VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Database Systems (CO2014)

Assignment 2

Quarantine Camp Database

Instructor: Phan Trong Nhan
Class: CC03
Authors: Nguyen Thanh Ngan - 1911667
Hoang Nhu Ngoc - 1911697
Vo Khue Tam Uyen - 1852858
Nguyen Thuy Khanh Linh - 1952319
Pham Le The Vinh - 1915948

HO CHI MINH CITY, NOVEMBER 2021

Contents

1 Physical Database Design

1.1 Implementing the database

For this assignment, we decided to choose MySQL as our DBMS (Database Management System). The tables below lists the data type, length of each attribute as well as the constraints associated to them.

- **medication**

Attribute		**Explanation**
Name	code	
Data type	varchar	Code contains both letters and numbers so we choose VARCHAR
Data length	6	Code is exactly 6 character-length
Constraint	NOT NULL	Code is not permitted to be NULL
	PRIMARY KEY	Code is used to identify different medications
Name	expiration_date	
Data type	date	Expiration_date is in form YYYY-MM-DD
Data length		
Constraint	NOT NULL	Expiration_date is not permitted to be NULL
Name	name	
Data type	varchar	Name contains only letters
Data length	100	Name is exactly 100 character-length
Constraint	NOT NULL	Name is not permitted to be NULL
Name	price	
Data type	decimal	Price is presented in form of digits
Data length	(8, 2)	The precision is 8 - the scale is 2 (xxxxxx.xx form)
Constraint	NOT NULL	Price is not permitted to be NULL

- **effect**

Attribute		**Explanation**
Name	medication_code	
Data type	varchar	Medication_code contains both letters and numbers
Data length	6	Medication_code is exactly 6 character-length
Constraint	NOT NULL	Medication_code is not permitted to be NULL
Name	description	
Data type	varchar	Description contains both letters and numbers
Data length	500	Description is exactly 500 character-length
Constraint	NOT NULL	Description is not permitted to be NULL

- **building**

Attribute		**Explanation**
Name	building_number	
Data type	int	Building_number is presented as an integer
Data length		
Constraint	NOT NULL	Building_number is not permitted to be NULL

Attribute		Explanation
Name	camp_number	
Data type	int	Camp_number is presented as an integer
Data length		
Constraint	NOT NULL	Camp_number is not permitted to be NULL

- **camp**

Attribute		Explanation
Name	camp_number	
Data type	int	Camp_number is presented as an integer
Data length		
Constraint	NOT NULL	Camp_number is not permitted to be NULL
	PRIMARY KEY	Camp_number is used to identify different camps
Name	head_of_camp	
Data type	int	Head_of_camp is presented as an integer
Data length		
Constraint	NOT NULL	Head_of_camp is not permitted to be NULL

- **comorbidity**

Attribute		Explanation
Name	patient_number	
Data type	int	patient_number is presented as an integer
Data length		
Constraint	NOT NULL	patient_number is not permitted to be NULL
Name	name	
Data type	varchar	Name contains only letters
Data length	100	Name is exactly 100 character-length
Constraint	NOT NULL	Name is not permitted to be NULL

- **manager, staff, volunteer and nurse**

Attribute		Explanation
Name	personnel_number	
Data type	int	personnel_number is presented as an integer
Data length		
Constraint	NOT NULL	personnel_number is not permitted to be NULL
	PRIMARY KEY	personnel_number is used to identify different doctors
Name	full_name	
Data type	varchar	full_name contains only letters
Data length	100	full_name is exactly 100 character-length
Constraint	NOT NULL	full_name is not permitted to be NULL
Name	responsibility	
Data type	varchar	responsibility contains both letters and numbers so we choose VARCHAR
Data length	500	responsibility is exactly 500 character-length
Constraint	NOT NULL	responsibility is not permitted to be NULL

Attribute		Explanation
Name	camp_number	
Data type	int	Camp_number is presented as an integer
Data length		
Constraint	NOT NULL	

- **doctor**

Attribute		Explanation
Name	personnel_number	
Data type	int	personnel_number is presented as an integer
Data length		
Constraint	NOT NULL	personnel_number is not permitted to be NULL
	PRIMARY KEY	personnel_number is used to identify different doctors
Name	full_name	
Data type	varchar	full_name contains only letters
Data length	100	full_name is exactly 100 character-length
Constraint	NOT NULL	full_name is not permitted to be NULL
Name	responsibility	
Data type	varchar	responsibility contains both letters and numbers so we choose VARCHAR
Data length	500	responsibility is exactly 500 character-length
Constraint	NOT NULL	responsibility is not permitted to be NULL
Name	camp_number	
Data type	int	Camp_number is presented as an integer
Data length		
Constraint		Unlike the other four types of personnel, the camp_number attribute of doctor is nullable so that we can avoid circular reference between doctor and camp.

- **floor**

Attribute		Explanation
Name	floor_number	
Data type	int	floor_number is presented as an integer
Data length		
Constraint	NOT NULL	floor_number is not permitted to be NULL
Name	building_number	
Data type	int	building_number is presented as an integer
Data length		
Constraint	NOT NULL	building_number is not permitted to be NULL
Name	camp_number	
Data type	int	Camp_number is presented as an integer
Data length		
Constraint	NOT NULL	Camp_number is not permitted to be NULL

- **patient**

Attribute		Explanation
Name	patient_number	
Data type	int	patient_number is presented as an integer
Data length		
Constraint	NOT NULL	patient_number is not permitted to be NULL
	PRIMARY KEY	patient_number is used to identify different patients
Name	full_name	
Data type	varchar	full_name contains only letters
Data length	100	full_name is exactly 100 character-length
Constraint	NOT NULL	full_name is not permitted to be NULL
Name	gender	
Data type	varchar	gender contains only letters (M, F)
Data length	1	gender is exactly 1 character-length
Constraint	NOT NULL	gender is not permitted to be NULL
Name	identity_number	
Data type	varchar	identity_number contains only letters
Data length	12	identity_number is exactly 12 character-length
Constraint	NOT NULL	identity_number is not permitted to be NULL
Name	phone	
Data type	varchar	phone contains only letters
Data length	12	phone is exactly 12 character-length
Constraint	NOT NULL	phone is not permitted to be NULL
Name	address	
Data type	varchar	address contains only letters
Data length	500	address is exactly 12 character-length
Constraint	NOT NULL	address is not permitted to be NULL
Name	patient_status	
Data type	varchar	identity_number contains only letters
Data length	7	identity_number is exactly 12 character-length
Constraint	NOT NULL	identity_number is not permitted to be NULL
	DEFAULT 'normal'	patient_status is automatically assigned "normal" if it is not inserted
Name	discharge_date	
Data type	date	discharge_date is in form YYYY-MM-DD
Data length		
Constraint		
Name	admitting_staff	
Data type	int	admitting_staff is presented as an integer
Data length		
Constraint	NOT NULL	admitting_staff is not permitted to be NULL
Name	admission_date	
Data type	date	admission_date is in form YYYY-MM-DD
Data length		
Constraint	NOT NULL	admission_date is not permitted to be NULL
Name	last_location	
Data type	varchar	last_location contains both letters and numbers
Data length	500	last_location is exactly 500 character-length
Constraint		

Attribute		Explanation
Name	nurse_number	
Data type	int	nurse_number is presented as an integer
Data length		
Constraint	NOT NULL	nurse_number is not permitted to be NULL
Name	room_number	
Data type	int	room_number is presented as an integer
Data length		
Constraint	NOT NULL	room_number is not permitted to be NULL
Name	floor_number	
Data type	int	floor_number is presented as an integer
Data length		
Constraint	NOT NULL	floor_number is not permitted to be NULL
Name	building_number	
Data type	int	building_number is presented as an integer
Data length		
Constraint	NOT NULL	building_number is not permitted to be NULL
Name	camp_number	
Data type	int	camp_number is presented as an integer
Data length		
Constraint	NOT NULL	camp_number is not permitted to be NULL

- **room**

Attribute		Explanation
Name	room_number	
Data type	int	room_number is presented as an integer
Data length		
Constraint	NOT NULL	room_number is not permitted to be NULL
Name	floor_number	
Data type	int	floor_number is presented as an integer
Data length		
Constraint	NOT NULL	floor_number is not permitted to be NULL
Name	building_number	
Data type	int	building_number is presented as an integer
Data length		
Constraint	NOT NULL	building_number is not permitted to be NULL
Name	camp_number	
Data type	int	camp_number is presented as an integer
Data length		
Constraint	NOT NULL	camp_number is not permitted to be NULL
Name	capacity	
Data type	int	capacity is presented as an integer
Data length		
Constraint		
Name	room_type	
Data type	varchar	room_type contains only letters
Data length	100	room_type is exactly 100 character-length
Constraint	DEFAULT 'normal'	room_type is automatically assigned "normal" if it is not inserted

