs are in the form $X \rightarrow Y$ where X is a superkey; therefore, vaccinated is in BCNF.

infections(type, inf_id)

CKs: inf_id

FDs: {inf_id \rightarrow type}

All FDs are in the form $X \rightarrow Y$ where X is a superkey; therefore, vaccines is in BCNF.

infected(inf_id, medicare_number, date)

Only trivial FDs, in BCNF

log(fromFacilityID, toMedicareNumber, subject, sentOn)

Only trivial FDs, in BCNF

email_subjects(subject, body)

CKs: subject

FDs: {inf_id \rightarrow type}

All FDs are in the form $X \rightarrow Y$ where X is a superkey; therefore, vaccines is in BCNF.

TRIGGER CODES

TRIGGER TO CHECK FACILITY CAPACITY

```
DELIMITER $$  
CREATE TRIGGER check_capacity  
BEFORE INSERT ON works_at  
FOR EACH ROW  
BEGIN  
    DECLARE total_employees INT;  
    SELECT COUNT(*) INTO total_employees  
    FROM works_at  
    WHERE facility_id = NEW.facility_id  
        AND (end_date IS NULL);  
    IF total_employees >= (SELECT capacity FROM facilities WHERE facility_id = NEW.facility_id)  
    THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Cannot add employee to facility, facility is at capacity.';  
    END IF;  
END $$  
DELIMITER;
```

TRIGGER TO CHECK ONE GENERAL MANAGER

```
DELIMITER $$  
CREATE TRIGGER check_general_manager  
BEFORE INSERT ON works_at  
FOR EACH ROW  
BEGIN  
    DECLARE general_manager_count INT;  
    SELECT COUNT(*) INTO general_manager_count  
    FROM works_at  
    WHERE facility_id = NEW.facility_id  
        AND role = 'general manager'  
        AND end_date IS NULL;  
    IF general_manager_count > 0 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Cannot add or update employee, facility already has a general manager.';  
    END IF;  
END $$
```

TRIGGER TO CHECK THE 1 HOUR WINDOW BETWEEN SHIFTS.

```
DELIMITER $$  
CREATE TRIGGER check_employee_schedule  
BEFORE INSERT ON schedule  
FOR EACH ROW  
BEGIN
```

```

-- Check if the employee is already scheduled to work at another facility during the same time
IF EXISTS (
    SELECT * FROM schedule
    WHERE medicare_number = NEW.medicare_number
    AND day = NEW.day
    AND start_time <= NEW.end_time
    AND end_time > (NEW.start_time - INTERVAL 1 HOUR)
) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Employee cannot be scheduled due to a
time conflict.';
END IF;
END;;

```

**TRIGGER TO ASSURE THAT NO employee IS SCHEDULED BEFORE 14 DAYS
AFTER INFECTION**

```

DELIMITER $$

CREATE TRIGGER check_infected
BEFORE INSERT ON schedule
FOR EACH ROW
BEGIN

    DECLARE infected BOOLEAN;
    SELECT EXISTS (SELECT * FROM infected WHERE medicare_number =
    NEW.medicare_number AND date >= DATE_SUB(NEW.day, INTERVAL 14 DAY) AND
    inf_id = 'M3N4O5') INTO infected;

    IF EXISTS (SELECT * FROM works_at WHERE medicare_number =
    NEW.medicare_number AND end_date IS NULL) AND infected THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Cannot schedule employee, employee is infected with
COVID-19!';
    END IF;
END;;

```

**TRIGGER TO NOT SCHEDULE EMPLOYEES IF THEY ARE NOT PROPERLY
VACCINATED**

```

DELIMITER $$

CREATE TRIGGER check_vaccinated
BEFORE INSERT ON schedule
FOR EACH ROW
BEGIN

```

```

IF NOT EXISTS (SELECT * FROM vaccinated WHERE medicare_number =
NEW.medicare_number AND date >= DATE_SUB(NEW.day, INTERVAL 6 MONTH) AND
vac_id = 'J0K1L2') THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Cannot schedule employee, employee is not vaccinated for
COVID-19.';
END IF;
END;$$

