

Group 2 Project Report

Tindoc

Member

Pasit Tiwawongrut 122442
Saratoon Khantasima 122149
Noppawan Nurnuansuwan 122410
Vineela Mukkamala 122050
Harold Popluhar 122556

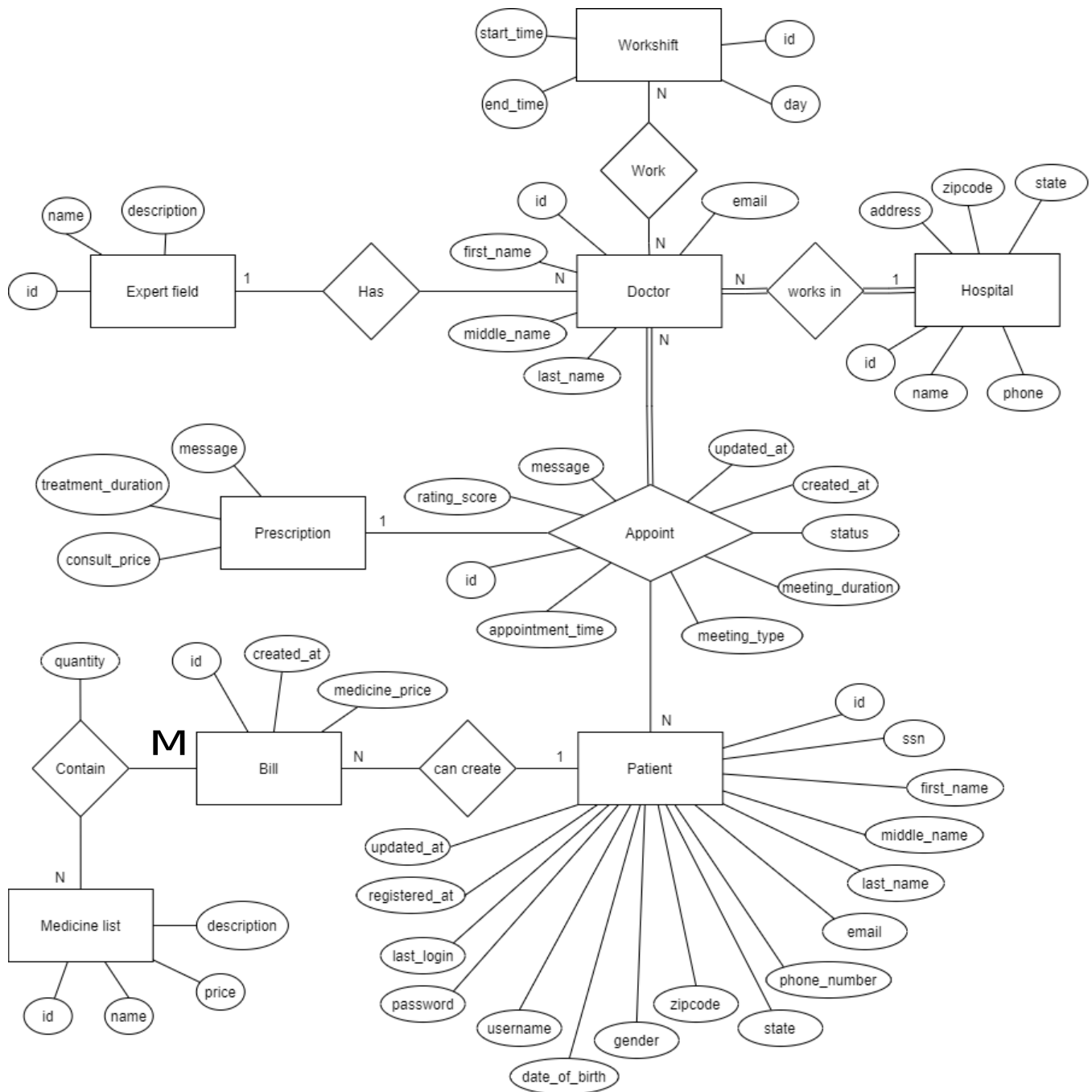
Table of contents

1) Project Description	1
2) Conceptual Design	2
3) Relational database schema diagram	3
4) Logical diagram	4
5) Use Case Diagram	5
6) Data dictionary	6
Patient	6
Hospital	6
Doctor	7
Appointment	7
Medicine	8
Bill	8
Expert field	8
Work	8
Bill item	9
Workshift	9
Prescription	9
7) Database Population and SQL data commands	9
10) Work Distribution and Contribution	12
11) Important Queries and Reports	15
Done by 122556 Harold Popluhar	15
Done by 122149 Saratoon Khantasima	21
Done by 122410 Nopphawan Nurnuansuwan	28
Done by 122442 Pasit Tiwawongrut	32
Done by 122050 Vineela Mukkamala	33
12) Create Table Queries	41
13) Conclusion	45
14) Work distribution and contribution	47

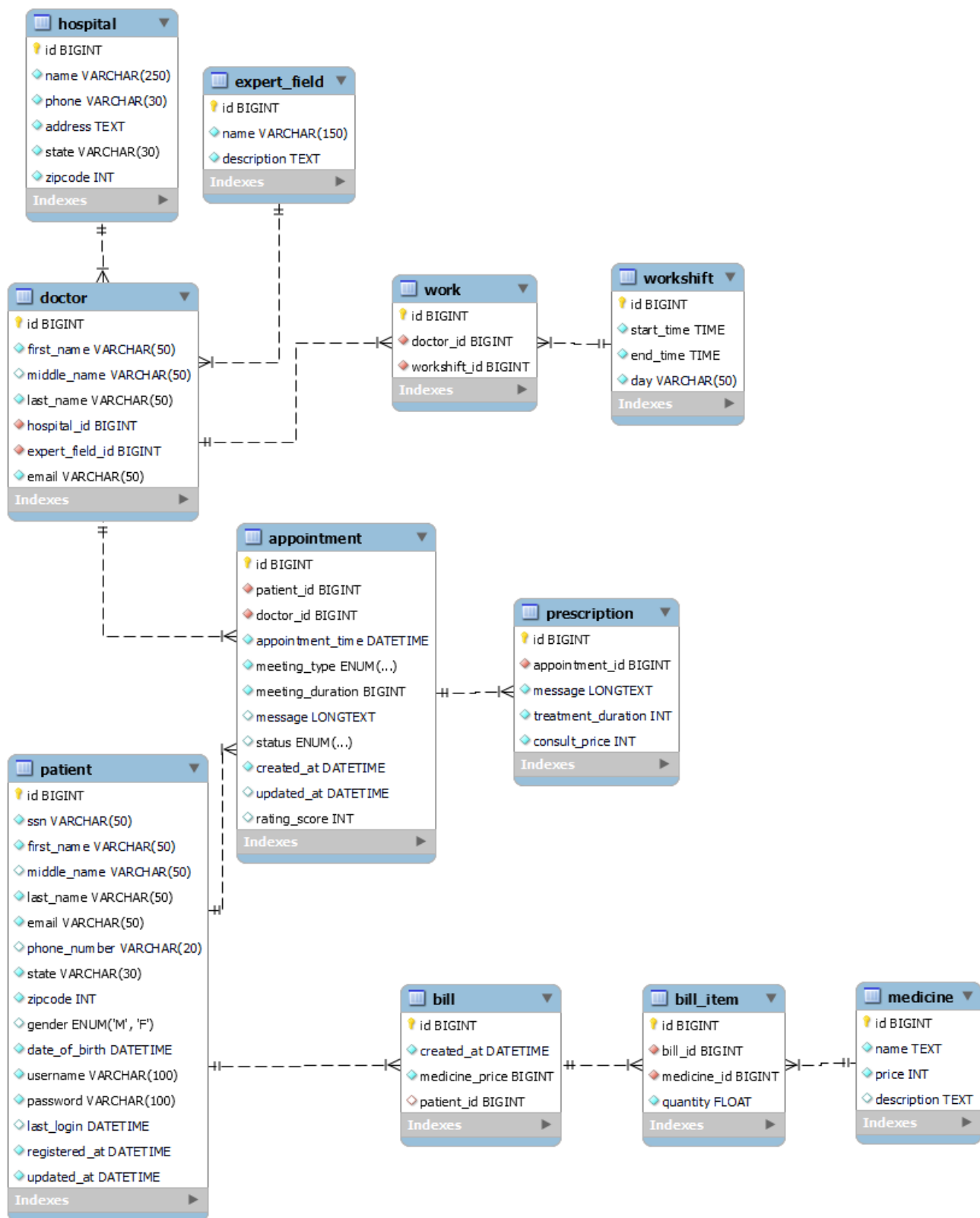
1) Project Description

Due to the covid, it's been hard for people to go and enquire about the information regarding taking an appointment from the doctor or consulting a doctor physically or online means. Not only that, it's even difficult for citizens to go to the medical store and buy some medicines. So, we decided to create an application to provide a doctor appointment service named Tindoc. Tindoc is a doctor appointment management system. When making an appointment, the patient can select a medical problem and choose a doctor that is an expert in a specific field. Patients can also add symptoms in detail into the appointment request. The application supports both face-to-face and online appointments. After the diagnosis with a doctor, patients can order medicine by themselves in the cases of online diagnosis. This system supports both online and offline meetings. The application also has features for a patient to give a rating score after an appointment.