- **symptom**

Attribute		Explanation
Name	patient_number	
Data type	int	patient_number is presented as an integer
Data length		
Constraint	NOT NULL	patient_number is not permitted to be NULL
Name	check_time	
Data type	date	check_time is in form YYYY-MM-DD
Data length		
Constraint	NOT NULL	check_time is not permitted to be NULL
Name	description	
Data type	varchar	description contains only letters
Data length	500	description is exactly 500 character-length
Constraint	NOT NULL	description is not permitted to be NULL

- **testing_information**

Attribute		Explanation
Name	patient_number	
Data type	int	patient_number is presented as an integer
Data length		
Constraint	NOT NULL	patient_number is not permitted to be NULL
Name	test_time	
Data type	date	test_time is in form YYYY-MM-DD
Data length		
Constraint	NOT NULL	test_time is not permitted to be NULL
Name	pcr_test_result	
Data type	char	pcr_test_result contains only a letter.
Data length	1	pcr_test_result is exactly 1 character-length
Constraint		
Name	pcr_test_ct_value	
Data type	decimal	pcr_test_ct_value is presented in form of digits
Data length	(4, 1)	pcr_test_ct_value is 4 - the scale is 1 (xxx.x form)
Constraint		
Name	quick_test_result	
Data type	char	quick_test_result contains only a letter.
Data length	1	quick_test_result is exactly 1 character-length
Constraint		
Name	quick_test_ct_value	
Data type	decimal	quick_test_ct_value is presented in form of digits
Data length	(4, 1)	quick_test_ct_value is 4 - the scale is 1 (xxx.x form)
Constraint		
Name	respiratory_rate	
Data type	int	respiratory_rate is presented as an integer
Data length		
Constraint		

Attribute		Explanation
Name	spo2	
Data type	decimal	spo2 is presented in form of digits
Data length	(4, 1)	spo2 is 4 - the scale is 1 (xxx.x form)
Constraint		

- **treatment**

Attribute		Explanation
Name	patient_number	
Data type	int	patient_number is presented as an integer
Data length		
Constraint	NOT NULL	patient_number is not permitted to be NULL
Name	doctor_number	
Data type	int	doctor_number is presented as an integer
Data length		
Constraint	NOT NULL	doctor_number is not permitted to be NULL
Name	medication_code	
Data type	varchar	medication_code contains both letters and numbers
Data length	6	medication_code is exactly 6 character-length
Constraint	NOT NULL	medication_code is not permitted to be NULL
Name	result	
Data type	varchar	result contains only letters
Data length	500	result is exactly 500 character-length
Constraint		
Name	start_date	
Data type	date	start_date is in form YYYY-MM-DD
Data length		
Constraint	NOT NULL	start_date is not permitted to be NULL
Name	end_date	
Data type	date	end_date is in form YYYY-MM-DD
Data length		
Constraint	NOT NULL	end_date is not permitted to be NULL

The table below shows our explanation for each key constraint which belongs to one of the following types of key constraints

– Composite primary keys such as pk_effect, pk_building, pk_comorbidity, pk_floor, pk_room, pk_symptom, pk_treatment

– Unique keys such as unq_effect_medication, unq_camp, unq_patient,

– Foreign keys such as fk_building_camp, fk_camp_doctor, fk_comorbidity, ...

- **Key Constraints**

Name	Type	Explaination
pk_effect	PRIMARY KEY	This is a combination of a UNIQUE (medication_code) and a NOT NULL (description) attribute
unq_effect _medication _code	UNIQUE	Used to identify attribute medication_code as UNIQUE to ensure that all values in this column are different
pk_building	PRIMARY KEY	This is a combination of a UNIQUE (camp_number) and a NOT NULL (building_number) attribute
unq_camp	UNIQUE	Used to identify attribute head_of_camp as UNIQUE to satisfy the requirement ”One doctor will be designated as the head of the camp”
pk_comorbidity	PRIMARY KEY	This is a combination of a UNIQUE (patient_number) and a NOT NULL (name) attribute
pk_floor	PRIMARY KEY	This is a combination of a UNIQUE (camp_number, building_number) and a NOT NULL (floor_number) attribute
unq_patient	UNIQUE	Used to identify attribute identity_number as UNIQUE to ensure that all values in this column are different
pk_room	PRIMARY KEY	This is a combination of a UNIQUE (camp_number, building_number, floor_number) and a NOT NULL (room_number) attribute
pk_symptom	PRIMARY KEY	This is a combination of a UNIQUE (patient_number) and a NOT NULL (check_time, description) attribute
pk_testing _information	UNIQUE	This combination of testing_results from these attributes is set UNIQUE to identify testing result of different patients
pk_treatment	PRIMARY KEY	This is a combination of a UNIQUE and NOT NULL (patient_number, doctor_number, medication_code) attribute
fk_building _camp	FOREIGN KEY	It indicates the link between table building and table camp through attribute camp_number

Name	Type	Explaination
fk_camp_doctor	FOREIGN KEY	It indicates the link between table camp and table doctor by mentioning that attribute head_of_camp of table camp infered from attribute personel_number of table doctor
fk_comorbidity_patient	FOREIGN KEY	It indicates the link between table comorbidity and table patient by mentioning that attribute patient_number of table comorbidity infered from attribute patient_number of table patient
fk_doctor_camp	FOREIGN KEY	It indicates the link between table doctor and table camp by mentioning that attribute camp_number of table doctor infered from attribute camp_number of table camp
fk_effect_medication	FOREIGN KEY	It indicates the link between table effect and table medication by mentioning that attribute medication_code of table effect infered from attribute code of table medication
fk_floor_building	FOREIGN KEY	It indicates the link between table floor and table building by mentioning that attribute building_number, camp_number of table floor infered from attribute building_number, camp_number of table building
fk_manager_camp	FOREIGN KEY	It indicates the link between table manager and table camp by mentioning that attribute camp_number of table manager infered from attribute camp_number of table camp
fk_nurse_camp	FOREIGN KEY	It indicates the link between table nurse and table camp by mentioning that attribute camp_number of table nurse infered from attribute camp_number of table camp
fk_patient_nurse	FOREIGN KEY	It indicates the link between table patient and table nurse by mentioning that attribute nurse_number of table patient infered from attribute personnel_number of table nurse
fk_patient_room	FOREIGN KEY	It indicates the link between table patient and table room by mentioning that attribute room_number, floor_number, building_number, camp_number of table patient infered from attribute room_number, floor_number, building_number, camp_number of table room
fk_patient_staff	FOREIGN KEY	It indicates the link between table patient and table staff by mentioning that attribute admitting_staff of table patient infered from attribute personnel_number of table staff
fk_room_floor	FOREIGN KEY	It indicates the link between table room and table floor by mentioning that attribute floor_number, building_number, camp_number of table room infered from attribute floor_number, building_number, camp_number of table floor
fk_staff_camp	FOREIGN KEY	It indicates the link between table staff and table camp by mentioning that attribute camp_number of table staff infered from attribute camp_number of table camp