```

**TRIGGER TO CANCEL SHIFTS OF CONTAMINATED EMPLOYEES AND NOTIFY
OTHERS NURSES OR DOCTORS SCHEDULED AT THE SAME TIME**

DELIMITER \$\$

```

CREATE TRIGGER cancel_assignments
AFTER INSERT ON infected
FOR EACH ROW
BEGIN
    IF NEW.inf_id = 'M3N4O5' THEN
        -- cancel the infected employee's assignments
        DELETE FROM schedule
        WHERE medicare_number = NEW.medicare_number
        AND day >= NEW.date AND day <= DATE_ADD(NEW.date, INTERVAL 13 DAY);
        INSERT INTO log(fromFacilityID, toMedicareNumber, subject, sentOn)
        (SELECT DISTINCT s.facility_id, s.medicare_number, 'COVID-19 exposure', NEW.date
        FROM schedule AS s
        WHERE s.medicare_number != NEW.medicare_number
        AND s.day >= DATE_SUB(NEW.date, INTERVAL 13 DAY) AND s.day <= NEW.date
        AND s.day IN (SELECT day FROM schedule WHERE medicare_number =
        NEW.medicare_number AND day >= DATE_SUB(NEW.date, INTERVAL 13 DAY) AND day
        <= NEW.date)
        AND s.facility_id IN (SELECT facility_id FROM schedule WHERE medicare_number =
        NEW.medicare_number AND day >= DATE_SUB(NEW.date, INTERVAL 13 DAY) AND day
        <= NEW.date))
    );
    END IF;
END;$$

```

EMAIL SCHEDULES EVERY SUNDAY FOR EMPLOYEES WITH SHIFTS

DELIMITER \$\$

```

CREATE EVENT schedule_mailer
ON SCHEDULE

```

```
EVERY 1 WEEK
STARTS '2023-01-01 00:00:00'
ON COMPLETION PRESERVE
DO
IF WEEKDAY(NOW()) = 6 THEN
    INSERT INTO log(fromFacilityID, toMedicareNumber, subject, sendOn)
        (SELECT DISTINCT facility_id, medicare_number, 'Weekly Schedule', CURDATE()
         FROM schedule WHERE day BETWEEN DATE_ADD(CURDATE(), INTERVAL 1 DAY)
AND DATE_ADD(CURDATE(), INTERVAL 6 DAY));
END IF;$$
```

EMAIL SCHEDULES EVERY SUNDAY FOR EMPLOYEES WITHOUT SHIFTS

```
DELIMITER $$

CREATE EVENT noschedule_mailer
ON SCHEDULE
EVERY 1 WEEK
STARTS '2023-01-01 00:00:00'
ON COMPLETION PRESERVE
DO
IF WEEKDAY(NOW()) = 6 THEN
    INSERT INTO log(fromFacilityID, toMedicareNumber, subject, sendOn)
        (SELECT DISTINCT facility_id, medicare_number, 'No ASSIGNMENT', CURDATE()
         FROM schedule WHERE NOT(day BETWEEN DATE_ADD(CURDATE(), INTERVAL 1
DAY) AND DATE_ADD(CURDATE(), INTERVAL 6 DAY)));
END IF;$$
```

CONSTRAINT/TRIGGER DEMONSTRATIONS

TRIGGER TO CHECK FACILITY CAPACITY

facility_id	name	address	postal_code	phone_number	web_address	the trigger(s) used	type	capacity
F001	General Hospital	123 Main St	M1B 2K5	4444444444	www.generalHospital.com	Facility Capacity Trigger	Hospital	500
F002	CLSC Saint-Laurent	456 Saint-Laurent Blvd	H2T 2T3	5555555556	www.CLSCSaintLaurent.com	Facility Capacity Trigger	CLSC	2

facility_id	medicare_number	start_date	end_date	role
F001	M001	2020-01-01	NULL	Doctor
F002	M003	2021-01-01	NULL	Receptionist
F002	M007	2022-01-01	NULL	Doctor

F002 has a capacity of 2. There are already two people working there. We try inserting another.

Insert into works_at values ('F002', 'M004', '2023-01-01', NULL, 'Doctor');

```
mysql> Insert into works_at values ('F002', 'M004', '2023-01-01', NULL, 'Doctor');
ERROR 1644 (45000): Cannot add employee to facility, facility is at capacity.
```

TRIGGER TO CHECK ONE GENERAL MANAGER

Facility F004 already has a general manager, we try add another one.

```
mysql> Insert into works_at values ('F004', 'M008', '2023-01-01', NULL, 'general manager');
ERROR 1644 (45000): Cannot add or update employee, facility already has a general manager.
```

TRIGGER TO CHECK THE 1 HOUR WINDOW BETWEEN SHIFTS.