2) Conceptual Design



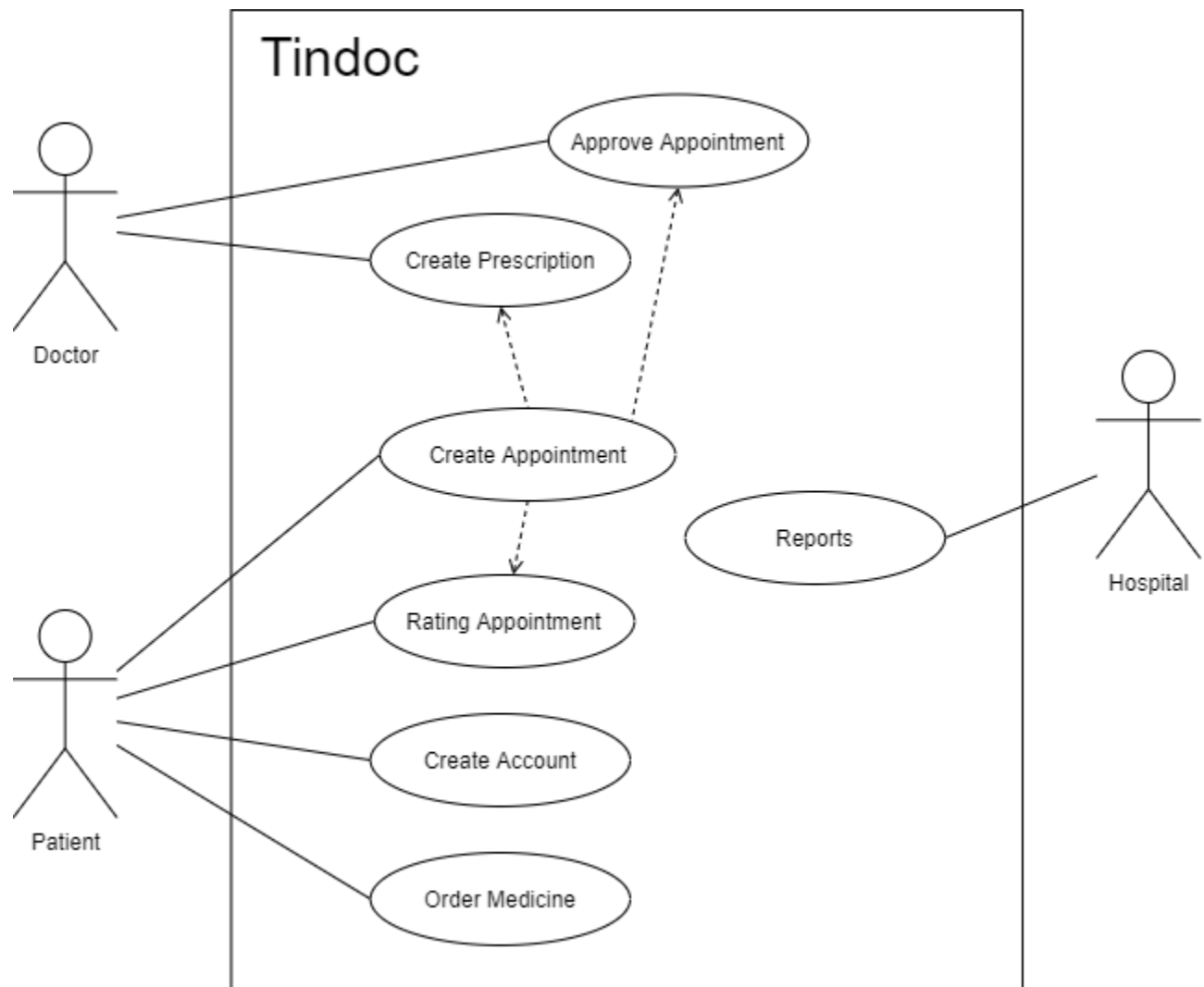
3) Relational database schema diagram



4) Logical diagram



5) Use Case Diagram



6) Data dictionary

Patient

Column	Datatype	Allow Null	Key	Reference	Description
id	BIGINT	No	PK, AI		ID of patient
ssn	VARCHAR(50)	No			Identification Number of patient
first_name	VARCHAR(50)	No			First Name of patient
middle_name	VARCHAR(50)	Yes			Middle Name of patient
last_name	VARCHAR(50)	No			Last Name of patient
email	VARCHAR(50)	No			Email of patient
phone_number	VARCHAR(20)	Yes			Phone number of patient
state	VARCHAR(30)	No			State or province of patient
zipcode	INT	No			Zipcode of patient
gender	ENUM('M', 'F')	Yes			Gender 'M' = male 'F' = female
date_of_birth	DATETIME	No			Birth date of patient
username	VARCHAR(30)	No			Username for application login
password	VARCHAR(30)	No			Password for application login
last_login	DATETIME	Yes			Latest login date
registered_at	DATETIME	No			Patient created date
updated_at	DATETIME	No			Latest information update date

Hospital

Column	Data type	Allow Null	Key	Reference	Description
id	BIGINT	No	PK, AI		Hospital id
name	VARCHAR(250)	No			Hospital name
phone	VARCHAR(30)	No			Hospital phone number
address	TEXT	No			Hospital address
state	VARCHAR(30)	No			The state hospital is located at
zipcode	INT	No			The hospital zip code number

Doctor

Column	Datatype	Allow NULL	KEY	Reference	Description
id	BIGINT	NO	PK		ID of doctor
first_name	VARCHAR(50)	NO			First name of doctor
middle_name	VARCHAR(50)	YES			Middle name of doctor
last_name	VARCHAR(50)	NO			Last name of doctor
hospital_id	BIGINT	NO	FK	hospital.id	ID of hospital which doctor works in
expert_field_id	BIGINT	NO	FK	expert_field.id	ID of expert field of doctor
email	VARCHAR(50)	NO			E-mail address of doctor

Appointment

Column	Datatype	Allow null	Key	Reference	Description
id	int	no	primary		Id of the appointment
patient_id	int	no	foreign	patient.id	Id of the patient in the database
doctor_id	int	no	foreign	doctor.id	Id of the doctor in the database
appointment_time	datetime	no			Appointment time
meeting_type	varchar	no			Define if the appointment is online or offline
meeting_duration	int	no			Duration of the appointment in minutes
status	varchar	no			Define the appointment status between three states: new, accepted, or rejected
created_at	datetime	no			Inform about the date of creation of the appointment
updated_at	datetime	yes			Inform about the date of the last updating of the appointment
rating_score	int	yes			Rating from one to five of the appointment by the patient
message	text	yes			Space in which the patient can provide further details to the doctor

Medicine

Column	Datatype	Allow null	Key	Reference	Description
id	int	no	PK, AI		Id of the drug
name	varchar	no			Name of the drug
price	int	no			Price of the drug in baht
description	text	no			Composition and instructions of use of the drug.

Bill

Column	Datatype	Allow Null	Key	Reference	Description
id	BIGINT	No	PK, AI		ID of medicine bill
created_at	DATETIME	Yes			Bill created date
medicine_price	BIGINT	No			Total price of medicine
patient_id	BIGINT	No	FK	patient.id	Owner of this bill

Expert field

Column	Datatype	Allow Null	Key	Reference	Description
id	BIGINT	No	PK, AI		ID of expert field categories
name	VARCHAR(150)	No			Expert field or problem categories name
description	TEXT	No			Description of that expert field

Work

Column	Datatype	Allow Null	Key	Reference	Description
id	BIGINT	No	PK, AI		ID of work
doctor_id	BIGINT	No	FK	doctor.id	ID of doctor
workshift_id	BIGINT	No	FK	workshift.id	ID of work shift.

Bill item

Column	Datatype	Allow Null	Key	Reference	Description
id	BIGINT	No	PK, AI		ID of bill item
bill_id	DATETIME	No	FK	bill.id	ID of bill
medicine_id	BIGINT	No	FK	medicine.id	ID of medicine
quantity	BIGINT	No			Quantity of the medicine selected by the patient for billing.

Workshift

Column	Data type	Allow Null	Key	Reference	Description
id	BIGINT	No	PK, AI		Workshift id
start_time	Time	No			The start time of each work shift
end_time	Time	No			The end time of each work shift
day	VARCHAR(50)	No			Day (7 days of the week)

Prescription

Column	Datatype	Allow NULL	KEY	Reference	Description
id	BIGINT	NO	PK, AI		ID of prescription
appointment_id	BIGINT	NO	FK	appointment.id	ID of appointment which the doctor create the prescription
message	LONGTEXT	NO			Message that the doctor write to the patient
treatment_duration	INT	NO			How long the petient should be treated (by medicine or any suggestion from the doctor)
consult_price	INT	NO			Doctor consultation fee

7) Database Population and SQL data commands

1.Patient

We used 1000 records of mock data for this table

Example:

```
insert into patient (id, ssn, first_name, middle_name, last_name, email, phone_number, state, zipcode, gender, date_of_birth, registered_at, username, password, last_login, updated_at) values (1, '667-57-6722', 'Kellia', 'Hulda', 'Kiljan', 'hkiljan0@nps.gov', '916-644-8832', 'California', '89252', 'F', '1977-01-25 04:42:32', '2020-07-02 14:06:13', 'hkiljan0', 'cde051bdccb11b2569bee9e2bafc074a', '2021-05-25 21:30:36', '2020-10-04 18:25:30');
```

2.Hospital

We used 5 records of mock data for this table

Example:

```
insert into hospital (id, name, phone, address, state, zipcode) values (1, 'Ardea golieth', '504-938-7566', '688 3rd Way', 'Louisiana', '94481');
```

3.Doctor

We used 50 records of mock data for this table

Example:

```
insert into doctor (id, first_name, middle_name, last_name, email, hospital_id, expert_field_id) values (10, 'Kinna', null, 'Bucksey', 'kbucksey9@comcast.net', 4, 3);
```

4.Appointment

We used 500 records of mock data for this table

Example:

```
insert into appointment (id, patient_id, doctor_id, appointment_time, meeting_type, meeting_duration, status, created_at, updated_at, rating_score, message) values (5, 393, 31, '2020-09-14 21:07:39', 'online', 80, 'new', '2020-09-29 01:36:23', '2020-09-18 21:56:37', 4, 'This is message from patient to doctor');
```

5.Medicine

We used 200 records of mock data for this table

Example:

```
insert into medicine (id, name, price, description) values (56, 'quetiapine fumarate', 70, 'Nondisplaced lateral mass fracture of first cervical vertebra');
```

6.Bill

We used 600 records of mock data for this table

Example:

```
insert into bill (id, created_at, medicine_price, patient_id) values (35, '2021-04-02 11:45:43', 384, 577);
```

7.Expert field

We used 10 records of mock data for this table

Example:

```
insert into expert_field (id, name, description) values (9, 'Orthopedic', 'the branch of surgery concerned with conditions involving the musculoskeletal system. Orthopaedic surgeons use both surgical and nonsurgical means to treat musculoskeletal trauma spine diseases sports injuries degenerative diseases infections tumors and congenital disorders.');
```

8.Work

We used 250 records of mock data for this table

Example:

```
insert into work (id, doctor_id, workshift_id) values (4, 9, 16);
```