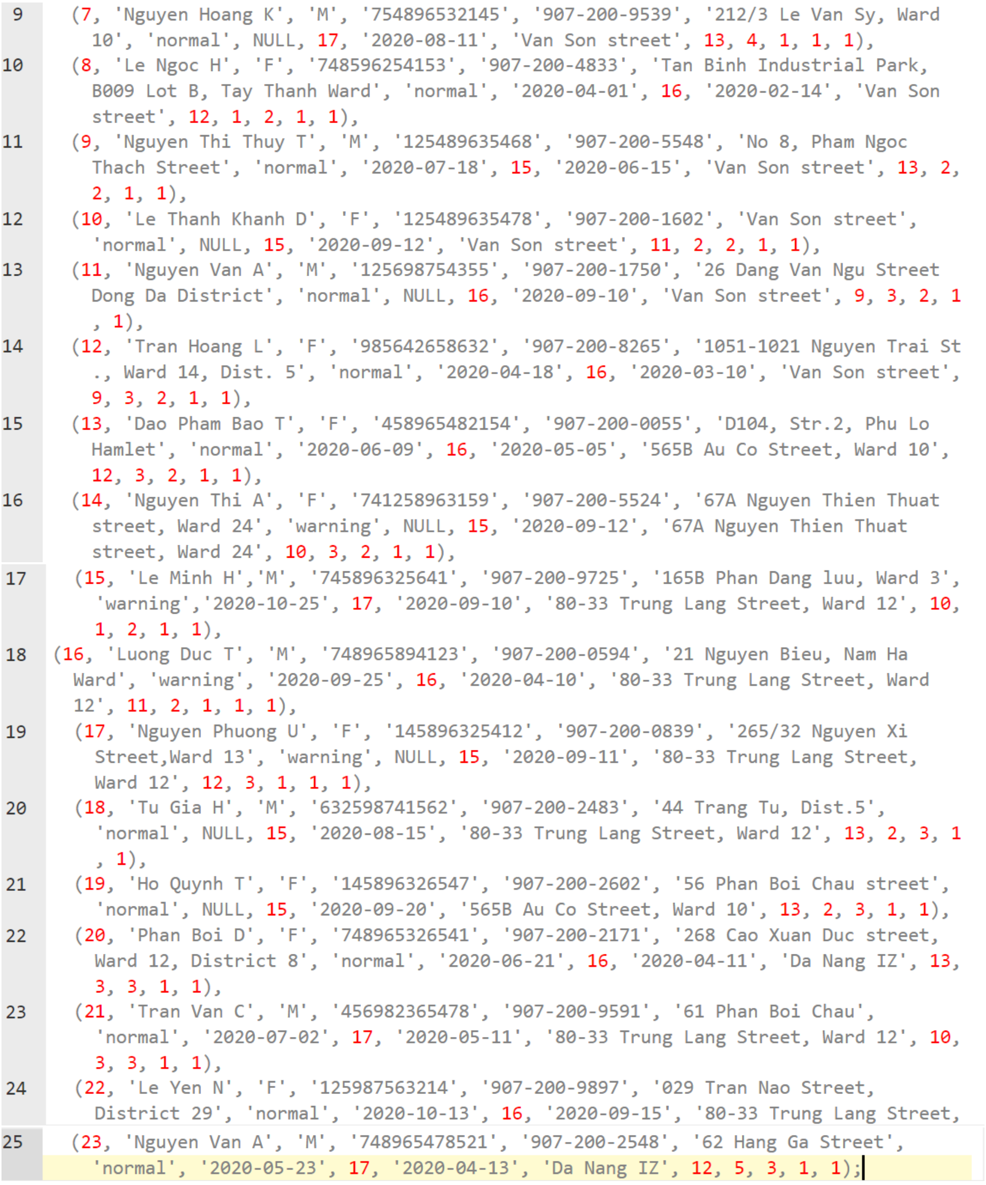
Name	Type	Explaination
fk_symptom_patient	FOREIGN KEY	It indicates the link between table symptom and table patient by mentioning that attribute patient_number of table symptom infered from attribute patient_number of table patient
fk_testing_information_patient	FOREIGN KEY	It indicates the link between table testing_information and table patient by mentioning that attribute patient_number of table testing_information infered from attribute patient_number of table patient
fk_treatment_doctor	FOREIGN KEY	It indicates the link between table treatment and table doctor by mentioning that attribute doctor_number of table treatment infered from attribute personnel_number of table doctor
fk_treatment_medication	FOREIGN KEY	It indicates the link between table treatment and table medication by mentioning that attribute medication_code of table treatment infered from attribute code of table medication
fk_treatment_patient	FOREIGN KEY	It indicates the link between table treatment and table patient by mentioning that attribute patient_number of table treatment infered from attribute patient_number of table patient
fk_volunteer_camp	FOREIGN KEY	It indicates the link between table volunteer and table camp by mentioning that attribute camp_number of table volunteer infered from attribute camp_number of table camp

1.2 Inserting data

- Insert data for all tables in the database.
- Requirements: The data in the tables must be meaningful, and each table has at least 4 rows.

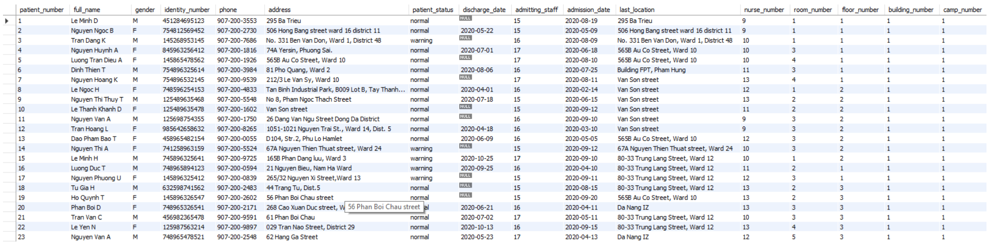
The code:

```
INSERT INTO patient ( patient_number, full_name, gender, identity_number, phone, address, patient_status, discharge_date, admitting_staff, admission_date, last_location, nurse_number, room_number, floor_number, building_number, camp_number )
VALUES
  (1, 'Le Minh D', 'M', '451284695123', '907-200-3553', '295 Ba Trieu', 'normal', NULL, 15, '2020-08-19', '295 Ba Trieu', 9, 1, 1, 1, 1),
  (2, 'Nguyen Ngoc B', 'F', '754812569452', '907-200-2730', '506 Hong Bang street ward 16 district 11', 'normal', '2020-05-22', 15, '2020-05-09', '506 Hong Bang street ward 16 district 11', 9, 1, 1, 1, 1),
  (3, 'Tran Dang K', 'M', '145268953145', '907-200-7686', 'No. 331 Ben Van Don, Ward 1, District 48', 'warning', NULL, 16, '2020-08-09', 'No. 331 Ben Van Don, Ward 1, District 48', 10, 1, 1, 1, 1),
  (4, 'Nguyen Huynh A', 'F', '845963256412', '907-200-1816', '74A Yersin, Phuong Sai.', 'normal', '2020-07-01', 17, '2020-06-18', '565B Au Co Street, Ward 10', 10, 3, 1, 1, 1),
  (5, 'Luong Tran Dieu A', 'F', '145865478562', '907-200-1926', '565B Au Co Street, Ward 10', 'normal', NULL, 17, '2020-08-10', '565B Au Co Street, Ward 10', 10, 4, 1, 1, 1),
  (6, 'Dinh Thien T', 'M', '754896325614', '907-200-3984', '81 Pho Quang, Ward 2', 'normal', '2020-08-06', 16, '2020-07-25', 'Building FPT, Pham Hung', 11, 3, 1, 1, 1),
```



```
(7, 'Nguyen Hoang K', 'M', '754896532145', '907-200-9539', '212/3 Le Van Sy, Ward
  10', 'normal', NULL, 17, '2020-08-11', 'Van Son street', 13, 4, 1, 1, 1),
(8, 'Le Ngoc H', 'F', '748596254153', '907-200-4833', 'Tan Binh Industrial Park,
  B009 Lot B, Tay Thanh Ward', 'normal', '2020-04-01', 16, '2020-02-14', 'Van Son
  street', 12, 1, 2, 1, 1),
(9, 'Nguyen Thi Thuy T', 'M', '125489635468', '907-200-5548', 'No 8, Pham Ngoc
  Thach Street', 'normal', '2020-07-18', 15, '2020-06-15', 'Van Son street', 13, 2,
  2, 1, 1),
(10, 'Le Thanh Khanh D', 'F', '125489635478', '907-200-1602', 'Van Son street',
  'normal', NULL, 15, '2020-09-12', 'Van Son street', 11, 2, 2, 1, 1),
(11, 'Nguyen Van A', 'M', '125698754355', '907-200-1750', '26 Dang Van Ngu Street
  Dong Da District', 'normal', NULL, 16, '2020-09-10', 'Van Son street', 9, 3, 2, 1
  , 1),
(12, 'Tran Hoang L', 'F', '985642658632', '907-200-8265', '1051-1021 Nguyen Trai St
  ., Ward 14, Dist. 5', 'normal', '2020-04-18', 16, '2020-03-10', 'Van Son street',
  9, 3, 2, 1, 1),
(13, 'Dao Pham Bao T', 'F', '458965482154', '907-200-0055', 'D104, Str.2, Phu Lo
  Hamlet', 'normal', '2020-06-09', 16, '2020-05-05', '565B Au Co Street, Ward 10',
  12, 3, 2, 1, 1),
(14, 'Nguyen Thi A', 'F', '741258963159', '907-200-5524', '67A Nguyen Thien Thuat
  street, Ward 24', 'warning', NULL, 15, '2020-09-12', '67A Nguyen Thien Thuat
  street, Ward 24', 10, 3, 2, 1, 1),
 (15, 'Le Minh H','M', '745896325641', '907-200-9725', '165B Phan Dang luu, Ward 3',
  'warning','2020-10-25', 17, '2020-09-10', '80-33 Trung Lang Street, Ward 12', 10,
  1, 2, 1, 1),
(16, 'Luong Duc T', 'M', '748965894123', '907-200-0594', '21 Nguyen Bieu, Nam Ha
Ward', 'warning', '2020-09-25', 16, '2020-04-10', '80-33 Trung Lang Street, Ward
12', 11, 2, 1, 1, 1),
 (17, 'Nguyen Phuong U', 'F', '145896325412', '907-200-0839', '265/32 Nguyen Xi
  Street,Ward 13', 'warning', NULL, 15, '2020-09-11', '80-33 Trung Lang Street,
  Ward 12', 12, 3, 1, 1, 1),
 (18, 'Tu Gia H', 'M', '632598741562', '907-200-2483', '44 Trang Tu, Dist.5',
  'normal', NULL, 15, '2020-08-15', '80-33 Trung Lang Street, Ward 12', 13, 2, 3, 1
  , 1),
 (19, 'Ho Quynh T', 'F', '145896326547', '907-200-2602', '56 Phan Boi Chau street',
  'normal', NULL, 15, '2020-09-20', '565B Au Co Street, Ward 10', 13, 2, 3, 1, 1),
 (20, 'Phan Boi D', 'F', '748965326541', '907-200-2171', '268 Cao Xuan Duc street,
  Ward 12, District 8', 'normal', '2020-06-21', 16, '2020-04-11', 'Da Nang IZ', 13,
  3, 3, 1, 1),
 (21, 'Tran Van C', 'M', '456982365478', '907-200-9591', '61 Phan Boi Chau',
  'normal', '2020-07-02', 17, '2020-05-11', '80-33 Trung Lang Street, Ward 12', 10,
  3, 3, 1, 1),
 (22, 'Le Yen N', 'F', '125987563214', '907-200-9897', '029 Tran Nao Street,
  District 29', 'normal', '2020-10-13', 16, '2020-09-15', '80-33 Trung Lang Street,
(23, 'Nguyen Van A', 'M', '748965478521', '907-200-2548', '62 Hang Ga Street',
  'normal', '2020-05-23', 17, '2020-04-13', 'Da Nang IZ', 12, 5, 3, 1, 1);
```