Insert into schedule values ('M004', 'F002', '2023-04-15', '08:00', '17:00');

inf_id	medicare_number	date
A2X4B5	M009	2022-01-02

medicare_number	facility_id	day	start_time	end_time
M002	F001	2023-03-16	08:00:00	09:00:00
M003	F001	2023-03-16	08:00:00	09:00:00
M004	F001	2023-03-14	08:00:00	09:00:00
M004	F002	2023-04-15	08:00:00	17:00:00

Inserting a conflicting schedule (not leaving 1 hour between shifts):

Insert into schedule values ('M004', 'F002', '2023-04-15', '17:00', '19:00');

```
mysql> Insert into schedule values ('M004', 'F002', '2023-04-15', '17:00', '19:00');
ERROR 1644 (45000): Employee cannot be scheduled due to a time conflict.
```

TRIGGER TO ASSURE THAT NO employee IS SCHEDULED BEFORE 14 DAYS AFTER INFECTION

```
mysql> select * from
+-----+-----+
| inf_id | type   |
+-----+-----+
| X1Y2Z3 | Flu    |
| M3N4O5 | COVID-19 |
+-----+-----+
| M3N4O5 | M003      | 2023-03-17 | ,
```

M003 was infected 2023-03-17.

We try to insert a shift on 2023-03-20. This is not two weeks after infections—should not work.
Insert into schedule values ('M003', 'F002', '2023-03-20', '08:00', '17:00');

```
mysql> Insert into schedule values ('M003', 'F002', '2023-03-20', '08:00', '17:00');
ERROR 1644 (45000): Cannot schedule employee, employee is infected with COVID-19.
```

TRIGGER TO NOT SCHEDULE EMPLOYEES IF THEY ARE NOT PROPERLY VACCINATED

M001 does not have a Covid vaccine. We will try to schedule that person.

Insert into schedule values ('M001', 'F002', '2023-03-20', '08:00', '17:00');

```
mysql> Insert into schedule values ('M001', 'F002', '2023-03-20', '08:00', '17:00');
ERROR 1644 (45000): Cannot schedule employee, employee is not vaccinated for COVID-19.
```

TRIGGER TO CANCEL SHIFTS OF CONTAMINATED EMPLOYEES AND NOTIFY OTHERS NURSES OR DOCTORS SCHEDULED AT THE SAME TIME

Insert into schedule values ('M003', 'F002', '2023-04-10', '08:00', '17:00'), ('M002', 'F002', '2023-04-10', '08:00', '17:00'), ('M004', 'F002', '2023-04-10', '08:00', '17:00');

```
mysql> Insert into schedule values ('M003', 'F002', '2023-04-10', '08:00', '17:00'), ('M002', 'F002', '2023-04-10', '08:00', '17:00'), ('M004', 'F002', '2023-04-10', '08:00', '17:00');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

mysql> select * from schedule;

medicare_number	facility_id	day	start_time	end_time
M003	F001	2023-03-16	08:00:00	09:00:00
M004	F001	2023-03-14	08:00:00	09:00:00
M003	F002	2023-04-10	08:00:00	17:00:00
M002	F002	2023-04-10	08:00:00	17:00:00
M004	F002	2023-04-15	08:00:00	17:00:00

4 rows in set (0.00 sec)

OF CONTAMINATED EMPLOYEES AND NOTIFY OTHERS SCHEDULED AT THE SAME TIME

M002, F002, 2023-04-10, 08:00, 17:00, M003, F002, 2023-04-10, 08:00, 17:00, M004, F002, 2023-04-10, 08:00, 17:00;

M003, and M002 should be notified. And The shift of M004 on 04-15 should be removed.

Insert into infected values ('M3N4O5', 'M004', '2023-04-12');

Now M003, M002, and M004 worked together on 04-10. Now a future shift for M004 is in 04-15. We will infect M004 on 04-12.

M003, and M002 should be notified. And The shift of M004 on 04-15 should be removed.

Insert into infected values ('M3N4O5', 'M004', '2023-04-12');