9.Bill item

We used 1000 records of mock data for this table

Example:

```
insert into bill_item (id, bill_id, medicine_id, quantity) values (28, 203, 50, 17);
```

10.Workshift

We used 21 records of mock data for this table

Example:

```
insert into workshift (id, day, start_time, end_time) values (15, 'Fr', '16:00:00', '00:00:00');
```

11.Prescription

We used 500 records of mock data for this table

Example:

```
insert into prescription (id, appointment_id, message, treatment_duration, consult_price) values (16, 16, 'Maecenas ut massa quis augue luctus tincidunt.', 46, 200)
```

Sample Update query:

```
Update patient set middle_name='venkata' where id=1;
```

Sample Delete query:

```
Delete from doctor where id=1;
```

8) Important Queries and Reports

Done by 122556 Harold Popluhar

1. Prescription from each doctor

```
SELECT d.id AS doctor_id,
       d.first_name,
       d.last_name,
       p.id AS prescription_id,
       p.message,
       p.treatment_duration,
       p.consult_price
FROM   prescription p
       INNER JOIN appointment a
           ON p.appointment_id = a.id
       INNER JOIN doctor d
           ON a.doctor_id = d.id
ORDER BY d.id;
```

	doctor_id	first_name	last_name	prescription_id	message	treatment_duration	consult_price
▶	1	Lucinda	Eykel	163	Duis ac nibh.	3	400
	1	Lucinda	Eykel	316	Nullam molestie nibh in lectus.	8	500
	1	Lucinda	Eykel	327	Vivamus metus arcu, adipiscing molestie, hendr...	32	450
	1	Lucinda	Eykel	431	Etiam faucibus cursus urna.	26	400
	1	Lucinda	Eykel	443	In congue.	25	450
	1	Lucinda	Eykel	476	Donec posuere metus vitae ipsum.	56	200
	2	Edsel	McLugish	3	Vestibulum sed magna at nunc commodo placerat.	46	50
	2	Edsel	McLugish	40	Integer aliquet, massa id lobortis convallis, tort...	60	50
	2	Edsel	McLugish	45	Nulla mollis molestie lorem.	29	450
	2	Edsel	McLugish	55	In hac habitasse platea dictumst.	17	150
	2	Edsel	McLugish	204	Quisque erat eros, viverra eget, congue eget, ...	23	400
	2	Edsel	McLugish	289	Vestibulum ante ipsum primis in faucibus orci luct...	48	300

2. List of medicine the patient has ordered

```
SELECT p.id AS patient_id,
       p.first_name,
       p.last_name,
       m.id AS medicine_id,
       m.NAME
FROM   patient p
       INNER JOIN bill b
           ON p.id = b.patient_id
```

```

INNER JOIN bill_item b_i
    ON b_i.bill_id = b.id
INNER JOIN medicine m
    ON m.id = b_i.medicine_id
ORDER BY p.id ASC;

```

	patient_id	first_name	last_name	medicine_id	name
▶	5	Ganny	Longfield	23	TEMOVATE
	5	Ganny	Longfield	69	JIANZE SURGICAL SCRUB BRUSH NAIL CLEANER
	6	Blinni	Dmitrievski	74	Amlodipine Besylate and Benazepril Hydrochloride
	6	Blinni	Dmitrievski	86	Pentoxifylline
	8	Osbourne	Duckett	12	Trazodone Hydrochloride
	8	Osbourne	Duckett	40	Nicardipine Hydrochloride
	8	Osbourne	Duckett	56	quetiapine fumarate
	9	Tanitansy	Giovanetti	25	Amitriptyline Hydrochloride
	9	Tanitansy	Giovanetti	48	Giorgio Armani Face Fabric Second Skin Nude M...
	9	Tanitansy	Giovanetti	55	rizatriptan benzoate
	12	Ulrikaumeko	Conman	12	Trazodone Hydrochloride
	12	Ulrikaumeko	Conman	56	quetiapine fumarate

3. Typical disease by each quarter

```

SELECT      w.id,
            w.NAME,
            w.prob_first_quarter,
            x.prob_second_quarter,
            y.prob_third_quarter,
            z.prob_fourth_quarter
FROM        (
            SELECT      wa.id,
                        wa.NAME,
                        (wa.number_of_cases_first_quarter / wb.total_of_cases) AS
prob_first_quarter
            FROM        (
                        SELECT      ef.id,
                                    ef.NAME,
                                    Count(ef.id) AS
number_of_cases_first_quarter
            FROM        expert_field ef
            INNER JOIN  doctor d
            ON          d.expert_field_id = ef.id
            INNER JOIN  appointment a
            ON          a.doctor_id = d.id
            WHERE       Month(a.appointment_time) BETWEEN 1 AND
3
            GROUP BY   ef.id
            ORDER BY   Count(ef.id) DESC) AS wa
            INNER JOIN  (
                        SELECT      Count(ef.id) AS total_of_cases
            FROM        expert_field ef
            INNER JOIN  doctor d

```

```

ON          d.expert_field_id = ef.id
INNER JOIN  appointment a
ON          a.doctor_id = d.id
WHERE       Month(a.appointment_time) BETWEEN 1 AND

3) AS wb) AS w
INNER JOIN (
    (
        SELECT      wa.id,
                    wa.NAME,
                    wa.number_of_cases_second_quarter / wb.total_of_cases AS
prob_second_quarter
FROM              (
                    SELECT      ef.id,
                                ef.NAME,
                                count(ef.id) AS
number_of_cases_second_quarter
FROM              expert_field ef
INNER JOIN        doctor d
ON                d.expert_field_id = ef.id
INNER JOIN        appointment a
ON                a.doctor_id = d.id
WHERE             month(a.appointment_time) BETWEEN 4 AND
6
                    GROUP BY    ef.id
                    ORDER BY    count(ef.id) DESC) AS wa
                INNER JOIN
                (
                    SELECT      count(ef.id) AS total_of_cases
FROM              expert_field ef
INNER JOIN        doctor d
ON                d.expert_field_id = ef.id
INNER JOIN        appointment a
ON                a.doctor_id = d.id
WHERE             month(a.appointment_time) BETWEEN 4 AND

6) AS wb)) AS x
ON          x.id = w.id
INNER JOIN (
    (
        SELECT      wa.id,
                    wa.NAME,
                    wa.number_of_cases_third_quarter / wb.total_of_cases AS
prob_third_quarter
FROM              (
                    SELECT      ef.id,
                                ef.NAME,
                                count(ef.id) AS
number_of_cases_third_quarter
FROM              expert_field ef
INNER JOIN        doctor d
ON                d.expert_field_id = ef.id
INNER JOIN        appointment a
ON                a.doctor_id = d.id
WHERE             month(a.appointment_time) BETWEEN 7 AND
9
                    GROUP BY    ef.id
                    ORDER BY    count(ef.id) DESC) AS wa
                INNER JOIN
                (
                    SELECT      count(ef.id) AS total_of_cases
FROM              expert_field ef
INNER JOIN        doctor d

```


G2 Tindoc Project

```

ON          d.expert_field_id = ef.id
INNER JOIN  appointment a
ON          a.doctor_id = d.id
WHERE       month(a.appointment_time) BETWEEN 7 AND

9) AS wb)) AS y
ON          w.id = y.id
INNER JOIN  (
    (
        SELECT      wa.id,
                    wa.NAME,
                    wa.number_of_cases_fourth_quarter / wb.total_of_cases AS
prob_fourth_quarter
FROM              (
                    SELECT      ef.id,
                                ef.NAME,
                                count(ef.id) AS
number_of_cases_fourth_quarter
FROM              expert_field ef
INNER JOIN        doctor d
ON                d.expert_field_id = ef.id
INNER JOIN        appointment a
ON                a.doctor_id = d.id
WHERE             month(a.appointment_time) BETWEEN 10 AND

12

GROUP BY        ef.id
ORDER BY        count(ef.id) DESC) AS wa
INNER JOIN      (
        SELECT      count(ef.id) AS total_of_cases
FROM              expert_field ef
INNER JOIN        doctor d
ON                d.expert_field_id = ef.id
INNER JOIN        appointment a
ON                a.doctor_id = d.id
WHERE             month(a.appointment_time) BETWEEN 10 AND      12) AS wb)) AS z
ON          w.id = z.id

```

	id	name	Prob_first_quarter	Prob_second_quarter	Prob_third_quarter	Prob_fourth_quarter
►	4	Psychiatry	0.2066	0.2308	0.2000	0.1679
	2	Dermatology	0.1901	0.1197	0.1760	0.1022
	5	Infectious disease	0.1322	0.1624	0.1360	0.1387
	1	Cardiology	0.1157	0.1624	0.1040	0.1679
	3	Respiratory	0.0909	0.0684	0.1520	0.1314
	10	Urology	0.0909	0.0940	0.0800	0.0949
	7	Nephrology	0.0579	0.0256	0.0240	0.0219
	8	Neurology	0.0496	0.0769	0.0640	0.0511
	6	Internal medicine	0.0413	0.0342	0.0560	0.0803
	9	Orthopedic	0.0248	0.0256	0.0080	0.0438

4. The average treatment duration for each medical problem

```

SELECT ef.id,
       ef.NAME,
       Avg(p.treatment_duration) AS average_treatment_duration,

```

```

FROM    expert_field ef
        INNER JOIN doctor d
            ON d.expert_field_id = ef.id
        INNER JOIN appointment a
            ON a.doctor_id = d.id
        INNER JOIN prescription p
            ON p.appointment_id = a.id
GROUP BY ef.id;

```

	id	name	average_treatment_duration
►	1	Cardiology	29.1884
	2	Dermatology	29.3014
	3	Respiratory	29.2857
	4	Psychiatry	30.7200
	5	Infectious disease	28.8592
	6	Internal medicine	26.1852
	7	Nephrology	29.3125
	8	Neurology	24.1000
	9	Orthopedic	37.7692
	10	Urology	31.9111

5. Get a report of previous medical problems of the patient.

```

SELECT p.id,
       p.first_name,
       p.last_name,
       ef.NAME,
       presc.treatment_duration,
       presc.message
FROM    expert_field ef
        INNER JOIN doctor d
            ON d.expert_field_id = ef.id
        INNER JOIN appointment a
            ON a.doctor_id = d.id
        INNER JOIN patient p
            ON a.patient_id = p.id
        INNER JOIN prescription presc
            ON a.id = presc.appointment_id
WHERE   p.id = 1 -- id of the patient the doctor wants to check
ORDER BY ef.id;

```

	id	first_Name	last_Name	name	treatment_duration	message
▶	1	Kellia	Kiljan	Psychiatry	59	Duis at velit eu est congue elementum.
	1	Kellia	Kiljan	Infectious disease	48	Vestibulum quam sapien, varius ut, blandit non, ...