The result:

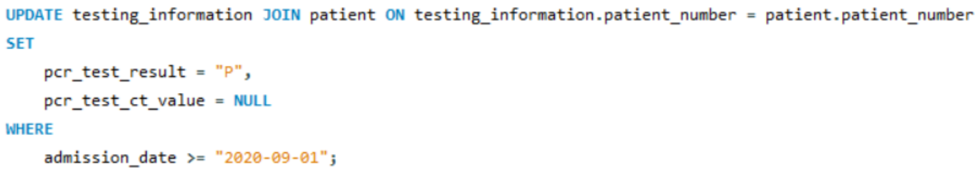


	patient_number	full_name	gender	identity_number	phone	address	patient_status	discharge_date	admitting_staff	admission_date	last_location	nurse_number	room_number	floor_number	building_number	camp_number
▸	1	Le Minh D	M	451284695123	907-200-3553	295 Ba Trieu	normal	NULL	15	2020-08-19	295 Ba Trieu	9	1	1	1	1
	2	Nguyen Ngoc B	F	754812569452	907-200-2730	506 Hong Bang street ward 16 district 11	normal	2020-05-22	15	2020-05-09	506 Hong Bang street ward 16 district 11	9	1	1	1	1
	3	Tran Dang K	M	145268953145	907-200-7686	No. 331 Ben Van Don, Ward 1, District 48	warning	NULL	16	2020-08-09	No. 331 Ben Van Don, Ward 1, District 48	10	1	1	1	1
	4	Nguyen Huynh A	F	845963256412	907-200-1816	74A Yersin, Phuong Sai.	normal	2020-07-01	17	2020-06-18	565B Au Co Street, Ward 10	10	3	1	1	1
	5	Luong Tran Dieu A	F	145865478562	907-200-1926	565B Au Co Street, Ward 10	normal	NULL	17	2020-08-10	565B Au Co Street, Ward 10	10	4	1	1	1
	6	Dinh Thien T	M	754896325614	907-200-3984	81 Pho Quang, Ward 2	normal	2020-08-06	16	2020-07-25	Building FPT, Pham Hung	11	3	1	1	1
	7	Nguyen Hoang K	M	754896532145	907-200-9539	212/3 Le Van Sy, Ward 10	normal	NULL	17	2020-08-11	Van Son street	13	4	1	1	1
	8	Le Ngoc H	F	748596254153	907-200-4833	Tan Binh Industrial Park, B009 Lot B, Tay Thanh...	normal	2020-04-01	16	2020-02-14	Van Son street	12	1	2	1	1
	9	Nguyen Thi Thuy T	M	125489635468	907-200-5548	No 8, Pham Ngoc Thach Street	normal	2020-07-18	15	2020-06-15	Van Son street	13	2	2	1	1
	10	Le Thanh Khanh D	F	125489635478	907-200-1602	Van Son street	normal	NULL	15	2020-09-12	Van Son street	11	2	2	1	1
	11	Nguyen Van A	M	125698754355	907-200-1750	26 Dang Van Ngu Street Dong Da District	normal	NULL	16	2020-09-10	Van Son street	9	3	2	1	1
	12	Tran Hoang L	F	985642658632	907-200-8265	1051-1021 Nguyen Trai St., Ward 14, Dist. 5	normal	2020-04-18	16	2020-03-10	Van Son street	9	3	2	1	1
	13	Dao Pham Bao T	F	458965482154	907-200-0055	D104, Str.2, Phu Lo Hamlet	normal	2020-06-09	16	2020-05-05	565B Au Co Street, Ward 10	12	3	2	1	1
	14	Nguyen Thi A	F	741258963159	907-200-5524	67A Nguyen Thien Thuat street, Ward 24	warning	NULL	15	2020-09-12	67A Nguyen Thien Thuat street, Ward 24	10	3	2	1	1
	15	Le Minh H	M	745896325641	907-200-9725	165B Phan Dang luu, Ward 3	warning	2020-10-25	17	2020-09-10	80-33 Trung Lang Street, Ward 12	10	1	2	1	1
	16	Luong Duc T	M	748965894123	907-200-0594	21 Nguyen Bieu, Nam Ha Ward	warning	2020-09-25	16	2020-04-10	80-33 Trung Lang Street, Ward 12	11	2	1	1	1
	17	Nguyen Phuong U	F	145896325412	907-200-0839	265/32 Nguyen Xi Street,Ward 13	warning	NULL	15	2020-09-11	80-33 Trung Lang Street, Ward 12	12	3	1	1	1
	18	Tu Gia H	M	632598741562	907-200-2483	44 Trang Tu, Dist.5	normal	NULL	15	2020-08-15	80-33 Trung Lang Street, Ward 12	13	2	3	1	1
	19	Ho Quynh T	F	145896326547	907-200-2602	56 Phan Boi Chau street	normal	NULL	15	2020-09-20	565B Au Co Street, Ward 10	13	2	3	1	1
	20	Phan Boi D	F	748965326541	907-200-2171	268 Cao Xuan Duc street, W 56 Phan Boi Chau street	al	2020-06-21	16	2020-04-11	Da Nang IZ	13	3	3	1	1
	21	Tran Van C	M	456982365478	907-200-9591	61 Phan Boi Chau	normal	2020-07-02	17	2020-05-11	80-33 Trung Lang Street, Ward 12	10	3	3	1	1
	22	Le Yen N	F	125987563214	907-200-9897	029 Tran Nao Street, District 29	normal	2020-10-13	16	2020-09-15	80-33 Trung Lang Street, Ward 12	13	4	3	1	1
	23	Nguyen Van A	M	748965478521	907-200-2548	62 Hang Ga Street	normal	2020-05-23	17	2020-04-13	Da Nang IZ	12	5	3	1	1

2 Store Procedure / Function / SQL

a) **Update patient PCR test to positive with null cycle threshold value for all patients whose admission date is from 01/09/2020.**

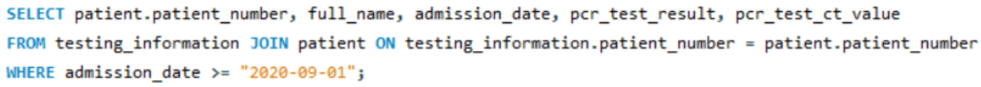
We used UPDATE statement to modify existing records on the table. Since the admission date is in the patient table and the PCR test results are in the testing information table, we JOINED these 2 tables together first based on patient number.
After that, we SET PCR test result to positive with null cycle threshold value WHERE admission date is from 2020-09-01.



```
UPDATE testing_information JOIN patient ON testing_information.patient_number = patient.patient_number
SET
    pcr_test_result = "P",
    pcr_test_ct_value = NULL
WHERE
    admission_date >= "2020-09-01";
```