```

mysql> select * from schedule;
+-----+-----+-----+-----+-----+
| medicare_number | facility_id | day   | start_time | end_time |
+-----+-----+-----+-----+-----+
| M002           | F001        | 2023-03-16 | 08:00:00   | 09:00:00   |
| M002           | F002        | 2023-04-10 | 08:00:00   | 17:00:00   |
| M003           | F001        | 2023-03-16 | 08:00:00   | 09:00:00   |
| M003           | F002        | 2023-04-10 | 08:00:00   | 17:00:00   |
| M004           | F001        | 2023-03-14 | 08:00:00   | 09:00:00   |
| M004           | F002        | 2023-04-10 | 08:00:00   | 17:00:00   |
| M004           | F002        | 2023-04-16 | 08:00:00   | 17:00:00   |
+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> Insert into infected values ('M3N405', 'M004', '2023-04-12');
Query OK, 1 row affected (0.00 sec)

mysql> select * from schedule;
+-----+-----+-----+-----+-----+
| medicare_number | facility_id | day   | start_time | end_time |
+-----+-----+-----+-----+-----+
| M002           | F001        | 2023-03-16 | 08:00:00   | 09:00:00   |
| M002           | F002        | 2023-04-10 | 08:00:00   | 17:00:00   |
| M003           | F001        | 2023-03-16 | 08:00:00   | 09:00:00   |
| M003           | F002        | 2023-04-10 | 08:00:00   | 17:00:00   |
| M004           | F001        | 2023-03-14 | 08:00:00   | 09:00:00   |
| M004           | F002        | 2023-04-10 | 08:00:00   | 17:00:00   |
| M004           | F002        | 2023-04-16 | 08:00:00   | 17:00:00   |
+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

```

The future shift is gone. Now we check emails.

```

mysql> select * from log;
+-----+-----+-----+-----+
| fromFacilityID | toMedicareNumber | subject          | sentOn    |
+-----+-----+-----+-----+
| F001           | M002             | COVID-19 exposure | 2023-03-17 |
| F002           | M002             | COVID-19 exposure | 2023-04-12 |
| F002           | M003             | COVID-19 exposure | 2023-04-12 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

M002 and M003 were notified.

GUI

Public Health Care System

List of Tables in the Database

employees

facilities

infected

infections

location

log

schedule

vaccinated

vaccines

works_at

Employees

[Back to Home Page](#)

[Create a New Entry](#)

Medicare Number	First Name	Last Name	Date of Birth	Phone Number	Address	Postal Code	Citizenship	Email Adress	Actions
M001	John	Doe	1980-01-01	4385559711	123 Main St	M1B 2K5	Canadian	johndoe@email.com	<button>DELETE</button>
M002	Jane	Doe	1985-01-01	514555512	456 Saint-Laurent Blvd	H2T 2T3	Canadian	janedoe@email.com	<button>DELETE</button>
M003	Jim	Smith	1990-01-01	5285858553	789 John St	V6E 2E9	Canadian	jimsmith@email.com	<button>DELETE</button>
M004	Juan	Pedro	1968-01-01	4388123331	93 Alps St	M1K 2L5	Canadian	juanpedro@email.com	<button>DELETE</button>
M005	Xiung	Cheng	1983-01-01	7058573332	3 Deuxième Rang	J2G 2A3	Canadian	xiungcheng@email.com	<button>DELETE</button>
M006	Fair	Olaade	1997-01-01	9614672349	45 Parc Ave	V1E 2C9	Canadian	fairolaade@email.com	<button>DELETE</button>
M007	John	Doe	1990-01-01	1234567890	123 Main St	A1B 2C3	Canadian	johndoe@example.com	<button>DELETE</button>

Employees

[Back to Home Page](#)

Insert Employee Data

Enter Data to Store

Medicare Number:

First Name:

Last Name:

Date of Birth:

yyyy - mm - dd



Phone Number:

Address:

Postal Code:

Citizenship:

Email Address:

SQL scripts to create relations

```
CREATE TABLE facilities (
    facility_id char(10) PRIMARY KEY,
    name varchar(255) not null,
    address varchar(255) not null,
    postal_code char(7) not null,
    phone_number char(10) not null,
    web_address varchar(255),
    type varchar(255) not null,
    capacity int not null,
    FOREIGN KEY (postal_code) references location(postal_code);
```

```
CREATE TABLE location (
    postal_code char(7) PRIMARY KEY,
    city varchar(35) not null,
    province char(2) not null);
```

```
CREATE TABLE works_at (
    facility_id char(10),
    medicare_number char(10),
    start_date date not null,
    end_date varchar(20),
    role varchar(255) not null,
    PRIMARY KEY(medicare_number, start_date),
    FOREIGN KEY (medicare_number) REFERENCES employees(medicare_number),
    FOREIGN KEY (facility_id) REFERENCES facilities(facility_id);
```

```
CREATE TABLE employees(
    medicare_number char(10) PRIMARY KEY,
    first_name varchar(255) not null,
    last_name varchar(255) not null,
    date_of_birth date not null,
    phone_number char(10) not null,
    address varchar(255) not null,
    postal_code char(7) not null,
    citizenship varchar(255),
    email_address varchar(255) not null,
    FOREIGN KEY (postal_code) references location(postal_code));
```

```
CREATE TABLE vaccines(
    vac_id char(10) PRIMARY KEY,
    type char(10));
```

```
CREATE TABLE vaccinated(
    vac_id char(10),
    medicare_number char(10),
```

```
    date date not null,  
    dose_number int not null,  
PRIMARY KEY (medicare_number, vac_id, dose_number),  
FOREIGN KEY (vac_id) references vaccines(vac_id));
```

```
CREATE TABLE infected(  
    inf_id char(10),  
    medicare_number char(10),  
    date date not null,  
PRIMARY KEY (inf_id, medicare_number, date),  
FOREIGN KEY (inf_id) references infections(inf_id),  
FOREIGN KEY (medicare_number) references employees(medicare_number));
```

```
CREATE TABLE infections(  
    inf_id char(10) PRIMARY KEY,  
    type varchar(255));
```

```
CREATE TABLE schedule(  
    medicare_number char(10),  
    facility_id char(10),  
    day date,  
    start_time time,  
    end_time time,  
    CHECK(start_time < end_time),  
    FOREIGN KEY (medicare_number) REFERENCES employees(medicare_number),  
    FOREIGN KEY (medicate_number) references works_at(medicare_number),  
    FOREIGN KEY (facility_id) references works_at(facility_id),  
    FOREIGN KEY (facility_id) REFERENCES facilities(facility_id),  
    PRIMARY KEY(medicare_number, facility_id, day, start_time));
```

```
CREATE TABLE log(  
    fromFacilityID varchar(10),  
    toMedicareNumber varchar(10),  
    subject varchar(255),  
    sentOn date,  
    PRIMARY KEY(fromFacilityID, toMedicareNumber, subject, sentOn)  
    FOREIGN KEY (fromFacilityID) references facilities(facility_id)  
    FOREIGN KEY (toMedicateNumber) references employees(medicare_number)  
    FOREIGN KEY (subject) references emailSubjects(subject));
```

```
CREATE TABLE emailSubjects(  
    subject varchar(255) PRIMARY KEY,  
    body varchar(255));
```

QUERIES

6. 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 name, address, location.city, location.province, facilities.postal_code, phone_number,
web_address,
type, capacity, works_at.facility_id, COUNT(works_at.facility_id) AS 'number_Of_Employees' FROM
facilities, works_at, location
WHERE facilities.postal_code = location.postal_code AND facilities.facility_id = works_at.facility_id
GROUP BY works_at.facility_id
ORDER BY location.province, location.city, type, number_Of_Employees;
```

name	address	city	province	postal_code	phone_number	web_address	type	capacity	facility_id	number_of_employees
Special Installation	222 Queen St	Calgary	AB	T2P 2P2	7777777777	www.SpecialInstallation.com	Special Installment	75	F005	1
Family Clinic	789 John St	Vancouver	BC	V6E 2E9	3333333333	www.FamilyClinic.com	Clinic	100	F003	1
Rexall Pharmacy	111 King St	Ottawa	ON	K1P 5P5	8888888888	www.RexallPharmacy.com	Pharmacy	50	F004	2
General Hospital	123 Main St	Toronto	ON	M1B 2K5	4444444444	www.GeneralHospital.com	Hospital	500	F001	1
CLSC Saint-Laurent	456 Saint-Laurent Blvd	Montreal	QC	H2T 2T3	5555555556	www.CLSCSaintLaurent.com	CLSC	2	F002	2
Hospital Maisonneuve Rosemont	333 Sher St	Montreal	QC	H3H 1XY	9999999999	rosemont.com	Hospital	200	F006	5

7. 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.

```
SELECT CONCAT(first_name, " ", last_name) AS 'name', start_date, date_of_birth,
employees.medicare_number, employees.phone_number, employees.postal_code, citizenship,
employees.email_address, works_at.facility_id, role
FROM works_at, employees
WHERE works_at.medicare_number = employees.medicare_number GROUP BY works_at.facility_id
ORDER BY role, first_name, last_name;
```

name	start_date	date_of_birth	medicare_number	phone_number	postal_code	citizenship	email_address	facility_id	role
John Doe	2020-01-01	1980-01-01	M001	4388559711	M1B 2K5	Canadian	john.doe@email.com	F001	Doctor
Juan Pedro	2020-01-01	1968-01-01	M004	4388123331	M1K 2L5	Canadian	juanpedro@email.com	F003	Doctor
Jane Doe	2020-01-01	1985-01-01	M002	5145555512	H2T 2T3	Canadian	jane.doe@email.com	F006	Nurse
Xiung Cheng	2022-01-01	1983-01-01	M005	7058573332	J2G 2A3	Chinese	xiungcheng@email.com	F004	Pharmacist
Jim Smith	2021-01-01	1990-01-01	M003	5285858553	V6E 2E9	Canadian	jim.smith@email.com	F002	Receptionist
Fair Olade	2021-01-01	1997-01-01	M006	9614672349	V1E 2C9	Canadian	fair.olade@email.com	F005	Security

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, the by start time.