6. Get the right distribution of doctors in each medical sector (number of appointment hours over number of working hours by doctors)

```

SELECT b1.h_id,
       b1.ef_id,
       b1.ef_name,
       b1.average_of_appointment_hours_per_week,
       b2.number_of_doctor,
       b2.number_of_working_hours_in_the_sector,
       b2.average_number_of_working_hours_per_doctor_per_week
FROM   (SELECT a1.h_id,
               a1.ef_id,
               a1.ef_name,
               Avg(a1.total_appointment_minute_per_week) / 60 AS
               average_of_appointment_hours_per_week
        FROM   (SELECT h.id AS h_id,
                       ef.id AS ef_id,
                       ef.NAME AS ef_name,
                       Week(a.appointment_time) AS week,
                       Sum(a.meeting_duration) AS
                       total_appointment_minute_per_week
                FROM   expert_field ef
                      INNER JOIN doctor d
                          ON d.expert_field_id = ef.id
                      INNER JOIN appointment a
                          ON a.doctor_id = d.id
                      INNER JOIN hospital h
                          ON d.hospital_id = h.id
                GROUP BY Week(a.appointment_time),
                       ef.id,
                       h.id
                ORDER BY Week(a.appointment_time)) AS a1
        GROUP BY a1.h_id,
                 a1.ef_id
        ORDER BY a1.h_id) AS b1
INNER JOIN (SELECT
a2.h_id,
a2.ef_id,
a2.ef_name,
Count(a2.doc_id) AS number_of_doctor,
Sum(a2.number_of_working_hours) AS
number_of_working_hours_in_the_sector
,
Avg(a2.number_of_working_hours) AS
average_number_of_working_hours_per_doctor_per_week
FROM   (SELECT ef.id
        AS

```

```

        ef_id,
        ef.NAME
    AS
    ef_name,
    d.id
    AS
    doc_id,
    h.id
    AS
    h_id,
    Sum(Abs(Timestampdiff(hour, wsh.end_time,
        wsh.start_time))) AS
    number_of_working_hours
FROM expert_field ef
INNER JOIN doctor d
    ON d.expert_field_id = ef.id
INNER JOIN work w
    ON w.doctor_id = d.id
INNER JOIN workshift wsh
    ON wsh.id = w.workshift_id
INNER JOIN hospital h
    ON d.hospital_id = h.id
GROUP BY d.id
ORDER BY h.id) AS a2
GROUP BY a2.h_id,
    a2.ef_id
ORDER BY a2.h_id) AS b2
ON b2.h_id = b1.h_id
AND b2.ef_id = b1.ef_id
ORDER BY b1.h_id;

```

	h_id	ef_id	ef_name	average_of_appointment_hours_f	number_of_doctor	number_of_working_hours_in_the_sec	average_number_of_working_hours_per_
►	1	1	Cardiology	0.80833333	2	112	56.0000
	1	3	Respiratory	0.87037037	1	96	96.0000
	1	4	Psychiatry	0.80952381	1	40	40.0000
	2	1	Cardiology	1.30158730	2	120	60.0000
	2	2	Dermatology	1.28240741	3	144	48.0000
	2	3	Respiratory	0.73055556	1	56	56.0000
	2	4	Psychiatry	0.66805556	1	56	56.0000
	2	5	Infectious disease	1.00277778	2	120	60.0000
	2	10	Urology	0.88235294	2	64	32.0000
	3	1	Cardiology	0.86041667	1	16	16.0000

Done by 122149 Saratoon Khantasima

1. Most needed field of expertise filter by age and gender**

```

SELECT (SELECT e.name
        FROM myfirstdb.patient p
        INNER JOIN myfirstdb.appointment a
            ON p.id = a.patient_id

```

```

        INNER JOIN myfirstdb.doctor d
            ON a.doctor_id = d.id
        INNER JOIN myfirstdb.expert_field e
            ON d.expert_field_id = e.id
WHERE p.gender = 'M'
      AND Timestampdiff(year, p.date_of_birth, Now()) < 20
GROUP BY e.id
ORDER BY Count(e.id) DESC
LIMIT 1) AS "Most needed field of expertise for Male age below 20",
(SELECT e.name
FROM   myfirstdb.patient p
      INNER JOIN myfirstdb.appointment a
            ON p.id = a.patient_id
      INNER JOIN myfirstdb.doctor d
            ON a.doctor_id = d.id
      INNER JOIN myfirstdb.expert_field e
            ON d.expert_field_id = e.id
WHERE  p.gender = 'M'
      AND Timestampdiff(year, p.date_of_birth, Now()) > 20
      AND Timestampdiff(year, p.date_of_birth, Now()) < 40
GROUP BY e.id
ORDER BY Count(e.id) DESC
LIMIT 1) AS
"Most needed field of expertise for Male age between 20 and 40",
(SELECT e.name
FROM   myfirstdb.patient p
      INNER JOIN myfirstdb.appointment a
            ON p.id = a.patient_id
      INNER JOIN myfirstdb.doctor d
            ON a.doctor_id = d.id
      INNER JOIN myfirstdb.expert_field e
            ON d.expert_field_id = e.id
WHERE  p.gender = 'M'
      AND Timestampdiff(year, p.date_of_birth, Now()) > 40
GROUP BY e.id
ORDER BY Count(e.id) DESC
LIMIT 1) AS "Most needed field of expertise for Male age above 40",
(SELECT e.name
FROM   myfirstdb.patient p
      INNER JOIN myfirstdb.appointment a
            ON p.id = a.patient_id
      INNER JOIN myfirstdb.doctor d
            ON a.doctor_id = d.id
      INNER JOIN myfirstdb.expert_field e
            ON d.expert_field_id = e.id
WHERE  p.gender = 'F'
      AND Timestampdiff(year, p.date_of_birth, Now()) < 20
GROUP BY e.id
ORDER BY Count(e.id) DESC
LIMIT 1) AS "Most needed field of expertise for Female age below 20",
(SELECT e.name
FROM   myfirstdb.patient p
      INNER JOIN myfirstdb.appointment a
            ON p.id = a.patient_id

```

```

        INNER JOIN myfirstdb.doctor d
            ON a.doctor_id = d.id
        INNER JOIN myfirstdb.expert_field e
            ON d.expert_field_id = e.id
    WHERE p.gender = 'F'
        AND Timestampdiff(year, p.date_of_birth, Now()) > 20
        AND Timestampdiff(year, p.date_of_birth, Now()) < 40
    GROUP BY e.id
    ORDER BY Count(e.id) DESC
    LIMIT 1) AS
    "Most needed field of expertise for Female age between 20 and 40",
    (SELECT e.name
    FROM myfirstdb.patient p
        INNER JOIN myfirstdb.appointment a
            ON p.id = a.patient_id
        INNER JOIN myfirstdb.doctor d
            ON a.doctor_id = d.id
        INNER JOIN myfirstdb.expert_field e
            ON d.expert_field_id = e.id
    WHERE p.gender = 'F'
        AND Timestampdiff(year, p.date_of_birth, Now()) > 40
    GROUP BY e.id
    ORDER BY Count(e.id) DESC
    LIMIT 1) AS "Most needed field of expertise for Female age above 40";

```

	Most needed field of expertise for Male age below 20	Most needed field of expertise for Male age between 20 and 40	Most needed field of expertise for Male age above 40
►	Psychiatry	Dermatology	Psychiatry

Most needed field of expertise for Female age below 20	Most needed field of expertise for Female age between 20 and 40	Most needed field of expertise for Female age above 40
Psychiatry	Psychiatry	Psychiatry

2. How many patients per month**

```

SELECT (SELECT Count(a.patient_id)
    FROM myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
            ON a.patient_id = p.id
    WHERE a.appointment_time BETWEEN
        '2020-09-01 00:00:00' AND '2020-10-01 00:00:00') AS
    "Number of patients in Sep 2020",
    (SELECT Count(a.patient_id)
    FROM myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
            ON a.patient_id = p.id
    WHERE a.appointment_time BETWEEN
        '2020-10-01 00:00:00' AND '2020-11-01 00:00:00') AS
    "Number of patients in Oct 2020",
    (SELECT Count(a.patient_id)

```

```

FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2020-11-01 00:00:00' AND '2020-12-01 00:00:00') AS
"Number of patients in Nov 2020",
(SELECT Count(a.patient_id)
FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2020-12-01 00:00:00' AND '2021-01-01 00:00:00') AS
"Number of patients in Dec 2020",
(SELECT Count(a.patient_id)
FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2021-01-01 00:00:00' AND '2021-02-01 00:00:00') AS
"Number of patients in Jan 2021",
(SELECT Count(a.patient_id)
FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2021-02-01 00:00:00' AND '2021-03-01 00:00:00') AS
"Number of patients in Feb 2021",
(SELECT Count(a.patient_id)
FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2021-03-01 00:00:00' AND '2021-04-01 00:00:00') AS
"Number of patients in Mar 2021",
(SELECT Count(a.patient_id)
FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2021-04-01 00:00:00' AND '2021-05-01 00:00:00') AS
"Number of patients in Apr 2021",
(SELECT Count(a.patient_id)
FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2021-05-01 00:00:00' AND '2021-06-01 00:00:00') AS
"Number of patients in May 2021",
(SELECT Count(a.patient_id)
FROM    myfirstdb.appointment a
        INNER JOIN myfirstdb.patient p
              ON a.patient_id = p.id
WHERE    a.appointment_time BETWEEN
        '2021-06-01 00:00:00' AND '2021-07-01 00:00:00') AS