Figure 1: UPDATE statement for Patient PCR Test

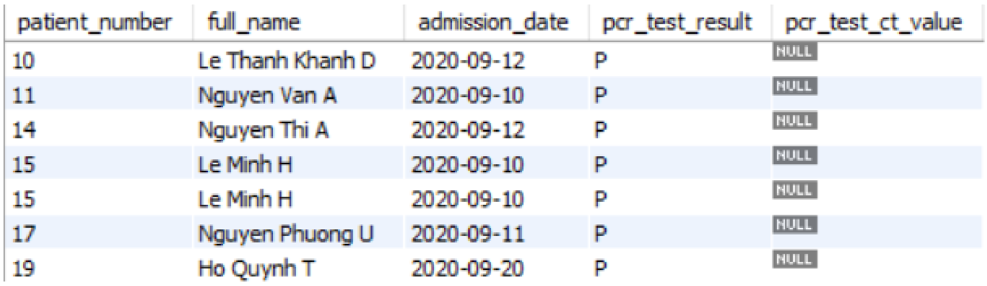
We used SELECT patient JOIN with testing information to see the result.



```
SELECT patient.patient_number, full_name, admission_date, pcr_test_result, pcr_test_ct_value
FROM testing_information JOIN patient ON testing_information.patient_number = patient.patient_number
WHERE admission_date >= "2020-09-01";
```

Figure 2: SELECT statement for results

And all the rows returned had *pcr_test_result* $= P$ and *pcr_test_ct_value* $= NULL$.

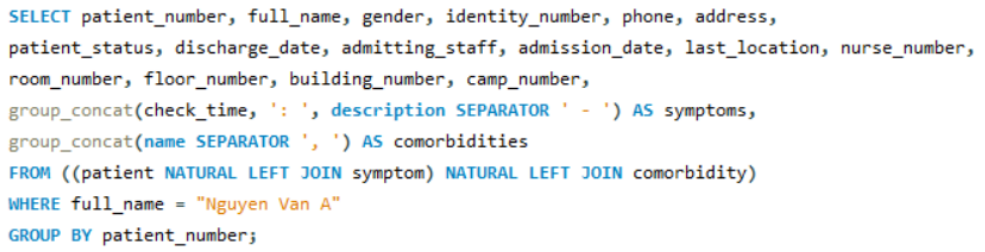


patient_number	full_name	admission_date	pcr_test_result	pcr_test_ct_value
10	Le Thanh Khanh D	2020-09-12	P	NULL
11	Nguyen Van A	2020-09-10	P	NULL
14	Nguyen Thi A	2020-09-12	P	NULL
15	Le Minh H	2020-09-10	P	NULL
15	Le Minh H	2020-09-10	P	NULL
17	Nguyen Phuong U	2020-09-11	P	NULL
19	Ho Quynh T	2020-09-20	P	NULL

Figure 3: Results for SELECT statement

b) **Select all the patient information whose name is ‘Nguyen Van A’.**

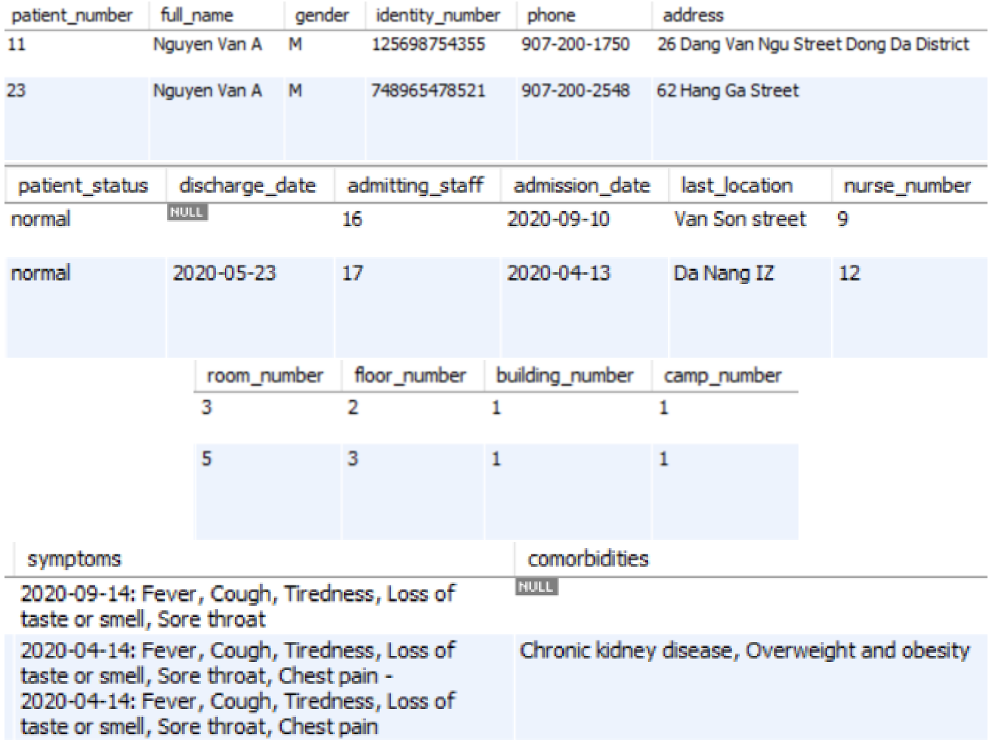
We decided that patient information would be the patient’s demographic information, comorbidities and symptoms.
Because of that, we JOINED 3 tables: patient, symptom and comorbidity together. Since symptoms and comorbidities had multiple rows for 1 patient, we used GROUP_CONCAT for these two to make sure that 1 patient only has 1 row with GROUP BY patient_number. The condition of SELECT is full_name = “Nguyen Van A”.



```
SELECT patient_number, full_name, gender, identity_number, phone, address,
patient_status, discharge_date, admitting_staff, admission_date, last_location, nurse_number,
room_number, floor_number, building_number, camp_number,
group_concat(check_time, ': ', description SEPARATOR ' - ') AS symptoms,
group_concat(name SEPARATOR ', ') AS comorbidities
FROM ((patient NATURAL LEFT JOIN symptom) NATURAL LEFT JOIN comorbidity)
WHERE full_name = "Nguyen Van A"
GROUP BY patient_number;
```

Figure 4: SELECT statement for All Patient Information

The results returned:



patient_number	full_name	gender	identity_number	phone	address
11	Nguyen Van A	M	125698754355	907-200-1750	26 Dang Van Ngu Street Dong Da District
23	Nguyen Van A	M	748965478521	907-200-2548	62 Hang Ga Street

patient_status	discharge_date	admitting_staff	admission_date	last_location	nurse_number
normal	NULL	16	2020-09-10	Van Son street	9
normal	2020-05-23	17	2020-04-13	Da Nang IZ	12

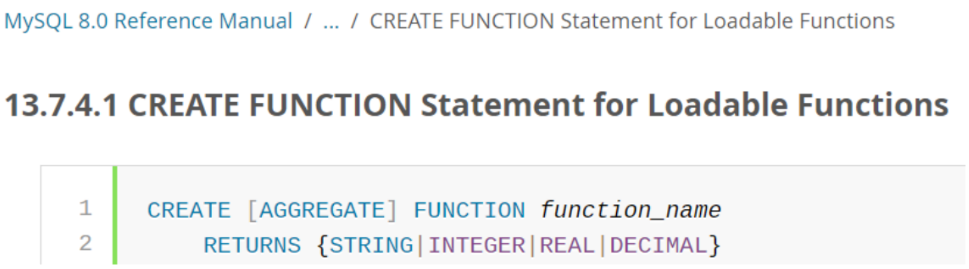
room_number	floor_number	building_number	camp_number
3	2	1	1
5	3	1	1

symptoms	comorbidities
2020-09-14: Fever, Cough, Tiredness, Loss of taste or smell, Sore throat	NULL
2020-04-14: Fever, Cough, Tiredness, Loss of taste or smell, Sore throat, Chest pain - 2020-04-14: Fever, Cough, Tiredness, Loss of taste or smell, Sore throat, Chest pain	Chronic kidney disease, Overweight and obesity

Figure 5: Results for the SELECT Statement

**c) Write a function to calculate the testing for each patient.
Input: Patient ID
Output: A list of testing**

Because MySQL doesn't allow us to return a list, we used a stored procedure for this instead.