```
SELECT name, day, start_time, end_time from schedule, facilities
WHERE schedule.medicare_number = 'M002' AND schedule.facility_id = facilities.facility_id
ORDER BY name, day, start_time;
```

```
mysql> SELECT name, day, start_time, end_time from schedule, facilities
-> WHERE schedule.medicare_number = 'M002' AND schedule.facility_id = facilities.facility_id
-> ORDER BY name, day, start_time;
+-----+-----+-----+-----+
| name | day | start_time | end_time |
+-----+-----+-----+-----+
| CLSC Saint-Laurent | 2023-04-10 | 08:00:00 | 17:00:00
| General Hospital | 2023-03-16 | 08:00:00 | 09:00:00
+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

9. Get details of all the doctors who have been infected by COVID-19 in the past two weeks. Details include 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 firstname of the doctor.

```
SELECT first_name, last_name, works_at.facility_id, works_at.medicare_number, works_at.role, inf_id,
date, name FROM (
infected INNER JOIN works_at ON works_at.medicare_number = infected.medicare_number)
INNER JOIN employees ON works_at.medicare_number = employees.medicare_number)
INNER JOIN facilities ON works_at.facility_id = facilities.facility_id)
WHERE inf_id = 'M3N4O5' AND role = 'Doctor' AND date >= DATE_SUB(curdate(), INTERVAL 13
DAY) AND date < CURDATE()
ORDER BY name, first_name;
```

```
mysql> SELECT first_name, last_name, works_at.facility_id, works_at.medicare_number, works_at.role, inf_id, date, name FROM (
-> (infected INNER JOIN works_at ON works_at.medicare_number = infected.medicare_number) past
-> INNER JOIN employees ON works_at.medicare_number = employees.medicare_number) infection, and
-> INNER JOIN facilities ON works_at.facility_id = facilities.facility_id)ng for. Results should
-> WHERE inf_id = 'M3N4O5' AND role = 'Doctor' AND date >= DATE_SUB(curdate(), INTERVAL 13 DAY) AND date < CURDATE()
-> ORDER BY name, first_name; of the doctor.
+-----+-----+-----+-----+-----+-----+-----+
| first_name | last_name | facility_id | medicare_number | role | inf_id | date | name |
+-----+-----+-----+-----+-----+-----+-----+
| John | Doe | F002 | M007 | Doctor | M3N4O5 | 2023-04-10 | CLSC Saint-Laurent
| Juan | Pedro | F003 | M004 | Doctor | M3N4O5 | 2023-04-12 | Family Clinic
| John | Doe | F001 | M001 | Doctor | M3N4O5 | 2023-04-10 | General Hospital
| John | Doe | F006 | M007 | Doctor | M3N4O5 | 2023-04-10 | Hospital Maisonneuve Rosemont
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

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

```
SELECT fromFacilityID, subject from log
```

```
WHERE fromFacilityID = 'F002'  
ORDER BY sentOn;
```

```
mysql> SELECT fromFacilityID, subject from log  
    -> WHERE fromFacilityID = 'F002'  
    -> ORDER BY sentOn;  
+-----+-----+  
| fromFacilityID | subject      |  
+-----+-----+  
| F002          | COVID-19 exposure |  
+-----+-----+  
5 rows in set (0.00 sec)
```

```
SELECT from  
WHERE from  
ORDER BY s
```

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.

```
SELECT first_<br/>(schedule INNER JOIN works_at ON schedule.facility_id = works_at.facility_id AND (role = 'Doctor' OR role = 'Nurse') AND day >= DATE_SUB(curdate(), INTERVAL 13 DAY) AND day < curdate() ORDER BY role, first_name;
```

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.