```

G2 Tindoc Project

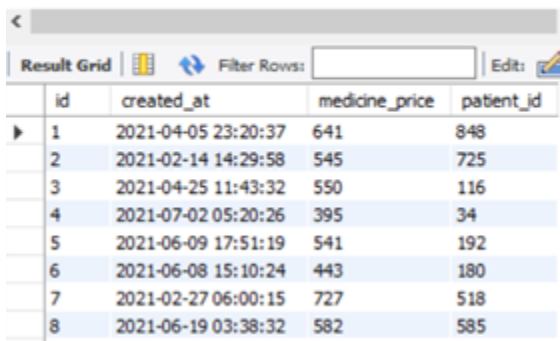
```
"Number of patients in Jun 2021",
(SELECT Count(a.patient_id)
FROM   myfirstdb.appointment a
      INNER JOIN myfirstdb.patient p
            ON a.patient_id = p.id
WHERE  a.appointment_time BETWEEN
      '2021-07-01 00:00:00' AND '2021-08-01 00:00:00') AS
"Number of patients in Jul 2021",
(SELECT Count(a.patient_id)
FROM   myfirstdb.appointment a
      INNER JOIN myfirstdb.patient p
            ON a.patient_id = p.id
WHERE  a.appointment_time BETWEEN
      '2021-08-01 00:00:00' AND '2021-09-01 00:00:00') AS
"Number of patients in Aug 2021";
```

Number of patients in Sep 2020	Number of patients in Oct 2020	Number of patients in Nov 2020	Number of patients in Dec 2020
20	33	50	54
Number of patients in Jan 2021	Number of patients in Feb 2021	Number of patients in Mar 2021	Number of patients in Apr 2021
45	43	33	38
Number of patients in May 2021	Number of patients in Jun 2021	Number of patients in Jul 2021	Number of patients in Aug 2021
35	44	49	36

3. See bill details

```
SELECT *
FROM   myfirstdb.bill;
```

```
96  -- See bill details
97  • SELECT * FROM myfirstdb.bill;
98
```



	id	created_at	medicine_price	patient_id
▶	1	2021-04-05 23:20:37	641	848
	2	2021-02-14 14:29:58	545	725
	3	2021-04-25 11:43:32	550	116
	4	2021-07-02 05:20:26	395	34
	5	2021-06-09 17:51:19	541	192
	6	2021-06-08 15:10:24	443	180
	7	2021-02-27 06:00:15	727	518
	8	2021-06-19 03:38:32	582	585

4. See the patient who is often sick*


```

SELECT patient_id,
       Count(id) AS "number of appointment"
FROM   myfirstdb.appointment
GROUP  BY patient_id
ORDER  BY Count(id) DESC;

```

	patient_id	number of appointment
▶	136	5
	353	5
	52	4
	329	4
	342	4
	3	3
	5	3
	13	3
	35	3
	39	3
	41	3

5. Number of case for each doctor*

```

SELECT d.id,
       d.first_name,
       d.middle_name,
       d.last_name,
       Count(a.id) AS "number of cases"
FROM   myfirstdb.doctor d
       INNER JOIN myfirstdb.appointment a
              ON d.id = a.doctor_id
GROUP  BY d.id;

```

	id	first_name	middle_name	last_name	number of cases
▶	1	Lucinda	Osanne	Eykel	6
	2	Edsel	MacFadzan	McLugish	11
	3	Donnie	Gooden	Guly	6
	4	Kaia	Quidenham	Lube	12
	5	Julienne	NULL	Cosins	7
	6	Alfi	Marnane	Semeradova	6
	7	Tamra	NULL	Scholar	8
	8	Lars	NULL	Latta	10
	9	Barry	Lower	St. Ledger	13
	10	Kinna	NULL	Bucksey	13
	11	Brandais	NULL	Battelle	7

6. Check the rating for doctor*

```

SELECT d.id,
       d.first_name,
       d.middle_name,
       d.last_name,
       Avg(a.rating_score) AS "Rating for doctor"
FROM   myfirstdb.appointment a
       INNER JOIN myfirstdb.doctor d
              ON a.doctor_id = d.id
GROUP BY d.id;

```

id	first_name	middle_name	last_name	Rating for doctor
1	Lucinda	Osanne	Eykel	2.3333
2	Edsel	MacFadzan	McLugish	2.9091
3	Donnie	Gooden	Guly	3.0000
4	Kaia	Quidenham	Lube	2.7500
5	Julienne	NULL	Cosins	3.8571
6	Alfi	Marnane	Semeradova	2.3333
7	Tamra	NULL	Scholar	2.3750
8	Lars	NULL	Latta	3.4000
9	Barry	Lower	St. Ledger	2.3846
10	Kinna	NULL	Bucksey	3.2308
11	Brandais	NULL	Battelle	4.1429

Done by 122410 Nopphawan Nurnuansuwan

1. Can see the appointment and prescription history of a particular patient.

```
SELECT ap.appointment_time AS 'Appointment
date/time',
       Concat(dr.first_name, ' ', dr.last_name) AS 'Doctor name',
       ap.meeting_type AS 'Meeting type',
       ap.meeting_duration AS 'Meeting duration
(min)',
       ap.message AS 'Appointment detail',
       ps.message AS 'Prescription
message',
       ps.consult_price AS 'Consult price (USD)'
FROM   ((patient pt
        LEFT JOIN appointment ap
              ON ap.patient_id = pt.id)
        LEFT JOIN prescription ps
              ON ap.id = ps.appointment_id)
        LEFT JOIN doctor dr
              ON ap.doctor_id = dr.id
WHERE  pt.id = 3
ORDER  BY ap.appointment_time DESC;
```

	Appointment date/time	Doctor name	Meeting type	Meeting duration (min)	Appointment detail	Prescription message	Consult price (USD)
▶	2021-07-01 16:37:20	Orsola Lindsell	offline	46	This is message from patient to doctor	Aliquam quis turpis eget elit sodales scelerisque.	150
	2021-05-05 22:41:46	Donnie Guly	offline	38	This is message from patient to doctor	In congue.	150
	2021-04-14 13:30:09	Kinna Bucksey	online	64	This is message from patient to doctor	Nam ultrices, libero non mattis pulvinar, nulla pe...	400

2. Can see average days that each doctor accepts an appointment.*

```
SELECT dr.first_name AS Doctor_name,
       Avg(( Datediff(ap.created_at, ap.updated_at) )) AS Average_day
FROM   appointment ap
       LEFT JOIN doctor dr
             ON ap.doctor_id = dr.id
WHERE  ap.status = 'accepted'
GROUP BY dr.id
ORDER BY average_day DESC;
```

	Doctor_name	Average_day
▶	Eba	95.6000
	Joshua	93.5000
	Gerta	91.6667
	Alida	90.6667
	Edsel	82.5000
	Randy	79.4000
	Dore	72.5000
	Kathy	65.0000

3. Number of patients living in each state in the selected hospital.*

```

SELECT pt.state,
       Count(pt.state) AS 'Number of patient'
FROM   patient pt
       LEFT JOIN appointment ap
           ON pt.id = ap.patient_id
       LEFT JOIN doctor dr
           ON ap.doctor_id = dr.id
       LEFT JOIN hospital hp
           ON dr.hospital_id = hp.id
WHERE  hp.id = 2
GROUP BY pt.state
ORDER BY pt.state;

```

	state	Number of patient
▶	Alabama	7
	Alaska	1
	Arizona	1
	California	7
	Colorado	1
	Connecticut	1
	District of Colu...	5

4. Get the most common disease in the state (Max number of cases in a specific field in each state).**

```

SELECT state,
       disease AS 'The most common disease',
       Max(ct) AS 'Number of case'
FROM   (SELECT pt.state AS State,
              ep.NAME   AS Disease,
              Count(*)  AS CT
        FROM   (patient pt
                RIGHT JOIN appointment ap
                    ON pt.id = ap.patient_id
                LEFT JOIN doctor dr
                    ON ap.doctor_id = dr.id
                LEFT JOIN expert_field ep
                    ON dr.expert_field_id = ep.id)
        GROUP BY ep.NAME,
                 pt.state) T

GROUP BY state
ORDER BY state;

```

	State	The most common disease	Number of case
►	Alabama	Cardiology	5
	Alaska	Respiratory	2
	Arizona	Psychiatry	4
	California	Infectious disease	10
	Colorado	Infectious disease	2
	Connecticut	Psychiatry	1
	District of Columbia	Urology	3
	Florida	Urology	8

5. Show all patients who didn't have any appointments and bills.

```

SELECT pt.ssn,
       Concat(pt.first_name, ' ', pt.last_name) AS 'Patient name'
FROM   patient pt
       LEFT JOIN appointment ap
           ON pt.id = ap.patient_id
       LEFT JOIN bill b
           ON pt.id = b.patient_id
WHERE  b.id IS NULL
       AND ap.id IS NULL;

```

	ssn	Patient name
▶	252-58-7697	Mona Winkell
	787-60-1560	Sarette Strodde
	390-84-0812	Rustin Longforth
	119-77-8053	Theodora Fider
	687-45-9303	Tallia Pyffe
	785-21-3994	Norma Durrand
	873-69-4853	Drake Leband
	642-03-1419	Hannie Tapson

6. Show the patient information with their current age, number of appointments that they have made, number of bills that they have created, total appointments and bills, and number of medicine that they have ordered.**

```

SELECT Concat(pt.first_name, ' ', pt.last_name)      AS 'Patient name',
       pt.phone_number,
       pt.email,
       pt.state,
       Timestampdiff(year, pt.date_of_birth, Now()) AS 'age',
       A.ap_no                                       AS Appointment_no,
       B.bill_no                                     AS Bill_no,
       (SELECT appointment_no + bill_no)            AS
Total_Appointment_Bill,
       BT.med_no                                     AS Medicine_no
FROM   patient pt
       LEFT JOIN (SELECT pt.id           AS id,
                          Count(ap.id)  AS Ap_no
                  FROM   patient pt
                  LEFT JOIN appointment ap
                        ON pt.id = ap.patient_id
                  GROUP BY pt.id) A
       ON A.id = pt.id
       LEFT JOIN (SELECT pt.id           AS id,
                          Count(b.id)    AS Bill_no
                  FROM   patient pt
                  LEFT JOIN bill b
                        ON pt.id = b.patient_id
                  GROUP BY pt.id) B
       ON B.id = pt.id
       LEFT JOIN (SELECT pt.id           AS id,
                          Count(bt.id)   AS Med_no
                  FROM   patient pt
                  LEFT JOIN bill b
                        ON pt.id = b.patient_id