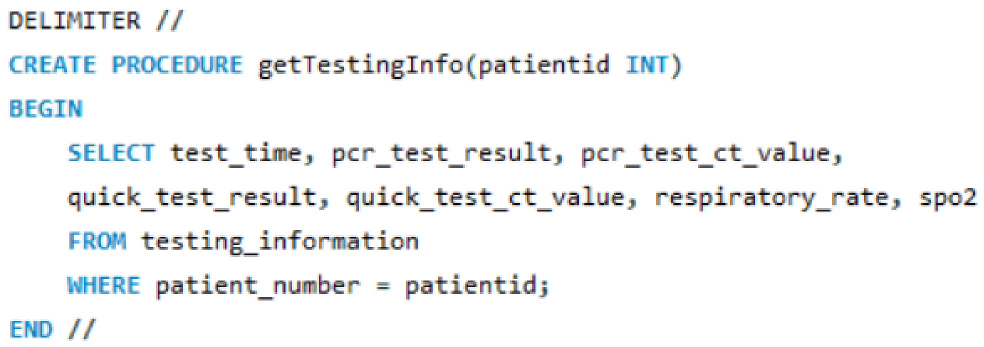
MySQL 8.0 Reference Manual / ... / CREATE FUNCTION Statement for Loadable Functions

13.7.4.1 CREATE FUNCTION Statement for Loadable Functions

```
CREATE [AGGREGATE] FUNCTION function_name
    RETURNS {STRING|INTEGER|REAL|DECIMAL}
```

Figure 6: MySQL CREATE FUNCTION statement

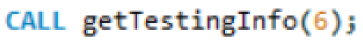
Since the input is the patient ID which is an integer, the parameter would be patientid INT. We needed to return a list of testing so we SELECTED it from testing_information table WHERE patient_number of that table equaled the input.



```
DELIMITER //
CREATE PROCEDURE getTestingInfo(patientid INT)
BEGIN
    SELECT test_time, pcr_test_result, pcr_test_ct_value,
    quick_test_result, quick_test_ct_value, respiratory_rate, spo2
    FROM testing_information
    WHERE patient_number = patientid;
END //
```

Figure 7: Testing List Procedure

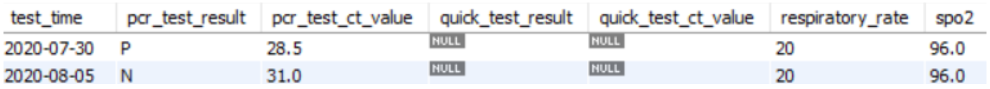
We called the procedure with example input:



```
CALL getTestingInfo(6);
```

Figure 8: Calling Testing List Procedure Example

Results returned:



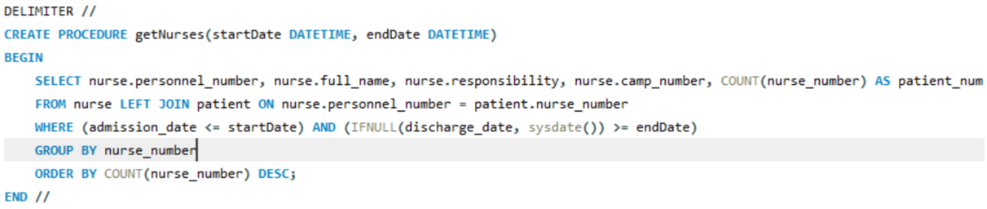
test_time	pcr_test_result	pcr_test_ct_value	quick_test_result	quick_test_ct_value	respiratory_rate	spo2
2020-07-30	P	28.5	NULL	NULL	20	96.0
2020-08-05	N	31.0	NULL	NULL	20	96.0

Figure 9: Results of the Procedure

d) **Write a procedure to sort the nurses in decreasing number of patients he/she takes care in a period of time.**
Input: Start date, End date
Output: A list of sorting nurses.

The stored procedure had 2 inputs: startDate DATETIME and endDate DATETIME. Because nurse information is in the "nurse" table and the date is in the "patient" table, we JOINED these two tables together based on the nurse ID.
The columns we selected were: all columns from the "nurse" table and the COUNTING for

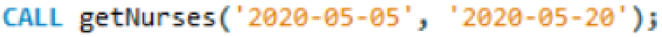
the same nurse from different patients. Each row of the list was a nurse so we GROUPED BY nurse_number and ORDERED BY DECREASING number of the COUNTING.



```
DELIMITER //
CREATE PROCEDURE getNurses(startDate DATETIME, endDate DATETIME)
BEGIN
    SELECT nurse.personnel_number, nurse.full_name, nurse.responsibility, nurse.camp_number, COUNT(nurse_number) AS patient_num
    FROM nurse LEFT JOIN patient ON nurse.personnel_number = patient.nurse_number
    WHERE (admission_date <= startDate) AND (IFNULL(discharge_date, sysdate()) >= endDate)
    GROUP BY nurse_number
    ORDER BY COUNT(nurse_number) DESC;
END //
```

Figure 10: Nurse List Procedure

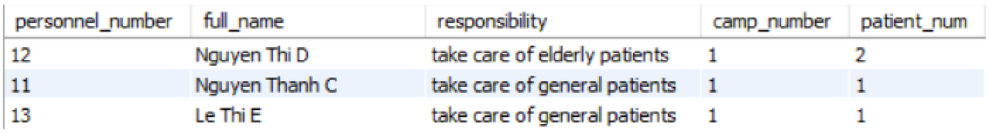
We called the procedure:



```
CALL getNurses('2020-05-05', '2020-05-20');
```

Figure 11: Calling Nurse List Procedure Example

Results returned:



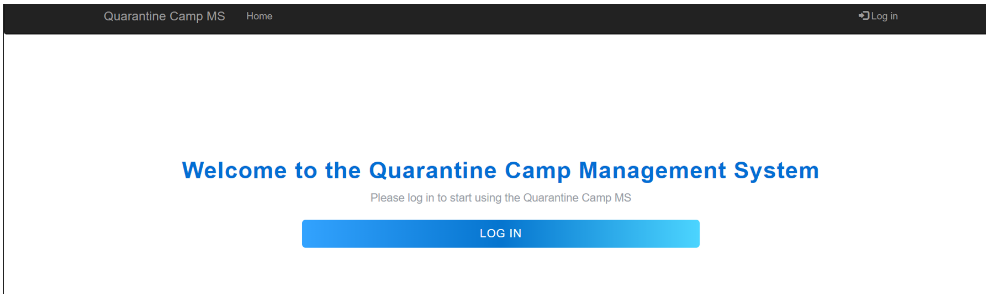
personnel_number	full_name	responsibility	camp_number	patient_num
12	Nguyen Thi D	take care of elderly patients	1	2
11	Nguyen Thanh C	take care of general patients	1	1
13	Le Thi E	take care of general patients	1	1

Figure 12: Results of the Procedure

3 Building Applications

3.1 Prerequisites

In order to run the application, user needs to install a HTTP server with PHP and a MySQL server on their local machine. It is necessary that the MySQLi extension is enabled in PHP.
First, user need to rename the folder containing the web application's source code to ***web.db*** and place the folder in the document root of their HTTP server.
Now, user can access to this website on local host via this address: http://localhost/web.db/. After successfully accessing, they will see the interface as the following picture:

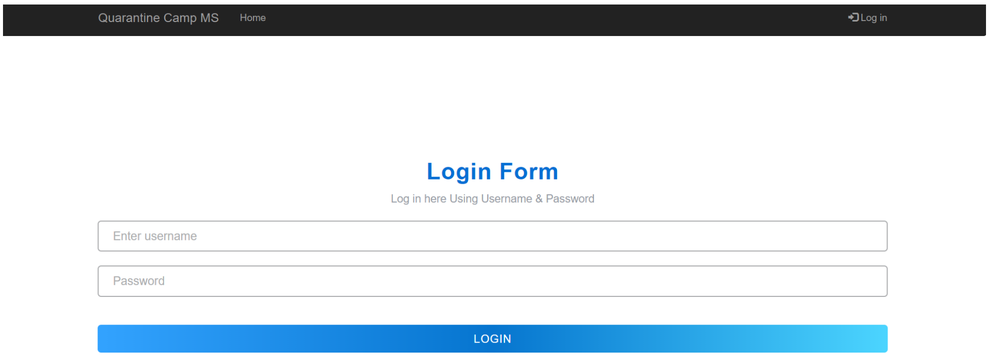


Welcome to the Quarantine Camp Management System

Please log in to start using the Quarantine Camp MS

Figure 13: Interface when first access to the website

Clicking the **LOG IN** button will lead the user to login page. User needs to log in as Manager to use functions of this web application.



Log in here Using Username & Password

Figure 14: Login Form Interface

3.2 Create user

Log in to the database with DBA privileges such as SYS / SYSTEM ...., create a user named "Manager" and assign all access rights to this user.