```
SELECT distinct first_name, last_name, role FROM (  
(schedule INNER JOIN works_at ON schedule.medicare_number = works_at.medicare_number)  
INNER JOIN employees ON works_at.medicare_number = employees.medicare_number)  
WHERE works_at.facility_id = 'F006' AND (role = 'Doctor' OR role = 'Nurse') AND day >= DATE_SUB(curdate(), INTERVAL 13 DAY) AND day < curdate()  
ORDER BY role, first_name;
```

```
mysql> SELECT distinct first_name, last_name, role FROM (  
    -> (schedule INNER JOIN works_at ON schedule.medicare_number = works_at.medicare_number)  
    -> INNER JOIN employees ON works_at.medicare_number = employees.medicare_number)  
    -> WHERE works_at.facility_id = 'F006' AND (role = 'Doctor' OR role = 'Nurse') AND day >= curdate()  
    -> ORDER BY role, first_name;  
+-----+-----+-----+  
| first_name | last_name | role   |  
+-----+-----+-----+  
| Fair       | Olade     | Doctor |  
| James      | Mulligan  | Doctor |  
| Jane       | Doe       | Nurse  |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```

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.

```
SELECT works_at.role, TRUNCATE(SUM(end_time - start_time)/10000, 0) AS 'total_number_of_hours_per_role' FROM schedule
```

```

INNER JOIN works_at ON schedule.medicare_number = works_at.medicare_number
WHERE works_at.facility_id = 'F006' AND schedule.day >= '2023-04-01' AND schedule.day <
'2023-04-30'
GROUP by role;

```

```

mysql> SELECT works_at.role,TRUNCATE(SUM(end_time - start_time)/10000, 0) AS 'total_number_Of_Hours_per_Role' FROM schedule
-> INNER JOIN works_at ON schedule.medicare_number = works_at.medicare_number
-> WHERE works_at.facility_id = 'F006' AND schedule.day >= '2023-04-01' AND schedule.day < '2023-04-30'
-> GROUP by role;
+-----+-----+
| role | total_number_Of_Hours_per_Role |
+-----+-----+
| Nurse | 36 |
| Doctor | 36 |
+-----+
2 rows in set (0.01 sec)

```

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 name, province, capacity, COUNT(works_at.facility_id) AS 'number_Of_Employees_Infected'
FROM (
(facilities INNER JOIN location ON facilities.postal_code = location.postal_code)
INNER JOIN works_at ON works_at.facility_id = facilities.facility_id)
INNER JOIN infected ON works_at.medicare_number = infected.medicare_number)
WHERE inf_id = 'M3N4O5'
AND date <= DATE_SUB(curdate(), INTERVAL 13 DAY)
ORDER BY province, number_Of_Employees_Infected;

```

```

mysql> SELECT name, province, capacity, COUNT(works_at.facility_id) AS 'number_Of_Employees_Infected' FROM (
-> (facilities INNER JOIN location ON facilities.postal_code = location.postal_code)
-> INNER JOIN works_at ON works_at.facility_id = facilities.facility_id)
-> INNER JOIN infected ON works_at.medicare_number = infected.medicare_number)
-> WHERE inf_id = 'M3N4O5'
-> AND date <= DATE_SUB(curdate(), INTERVAL 13 DAY)
-> ORDER BY province, number_Of_Employees_Infected;
+-----+-----+-----+
| name | province | capacity | number_Of_Employees_Infected |
+-----+-----+-----+
| CLSC Saint-Laurent | QC | 2 | 4 |
+-----+
1 row in set (0.00 sec)

```

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 first_name, last_name, location.city, COUNT(works_at.facility_id) AS
'number_Of_Establishments_Of_Employment'

```

```

FROM (((works_at INNER JOIN employees ON works_at.medicare_number =
employees.medicare_number)
INNER JOIN facilities ON works_at.facility_id = facilities.facility_id)
INNER JOIN location ON facilities.postal_code = location.postal_code)
WHERE location.province = 'QC' AND role = 'Doctor'
GROUP BY works_at.medicare_number;