```

```

LEFT JOIN bill_item bt
ON b.id = bt.bill_id
GROUP BY pt.id) BT
ON BT.id = pt.id
ORDER BY total_appointment_bill DESC;

```

	Patient name	phone_number	email	state	age	Appointment_no	Bill_no	Total_Appointment_Bill	Medicine_no
▶	Fidelity Heys	217-624-0363	hheys3r@biglobe.ne.jp	Illinois	17	5	2	7	4
	Inglis Scherer	269-404-3883	bscherer1f@cmu.edu	Michigan	17	4	2	6	4
	Cole Connew	678-723-6386	mconnewj@netscape.com	Georgia	36	2	3	5	1
	Issiah Keir	412-438-8151	rkeir18@cpanel.net	Pennsylvania	36	2	3	5	8
	Betteanne MacGuiness	325-467-0631	lmacguiness3a@taobao.com	Texas	37	3	2	5	1
	Farlie Hannent	573-628-4689	bhannent6d@squidoo.com	Missouri	49	2	3	5	9
	Lurleen Povah	612-812-8791	fpovah9q@ifeng.com	Minnesota	36	3	2	5	9
	Byrle Danniel	915-628-3989	kdanniel9s@w3.org	Texas	33	5	0	5	0

Done by 122442 Pasit Tiwawongrut

1. Patients can check the appointment time.

```

SELECT appointment.patient_id,
Concat(doctor.first_name, ' ', doctor.last_name) AS doctor_name,
appointment.appointment_time
FROM appointment
INNER JOIN doctor
ON appointment.doctor_id = doctor.id
WHERE patient_id = 13;

```

	patient_id	doctor_name	appointment_time
▶	13	Tamra Scholar	2021-01-27 04:11:13
	13	Edsel McLugish	2021-09-09 19:31:55
	13	Julienne Cosins	2020-11-23 04:28:26

2. Number of hospitals which treat the medical problem near to the location of the patient.

```

SELECT DISTINCT( hospital.id ) AS hospital_id,
hospital.NAME AS hospital_name,
hospital.state,
expert_field.NAME AS disease_field
FROM doctor
INNER JOIN hospital
ON doctor.hospital_id = hospital.id
INNER JOIN expert_field
ON doctor.expert_field_id = expert_field.id
WHERE doctor.expert_field_id = 4
AND hospital.state = 'Louisiana'
ORDER BY hospital.id;

```

G2 Tindoc Project

	hospital_id	hospital_name	state	disease_field
►	1	Ardea golieth	Louisiana	Psychiatry

3. Find which hospitals have more doctors than average.

```
SELECT hospital.id      AS hospital_id,
       hospital.NAME    AS hospital_name,
       hospital.state,
       Count(doctor.id) AS doctor_count
FROM   doctor
       INNER JOIN hospital
           ON doctor.hospital_id = hospital.id
GROUP BY hospital.id
HAVING doctor_count > (SELECT Avg(doctor_count)
                       FROM   (SELECT Count(doctor.id) AS doctor_count
                               FROM     doctor
                               INNER JOIN hospital
                                   ON doctor.hospital_id = hospital.id
                               GROUP BY hospital.id) count);
```

	hospital_id	hospital_name	state	doctor_count
►	2	Dasypus novemcinctus	Georgia	11
	3	Stercorarius longicausus	New Mexico	11
	4	Trichosurus vulpecula	Ohio	16

4. Doctors can check their work hours.

```
SELECT doctor.id,
       first_name,
       middle_name,
       last_name,
       start_time,
       end_time,
       day
FROM   doctor
       INNER JOIN work
           ON doctor.id = work.doctor_id
       INNER JOIN workshift
           ON work.workshift_id = workshift.id
WHERE  doctor.id = 18;
```


	id	first_name	middle_name	last_name	start_time	end_time	day
▶	18	Nevins	NULL	Wilcot	08:00:00	16:00:00	To
	18	Nevins	NULL	Wilcot	16:00:00	00:00:00	Mo
	18	Nevins	NULL	Wilcot	00:00:00	08:00:00	To
	18	Nevins	NULL	Wilcot	08:00:00	16:00:00	We
	18	Nevins	NULL	Wilcot	08:00:00	16:00:00	Su

5. Doctors can see the number of patients that make an appointment between particular times.

```

SELECT appointment.id,
       Concat(patient.first_name, ' ', patient.last_name) AS
patient_name,
       appointment_time,
       meeting_type,
       message,
       status,
       appointment.created_at,
       appointment.updated_at
FROM   appointment
       INNER JOIN patient
           ON appointment.patient_id = patient.id
WHERE  doctor_id = 1
       AND appointment_time BETWEEN '2021-08-01 23:59:59' AND
                                   '2021-09-09 23:59:59'
       AND status = 'accepted';

```

	id	patient_name	appointment_time	meeting_type	message	status	created_at	updated_at
▶	163	Dorian Dugan	2021-08-20 12:30:50	offline	This is message from patient to doctor	accepted	2021-01-11 19:52:59	2021-03-10 19:09:52
	316	Conny Possek	2021-09-04 11:51:33	offline	This is message from patient to doctor	accepted	2021-02-23 19:43:10	2020-11-22 09:26:41

6. Hospitals can see how many doctor for each work shift

```

SELECT workshift_id,
       workshift.day,
       workshift.start_time,
       workshift.end_time,
       Count(doctor_id)
FROM   work
       INNER JOIN workshift
           ON work.workshift_id = workshift.id
GROUP BY workshift_id;

```

	workshift_id	day	start_time	end_time	COUNT(doctor_id)
▶	1	Mo	00:00:00	08:00:00	12
	2	Mo	08:00:00	16:00:00	11
	3	Mo	16:00:00	00:00:00	10
	4	To	00:00:00	08:00:00	19
	5	To	08:00:00	16:00:00	14
	6	To	16:00:00	00:00:00	18
	7	We	00:00:00	08:00:00	12
	8	We	08:00:00	16:00:00	15
	9	We	16:00:00	00:00:00	19
	10	Th	00:00:00	08:00:00	8
	11	Th	08:00:00	16:00:00	11
	12	Th	16:00:00	00:00:00	7
	13	Fr	00:00:00	08:00:00	14
	14	Fr	08:00:00	16:00:00	11
	15	Fr	16:00:00	00:00:00	10
	16	Sa	00:00:00	08:00:00	6
	17	Sa	08:00:00	16:00:00	12
	18	Sa	16:00:00	00:00:00	16
	19	Su	00:00:00	08:00:00	9
	20	Su	08:00:00	16:00:00	7
	21	Su	16:00:00	00:00:00	9

7. Total cases for each expert field on the specific date.

```

SELECT expert_field_id,
       expert_field.NAME,
       expert_field.description,
       Count(appointment.id)
FROM   appointment
       INNER JOIN doctor
           ON appointment.doctor_id = doctor.id
       INNER JOIN expert_field
           ON expert_field.id = doctor.expert_field_id
WHERE  appointment_time BETWEEN '2020-01-01 23:59:59' AND '2021-09-09
23:59:59'
GROUP BY expert_field_id
ORDER BY expert_field_id;

```

	expert_field_id	name	description	COUNT(appointment.id)
▶	1	Cardiology	a branch of medicine that deals with the disorde...	69
	2	Dermatology	the branch of medicine dealing with the skin. It i...	73
	3	Respiratory	a medical specialty that deals with diseases invo...	56
	4	Psychiatry	the medical specialty devoted to the diagnosis p...	100
	5	Infectious disease	a medical specialty dealing with the diagnosis an...	71
	6	Internal medicine	the medical specialty dealing with the preventio...	27
	7	Nephrology	a specialty of adult internal medicine and pediat...	16
	8	Neurology	a branch of medicine dealing with disorders of t...	30
	9	Orthopedic	the branch of surgery concerned with condition...	13
	10	Urology	the branch of medicine that focuses on surgical ...	45

Done by 122050 Vineela Mukkamala

1. Show doctor information with their name, email,expert field, hospital which they belong to, no of appointments, no of working hours, average rating for a doctor, maximum rating, min rating of a doctor, no of patients who visited the doctor. **

```

SELECT dr.id,
       Concat(dr.first_name, ' ', dr.middle_name, ' ',
dr.last_name) AS
       'doctor name',
       a.a_count
AS
       'no of appointments',
       b.b_avg_rat
AS
       'averating rating of the doctor',
       c.c_max_rat
AS
       'maximum rating of the doctor',
       d.d_min_rat
AS
       'minimum rating of the doctor',
       e.e_cnt_of_patients_who_visited_him
AS
       'no of patients who have visited this doctor',
       ef.NAME
AS
       'expert field name',
       h.NAME
AS
       'hospital name',
       dr.email
FROM myfirstdb.doctor dr
     INNER JOIN (SELECT Count(ap.id) AS a_count,
                        dr.id        AS a_doctor_id
                  FROM myfirstdb.doctor dr

```

```

                INNER JOIN myfirstdb.appointment ap
                        ON ap.doctor_id = dr.id
        GROUP BY dr.id) AS a
    ON a.a_doctor_id = dr.id
    INNER JOIN (SELECT Round(Avg(ap.rating_score)) AS b_avg_rat,
                        dr.id AS
b_doctor_id
                FROM myfirstdb.appointment ap
                INNER JOIN myfirstdb.doctor dr
                        ON ap.doctor_id = dr.id
                GROUP BY dr.id) AS b
    ON b.b_doctor_id = dr.id
    INNER JOIN (SELECT Max(ap.rating_score) AS c_max_rat,
                        dr.id AS c_doctor_id
                FROM myfirstdb.appointment ap
                INNER JOIN myfirstdb.doctor dr
                        ON ap.doctor_id = dr.id
                GROUP BY dr.id) AS c
    ON c.c_doctor_id = dr.id
    INNER JOIN (SELECT Min(ap.rating_score) AS d_min_rat,
                        dr.id AS d_doctor_id
                FROM myfirstdb.appointment ap
                INNER JOIN myfirstdb.doctor dr
                        ON ap.doctor_id = dr.id
                GROUP BY dr.id) AS d
    ON d.d_doctor_id = dr.id
    INNER JOIN (SELECT Count(x.p_id) AS
e_cnt_of_patients_who_visited_him,
                        x.d_id AS e_doctor_id
                FROM (SELECT dr.id AS d_id,
                        Count(p.id) AS patient_count,
                        p.id AS p_id
                FROM doctor dr
                INNER JOIN myfirstdb.appointment
ap
                        ON ap.doctor_id = dr.id
                INNER JOIN myfirstdb.patient p
                        ON p.id = ap.patient_id
                GROUP BY dr.id,
                        p.id
                HAVING patient_count > 0
                ORDER BY dr.id) AS x
                GROUP BY x.d_id) AS e
    ON e.e_doctor_id = dr.id
    INNER JOIN myfirstdb.hospital h
    ON dr.hospital_id = h.id
    INNER JOIN myfirstdb.expert_field ef
    ON dr.expert_field_id = ef.id
ORDER BY dr.id;