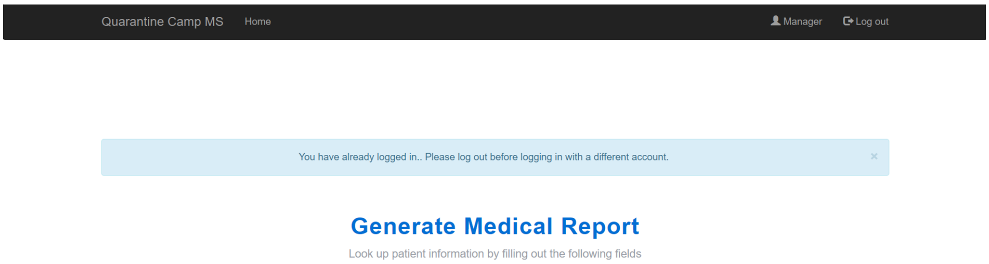
– First, we connect to our MySQL database of which name is *quarantine_camp* using **phpMyAdmin** or any MySQL client of choice.

– We create a user named "Manager" with password "manager" by this query:
`CREATE USER 'Manager'@'localhost' IDENTIFIED BY 'manager';`

– Assign all access rights to user "Manager" by:
`GRANT ALL ON quarantine_camp.* TO 'Manager'@'localhost' IDENTIFIED BY 'manager';`

3.3 Requirement function

- **Log in, log out (enter the user name/password for Manager account to log in-/out).**
  Now, it is possible to log in by username "Manager" with password "manager".



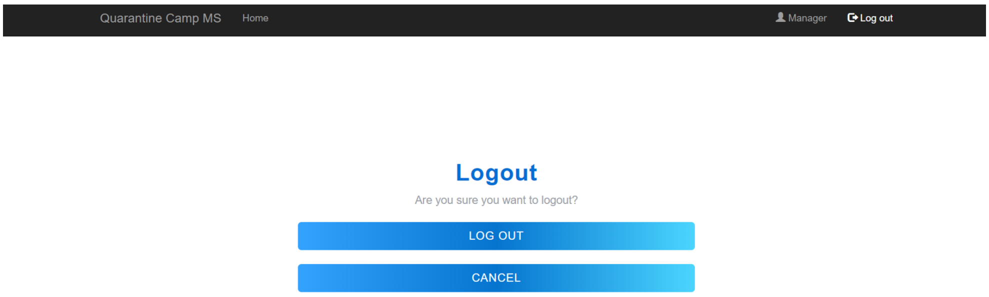
You have already logged in.. Please log out before logging in with a different account.

Generate Medical Report

Look up patient information by filling out the following fields

Figure 15: Interface after logging in as Manager

User can click the **Log out** button on navigation bar to log out.



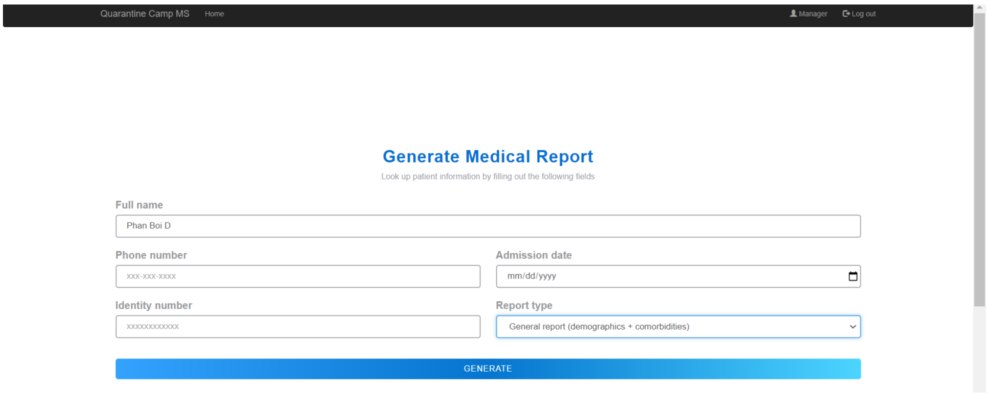
Logout

Are you sure you want to logout?

Figure 16: Logout Interface

- **Log in to the user manager and do the following:**

  1. **Search patient information: Search results include the name, phone number and information about his/her comorbidities.**
     User just needs to fill at least one fields of patient's information, and selects *General report (demographic + comorbidities)*.



Generate Medical Report

Look up patient information by filling out the following fields

Full name: Phan Boi D

Phone number: xxx-xxx-xxxx

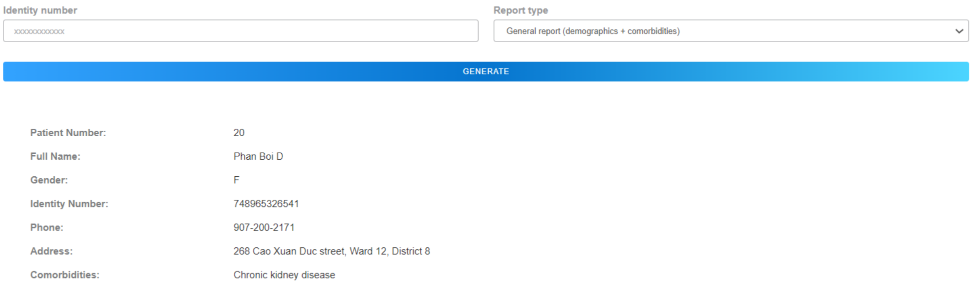
Admission date: mm/dd/yyyy

Identity number: xxxxxxxxxxxx

Report type: General report (demographics + comorbidities)

Figure 17: Information form for generating General report

The following picture is the result that is printed out.



Identity number: xxxxxxxxxxxx

Report type: General report (demographics + comorbidities)

Patient Number:	20
Full Name:	Phan Boi D
Gender:	F
Identity Number:	748965326541
Phone:	907-200-2171
Address:	268 Cao Xuan Duc street, Ward 12, District 8
Comorbidities:	Chronic kidney disease

Figure 18: Result printed out after generating General report

2. **Add information for a new patient.**
If user wants to add information for a new patient, just click the **Insert Patient Record** button and fill all the required fields in the form as the picture below:

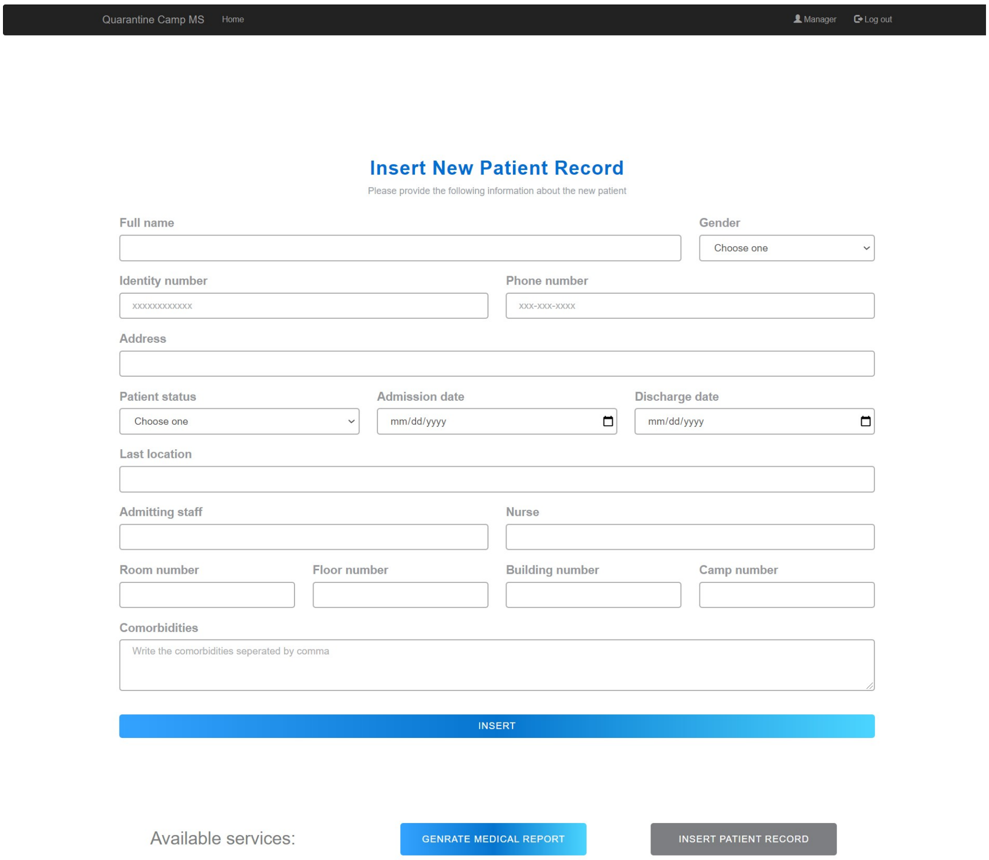
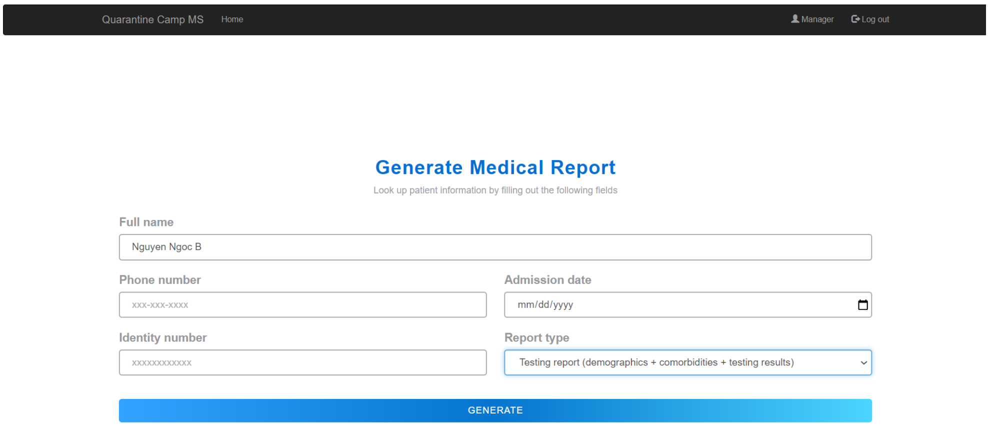