```

```

mysql> SELECT first_name, last_name, location.city, COUNT(works_at.facility_id) AS 'number_of_Establishments_of_Employment'
-> FROM (((works_at INNER JOIN employees ON works_at.medicare_number = employees.medicare_number)
-> INNER JOIN facilities ON works_at.facility_id = facilities.facility_id)
-> INNER JOIN location ON facilities.postal_code = location.postal_code)
-> WHERE location.province = 'QC' AND role = 'Doctor'
-> GROUP BY works_at.medicare_number;
+-----+-----+-----+-----+
| first_name | last_name | city | number_of_Establishments_of_Employment |
+-----+-----+-----+-----+
| Jim | Smith | Montreal | 1 |
| John | Doe | Montreal | 2 |
| Fair | Olade | Montreal | 1 |
| James | Mulligan | Montreal | 1 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

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 first_name, last_name, start_date, date_of_birth, email_address, TRUNCATE(MAX(end_time -
start_time)/10000, 0)
AS 'total_number_of_Hours_Scheduled'
FROM ((schedule INNER JOIN works_at ON schedule.medicare_number =
works_at.medicare_number)
INNER JOIN employees ON schedule.medicare_number = employees.medicare_number) WHERE role
= 'Nurse';

```

```

mysql> SELECT first_name, last_name, start_date, date_of_birth, email_address, TRUNCATE(MAX(end_time - start_time)/10000, 0)
-> AS 'total_number_of_Hours_Scheduled'
-> FROM ((schedule INNER JOIN works_at ON schedule.medicare_number = works_at.medicare_number)
-> INNER JOIN employees ON schedule.medicare_number = employees.medicare_number) WHERE role = 'Nurse';
+-----+-----+-----+-----+-----+
| first_name | last_name | start_date | date_of_birth | email_address | total_number_of_Hours_Scheduled |
+-----+-----+-----+-----+-----+
| Jane | Doe | 2020-01-01 | 1985-01-01 | janedoe@email.com | 9 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

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, lastname, 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 first_name, last_name, start_date, role, date_of_birth, email_address,
TRUNCATE(MAX(end_time - start_time)/10000, 0) AS 'total_number_of_Hours_Scheduled'

```

```

FROM (SELECT medicare_number, count(medicare_number) AS frequency FROM infected where
inf_id = 'M3N4O5' GROUP BY medicare_number) AS T1, works_at, employees, schedule
WHERE frequency >=3 AND works_at.medicare_number = T1.medicare_number AND
employees.medicare_number = T1.medicare_number
AND (role = 'Doctor' OR role = 'Nurse') AND day >= DATE_SUB(curdate(), INTERVAL 13 DAY)
ORDER BY role, first_name, last_name;

```

```

mysql> SELECT first_name, last_name, start_date, role, date_of_birth, email_address,
-> TRUNCATE(MAX(end_time - start_time)/10000, 0) AS 'total_number_Of_Hours_Scheduled'
-> FROM (SELECT medicare_number, count(medicare_number) AS frequency FROM infected where inf_id = 'M3N4O5' GROUP BY medicare_number
-> ORDER BY role, first_name, last_name)
-> WHERE frequency >=3 AND works_at.medicare_number = T1.medicare_number AND employees.medicare_number = T1.medicare_number
-> AND (role = 'Doctor' OR role = 'Nurse') AND day >= DATE_SUB(curdate(), INTERVAL 13 DAY)
-> ORDER BY role, first_name, last_name;
+-----+-----+-----+-----+-----+-----+
| first_name | last_name | start_date | role | date_of_birth | email_address | total_number_Of_Hours_Scheduled |
+-----+-----+-----+-----+-----+-----+
| James     | Mulligan | 2022-01-01 | Nurse | 2000-10-20 | jamesy@gmail.com | 9 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

ascending order by role, then by first name, then by last name

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 first_name, last_name, start_date, role, date_of_birth, email_address,
TRUNCATE(MAX(end_time - start_time)/10000, 0) AS 'total_number_Of_Hours_Scheduled'
FROM infected, works_at, employees, schedule
where works_at.medicare_number NOT IN (SELECT medicare_number FROM infected where inf_id =
'M3N4O5')
AND works_at.medicare_number = employees.medicare_number AND works_at.medicare_number =
schedule.medicare_number
AND (role = 'doctor' OR role = 'nurse')
ORDER BY role, first_name, last_name;

```

```

-> FROM infected, works_at, employees, schedule
-> where works_at.medicare_number NOT IN (SELECT medicare_number FROM infected where inf_id = 'M3N4O5')
-> AND works_at.medicare_number = employees.medicare_number AND works_at.medicare_number =
schedule.medicare_number
-> AND (role = 'doctor' OR role = 'nurse')
-> ORDER BY role, first_name, last_name;
+-----+-----+-----+-----+-----+-----+
| first_name | last_name | start_date | role | date_of_birth | email_address | total_number_Of_Hours_Scheduled |
+-----+-----+-----+-----+-----+-----+
| Fair      | Olade    | 2023-01-05 | Doctor | 1997-01-01 | fairolade@email.com | 9 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```