```

G2 Tindoc Project

	id	doctor name	no of appointments	averating rating of the doctor	maximum rating of the doctor	minimum rating of the doctor	no of patients who have visited this doctor
▶	1	Mable La Batie Wink	5	3	5	2	5
	2	Thorny Dalgarnowch Tregien	9	3	5	1	9
	3	Adolpho Astill Lambie	13	2	5	1	13

	maximum rating of the doctor	minimum rating of the doctor	no of patients who have visited this doctor	expert field name	hospital name	email
	5	2	5	Respiratory	Pterocles gutturalis	mwink0@t-online.de
	5	1	9	Urology	Eira barbata	ttregien1@1688.com
	5	1	13	Infectious disease	Pseudocheirus peregrinus	alamble2@cyberchimps.com

2) Get the average duration spent by each doctor on patients and display his details along with the expert field.*

```
SELECT dr.id,
       Concat(dr.first_name, ' ', COALESCE(dr.middle_name, ' '), ' ',
             dr.last_name) AS
       doctor_name,
       dr.email,
       dr.expert_field_id,
       ef.NAME,
       Avg(ap.meeting_duration)
       AS 'averageDuration spent on each meeting (min)'
FROM   myfirstdb.doctor dr
       INNER JOIN myfirstdb.appointment ap
               ON ap.doctor_id = dr.id
       INNER JOIN myfirstdb.expert_field ef
               ON ef.id = dr.expert_field_id
GROUP BY dr.id
ORDER BY doctor_id;
```

id	doctor_name	email	expert_field_id	NAME	averageDuration spent on each meeting (min)
1	Mable La Batie Wink	mwink0@t-online.de	3	Respiratory	50.0000
2	Thorny Dalgarnowch Tregien	ttregien1@1688.com	10	Urology	46.7778
3	Adolpho Astill Lambie	alamble2@cyberchimps.com	5	Infectious disease	55.6154
4	Karla Wareham Horbath	khorbath3@deliciousdays.com	5	Infectious disease	54.5000
5	Alexander Iris Trench	atrench4@chase.com	10	Urology	54.0444

3) Total bill of the patient made till date. *

```
SELECT p.id,
       Concat(p.first_name, ' ', p.middle_name, ' ', p.last_name) AS
       patient_name,
       Ifnull(b.medicine_price, 0) AS
       'medicine price',
       Ifnull(pr.consult_price, 0) AS
       'consultation price',
       ( Ifnull(b.medicine_price, 0)
         + Ifnull(pr.consult_price, 0) ) AS
```

```

        'total bill made by the patient till date',
        Avg(( Ifnull(b.medicine_price, 0)
              + Ifnull(pr.consult_price, 0) ))
        'average amount purchased by the patient till date'
FROM    myfirstdb.patient p
        LEFT JOIN myfirstdb.bill b
              ON p.id = b.patient_id
        LEFT JOIN myfirstdb.appointment ap
              ON p.id = ap.patient_id
        LEFT JOIN myfirstdb.prescription pr
              ON ap.id = pr.appointment_id
GROUP BY p.id
ORDER BY p.id;

```

id	patient_name	medicine price	consultation price	total bill made by the patient till date	average amount purchased by the patient till date
1	Eulalie Junina Loffhead	0	0	0	0.0000
2	Kaiser Nikita Oliff	526	100	626	621.6667
3	Reinwald Ferdinand Rowlatt	364	0	364	397.5000
4	Robbyn Claire Wankel	0	350	350	310.0000

4) Show the most frequent appointment type(online or offline) with respect to age and gender. **

```

SELECT *
FROM    myfirstdb.doctor dr
        LEFT JOIN myfirstdb.appointment ap
              ON dr.id = ap.doctor_id;

SELECT (SELECT Concat(ap.meeting_type, ' ',
                    '- ', 'no of patients ', Count(ap.meeting_type))
        FROM    myfirstdb.patient pa
                INNER JOIN myfirstdb.appointment ap
                      ON pa.id = ap.patient_id
        WHERE   pa.gender = 'M'
                AND Timestampdiff(year, pa.date_of_birth, Now()) < 20
        GROUP BY ap.meeting_type
        ORDER BY Count(ap.meeting_type) DESC
        LIMIT 1) AS 'most_frequent_type_for_male_under_20_years',
(SELECT Concat(ap.meeting_type, ' ',
                '- ', 'no of patients ', Count(ap.meeting_type))
        FROM    myfirstdb.patient pa
                INNER JOIN myfirstdb.appointment ap
                      ON pa.id = ap.patient_id
        WHERE   pa.gender = 'F'
                AND Timestampdiff(year, pa.date_of_birth, Now()) < 20
        GROUP BY ap.meeting_type
        ORDER BY Count(ap.meeting_type) DESC
        LIMIT 1) AS 'most_frequent_type_for_female_under_20_years',
(SELECT Concat(ap.meeting_type, ' ',

```

```

        '- ', 'no of patients ', Count(ap.meeting_type))
FROM    myfirstdb.patient pa
        INNER JOIN myfirstdb.appointment ap
            ON pa.id = ap.patient_id
WHERE   pa.gender = 'M'
        AND Timestampdiff(year, pa.date_of_birth, Now()) > 20
        AND Timestampdiff(year, pa.date_of_birth, Now()) < 40
GROUP   BY ap.meeting_type
ORDER   BY Count(ap.meeting_type) DESC
LIMIT   1) AS 'most_frequent_type_for_male_between 20 - 40 years',
(SELECT Concat(ap.meeting_type, ' ',
        '- ', 'no of patients ', Count(ap.meeting_type))
FROM    myfirstdb.patient pa
        INNER JOIN myfirstdb.appointment ap
            ON pa.id = ap.patient_id
WHERE   pa.gender = 'F'
        AND Timestampdiff(year, pa.date_of_birth, Now()) > 20
        AND Timestampdiff(year, pa.date_of_birth, Now()) < 40
GROUP   BY ap.meeting_type
ORDER   BY Count(ap.meeting_type) DESC
LIMIT   1) AS 'most_frequent_type_for_female_between 20 - 40
years',
(SELECT Concat(ap.meeting_type, ' ',
        '- ', 'no of patients ', Count(ap.meeting_type))
FROM    myfirstdb.patient pa
        INNER JOIN myfirstdb.appointment ap
            ON pa.id = ap.patient_id
WHERE   pa.gender = 'M'
        AND Timestampdiff(year, pa.date_of_birth, Now()) > 40
GROUP   BY ap.meeting_type
ORDER   BY Count(ap.meeting_type) DESC
LIMIT   1) AS 'most_frequent_type_for_male_above 40 years',
(SELECT Concat(ap.meeting_type, ' ',
        '- ', 'no of patients ', Count(ap.meeting_type))
FROM    myfirstdb.patient pa
        INNER JOIN myfirstdb.appointment ap
            ON pa.id = ap.patient_id
WHERE   pa.gender = 'F'
        AND Timestampdiff(year, pa.date_of_birth, Now()) > 40
GROUP   BY ap.meeting_type
ORDER   BY Count(ap.meeting_type) DESC
LIMIT   1) AS 'most_frequent_type_for_female_above 40 years';

```

most_frequent_type_for_male_under 20 years	most_frequent_type_for_female_under 20 years	most_frequent_type_for_male_between 20 - 40 years
online - no of patients 48	online - no of patients 40	online - no of patients 49
most_frequent_type_for_female_between 20 - 40 years	most_frequent_type_for_male_above 40 years	most_frequent_type_for_female_above 40 years
online - no of patients 56	online - no of patients 28	offline - no of patients 30

5) Patient can choose on which day he wants to have an appointment by seeing the total number of appointments for a particular doctor on a specific day. Show the number of appointments for a doctor on a specific day and probability of patients having appointments on that specific day (patients having appointment on a particular day with a specific doctor / number of patients having appointment with that specific doctor). **

```

SELECT cday.x_id AS 'doctor id',
       cday.x_name AS 'doctor
name',
       cday.cnt_per_day AS
       'no of appointments for a doctor on that particular day',
       cday.x_day AS 'day',
       ( ( cday.cnt_per_day ) / ( cdoc.ct_per_doctor ) ) AS
       'probably of patients having appointment on that particular day'
FROM   (SELECT Count(x.day) AS 'cnt_per_day',
              x.doctor_id AS x_id,
              x.doctor_name AS x_name,
              x.day AS x_day
        FROM   (SELECT Dayname(ap.appointment_time) AS day,
                      dr.id AS
'doctor_id',
                      Concat(dr.first_name, ' ', dr.last_name) AS
'doctor_name'
                FROM   myfirstdb.appointment ap
                INNER JOIN myfirstdb.doctor dr
                        ON dr.id = ap.doctor_id
                ORDER BY dr.id) AS x
        GROUP BY x.doctor_id,
                 x.day
        ORDER BY x.doctor_id) AS cday
INNER JOIN (SELECT Count(y.day) AS 'ct_per_doctor',
                  y.doctor_id AS y_id,
                  y.doctor_name AS y_name,
                  y.day AS y_day
            FROM   (SELECT Dayname(ap.appointment_time)
AS
                        day,
                        dr.id
AS
                        'doctor_id',
                        Concat(dr.first_name, ' ',
dr.last_name) AS
                        'doctor_name'
                FROM   myfirstdb.appointment ap
                INNER JOIN myfirstdb.doctor dr
                        ON dr.id = ap.doctor_id
                ORDER BY dr.id) AS y
            GROUP BY y.doctor_id
            ORDER BY y.doctor_id) AS cdoc
ON cday.x_id = cdoc.y_id;