Figure 19: Information form for adding new patient record

3. **List details of all testing which belong to a patient.**
User just needs to fill at least one fields of patient's information, and selects *Testing report (demographic + comorbidities + testing results)*.



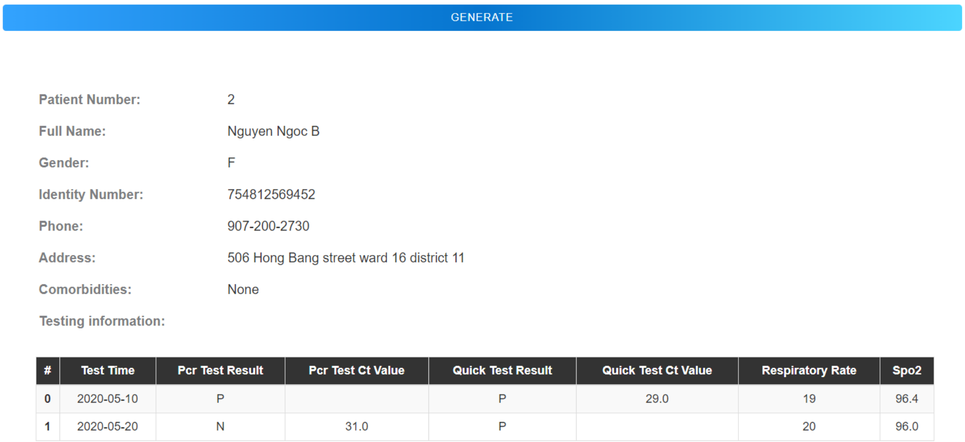
Generate Medical Report

Look up patient information by filling out the following fields

Full name	Nguyen Ngoc B
Phone number	xxx-xxx-xxxx
Admission date	mm/dd/yyyy
Identity number	xxxxxxxxxxxx
Report type	Testing report (demographics + comorbidities + testing results)

Figure 20: Information form for generating Testing report

The following picture is the result that is printed out. The details of all testings are listed in the table.

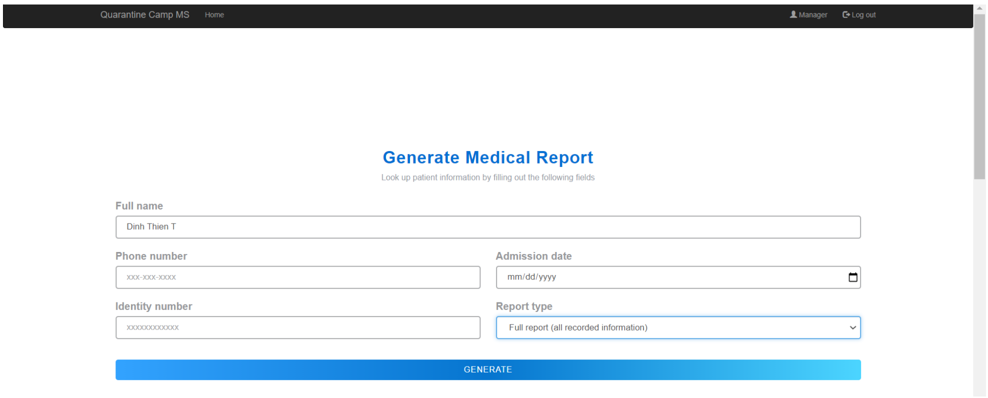


Patient Number:	2
Full Name:	Nguyen Ngoc B
Gender:	F
Identity Number:	754812569452
Phone:	907-200-2730
Address:	506 Hong Bang street ward 16 district 11
Comorbidities:	None
Testing information:	

#	Test Time	Pcr Test Result	Pcr Test Ct Value	Quick Test Result	Quick Test Ct Value	Respiratory Rate	Spo2
0	2020-05-10	P		P	29.0	19	96.4
1	2020-05-20	N	31.0	P		20	96.0

Figure 21: Result printed out after generating Testing report

4. **Make a report that provides full information about the patient including demographic information, comorbidities, symptoms, testing, and treatment.** User just needs to fill at least one fields of patient's information, and selects *Full report (all record information)*.



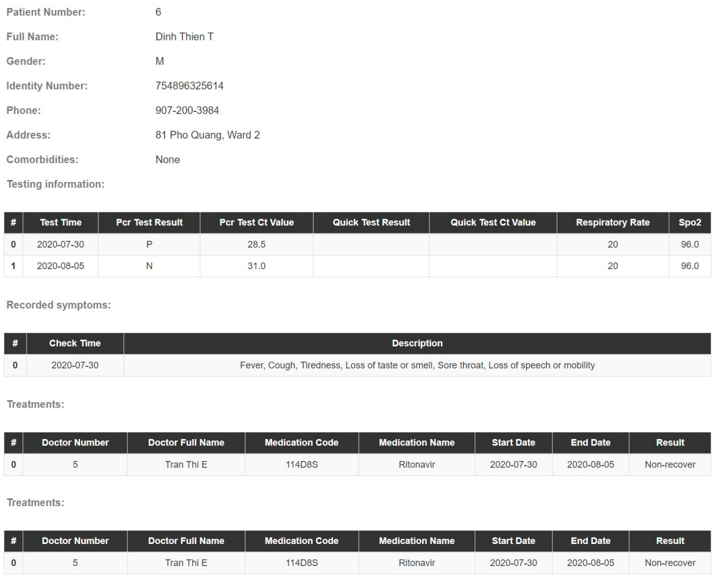
Generate Medical Report

Look up patient information by filling out the following fields

Full name	Dinh Thien T
Phone number	xxx-xxx-xxxx
Admission date	mm/dd/yyyy
Identity number	xxxxxxxxxxxx
Report type	Full report (all recorded information)

Figure 22: Information form for generating Full report

Full information about the patient including demographic information, comorbidities, symptoms, testing, and treatment. are printed out.



Patient Number:	6
Full Name:	Dinh Thien T
Gender:	M
Identity Number:	754896325614
Phone:	907-200-3984
Address:	81 Pho Quang, Ward 2
Comorbidities:	None

Testing information:

#	Test Time	Pcr Test Result	Pcr Test Ct Value	Quick Test Result	Quick Test Ct Value	Respiratory Rate	Spo2
0	2020-07-30	P	28.5			20	96.0
1	2020-08-05	N	31.0			20	96.0

Recorded symptoms:

#	Check Time	Description
0	2020-07-30	Fever, Cough, Tiredness, Loss of taste or smell, Sore throat, Loss of speech or mobility

Treatments:

#	Doctor Number	Doctor Full Name	Medication Code	Medication Name	Start Date	End Date	Result
0	5	Tran Thi E	114D8S	Ritonavir	2020-07-30	2020-08-05	Non-recover

Treatments:

#	Doctor Number	Doctor Full Name	Medication Code	Medication Name	Start Date	End Date	Result
0	5	Tran Thi E	114D8S	Ritonavir	2020-07-30	2020-08-05	Non-recover

Figure 23: Result printed out after generating Full report