```


doctor id	doctor name	no of appointments for a doctor on that particular day	day	probability of patients having appointment on that particular day
1	Mable Wink	1	Monday	0.2000
1	Mable Wink	1	Saturday	0.2000
1	Mable Wink	1	Sunday	0.2000
1	Mable Wink	1	Tuesday	0.2000
1	Mable Wink	1	Wednesday	0.2000

6) Show the latest appointment given by a particular doctor.

```

SELECT dr.id,
       Concat(dr.first_name, ' ', dr.middle_name, ' ', dr.last_name) AS
       'doctor name',
       ap.patient_id,
       Concat(p.first_name, ' ', p.middle_name, ' ', p.last_name) AS
       'patient name',
       ap.appointment_time
FROM   myfirstdb.doctor dr
       INNER JOIN myfirstdb.appointment ap
                ON ap.doctor_id = dr.id
       INNER JOIN myfirstdb.patient p
                ON ap.patient_id = p.id
WHERE  dr.id = 1
       AND status = 'accepted'
ORDER BY ap.appointment_time DESC
LIMIT 1;

```

id	doctor name	patient_id	patient name	appointment_time
1	Mable La Batie Wink	138	Grover Guillermo Ferney	2021-09-06 11:38:13

7) Get the total revenue generated for a particular year.

```

SELECT Year(b.created_at) AS year,
       Sum(Ifnull(b.medicine_price, 0)) AS 'total revenue (USD)'
FROM   myfirstdb.bill b
GROUP BY Year(b.created_at);

```

	year	total revenue (USD)
▶	2020	104864
	2021	228332

8) Get the average consult price, max consult price, min consultation price for each doctor. *

```

SELECT dr.id,

```

```

Concat(dr.first_name, ' ', COALESCE(dr.middle_name, ' '), ' ',
dr.last_name) AS
doctor_name,
ef.NAME
AS expert_field_name,
Avg(pr.consult_price)
AS average_consult_price,
Min(pr.consult_price)
AS min_consult_price,
Max(pr.consult_price)
AS max_consult_price
FROM myfirstdb.doctor dr
INNER JOIN myfirstdb.expert_field ef
ON dr.expert_field_id = ef.id
INNER JOIN myfirstdb.appointment ap
ON ap.doctor_id = dr.id
INNER JOIN myfirstdb.prescription pr
ON pr.appointment_id = ap.id
GROUP BY dr.id
ORDER BY dr.id;

```

id	doctor_name	expert_field_name	average_consult_price	min_consult_price	max_consult_price
1	Mable La Batie Wink	Respiratory	210.0000	50	500
2	Thorny Dalgarnowch Tregien	Urology	288.8889	50	450
3	Adolpho Astill Lamble	Infectious disease	250.0000	100	500
4	Karla Wareham Horbath	Infectious disease	290.0000	100	500
5	Alisander Innis Tench	Urology	233.3333	50	500

9) Create Table Queries

- Patient

```
CREATE TABLE project.patient
(
    id                BIGINT NOT NULL auto_increment,
    ssn               VARCHAR(50) NOT NULL,
    first_name        VARCHAR(50) NOT NULL,
    middle_name        VARCHAR(50),
    last_name          VARCHAR(50) NOT NULL,
    email              VARCHAR(50) NOT NULL,
    phone_number       VARCHAR(20),
    state              VARCHAR(30) NOT NULL,
    zipcode            INT NOT NULL,
    gender              ENUM('M', 'F'),
    date_of_birth       DATETIME NOT NULL,
    username            VARCHAR(100) NOT NULL,
    password            VARCHAR(100) NOT NULL,
    last_login          DATETIME,
    registered_at       DATETIME NOT NULL DEFAULT Now(),
    updated_at          DATETIME NOT NULL DEFAULT Now(),
    PRIMARY KEY (id)
);
ALTER TABLE project.patient
ADD UNIQUE (ssn),
ADD UNIQUE (email);
```

- Hospital

```
CREATE TABLE project.hospital
(
    id                BIGINT NOT NULL auto_increment,
    name               VARCHAR(250) NOT NULL,
    phone_number       VARCHAR(30) NOT NULL,
    address             TEXT NOT NULL,
    state              VARCHAR(30) NOT NULL,
    zipcode            INT NOT NULL,
    PRIMARY KEY (id)
);
```

- **Expert Field**

```
CREATE TABLE project.expert_field
(
    id          BIGINT NOT NULL auto_increment,
    name        VARCHAR(150) NOT NULL,
    description TEXT NOT NULL,
    PRIMARY KEY (id),
    UNIQUE KEY name (name)
);
```

- **Doctor**

```
CREATE TABLE project.doctor
(
    id          BIGINT NOT NULL auto_increment,
    first_name  VARCHAR(50) NOT NULL,
    middle_name VARCHAR(50) NULL,
    last_name   VARCHAR(50) NOT NULL,
    email       VARCHAR(50) NOT NULL,
    hospital_id BIGINT NOT NULL,
    expert_field_id BIGINT NOT NULL,
    PRIMARY KEY (id),
    KEY fk_hospital_id (hospital_id),
    KEY fk_expert_field(expert_field_id),
    UNIQUE KEY email (email),
    CONSTRAINT fk_hospital_id FOREIGN KEY (hospital_id) REFERENCES
project.hospital(id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_expert_field FOREIGN KEY (expert_field_id) REFERENCES
project.expert_field(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

- **Appointment**

```
CREATE TABLE project.appointment
(
    id          BIGINT NOT NULL auto_increment,
    patient_id  BIGINT NOT NULL,
    doctor_id   BIGINT NOT NULL,
    appointment_time DATETIME NOT NULL,
    meeting_type ENUM('online', 'offline') NOT NULL,
    meeting_duration BIGINT NOT NULL,
    status      ENUM('new', 'accepted', 'rejected') DEFAULT 'new',
    created_at  DATETIME NOT NULL DEFAULT Now(),
    updated_at  DATETIME,
```

```
rating_score      INT,  
    CHECK (rating_score <= 5),  
    message        LONGTEXT,  
PRIMARY KEY (id),  
KEY fk_patient_id(patient_id),  
KEY fk_doctor_id(doctor_id),  
CONSTRAINT fk_patient_id FOREIGN KEY (patient_id) REFERENCES  
    project.patient(id) ON DELETE CASCADE ON UPDATE CASCADE,  
CONSTRAINT fk_doctor_id FOREIGN KEY (doctor_id) REFERENCES  
    project.doctor(id) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- **Prescription**

```
CREATE TABLE project.prescription  
(  
    id                BIGINT NOT NULL auto_increment,  
    appointment_id    BIGINT NOT NULL,  
    message           LONGTEXT NOT NULL,  
    treatment_duration INT NOT NULL,  
    consult_price      INT,  
PRIMARY KEY (id),  
UNIQUE KEY appointment_id (appointment_id),  
CONSTRAINT fk_appointment_id FOREIGN KEY (appointment_id) REFERENCES  
    project.appointment(id) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

- **Workshift**

```
CREATE TABLE project.workshift  
(  
    id                BIGINT NOT NULL auto_increment,  
    day              VARCHAR(10) NOT NULL,  
    start_time       TIME NOT NULL,  
    end_time         TIME NOT NULL,  
PRIMARY KEY (id)  
);
```

- **Work**

```
CREATE TABLE project.work
(
    id          BIGINT NOT NULL auto_increment,
    doctor_id   BIGINT NOT NULL,
    workshift_id BIGINT NOT NULL,
    PRIMARY KEY(id),
    KEY doctor_id (doctor_id),
    KEY workshift_id (workshift_id),
    CONSTRAINT fk_workdoctor_id FOREIGN KEY (doctor_id) REFERENCES
project.doctor(id) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_workshift_id FOREIGN KEY (workshift_id) REFERENCES
project.workshift(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

- **Medicine**

```
CREATE TABLE project.medicine
(
    id          BIGINT NOT NULL auto_increment,
    name        TEXT NOT NULL,
    price       INT NOT NULL,
    description TEXT NULL,
    PRIMARY KEY(id)
);
```

- **Bill**

```
CREATE TABLE project.bill
(
    id          BIGINT NOT NULL auto_increment,
    created_at  DATETIME NOT NULL DEFAULT Now(),
    medicine_price BIGINT NOT NULL,
    patient_id  BIGINT NOT NULL,
    PRIMARY KEY(id),
    KEY fk_patient_id(patient_id),
    CONSTRAINT fk_billpatient_id FOREIGN KEY (patient_id) REFERENCES
project.patient(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

- **Bill_item**

```
CREATE TABLE project.bill_item
(
    id          BIGINT NOT NULL auto_increment,
    bill_id     BIGINT NOT NULL,
    medicine_id BIGINT NOT NULL,
    quantity    FLOAT NOT NULL,
    PRIMARY KEY(id),
    KEY bill_id (bill_id),
    KEY medicine_id (medicine_id),
    CONSTRAINT fk_bill_id FOREIGN KEY (bill_id) REFERENCES
project.bill(id) ON
    DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_medicine_id FOREIGN KEY (medicine_id) REFERENCES
project.medicine(id) ON DELETE CASCADE ON UPDATE CASCADE
);
```

10) Conclusion

Summary

During this project, we designed a doctor appointment system. Even if our model is not complete yet, we think that it is some basics of a real online appointment system. We also learned how to reduce the scope of a project to make it more interesting to present. Besides, we learned how to work in a small group via many online platforms such as google docs, slides, meet draw.io, mockaroo and zoom. We are sure that this knowledge will be useful in the future and we are glad to have this opportunity. Besides, even if our reports are not realistic, if we had real data we would be able to give interesting reports in the medical sector.

Future Work

- Develop the application
- Add more detail to the table structure
- Add more functionalities like lab tests
- Implement the application
- Develop the application
- Add more detail to the table structure
- Add a post and message system to make it more collaborative.
- The remainders can also be added so that the user can keep a remainder in order to get the medicine regularly.

G2 Tindoc Project

- We also include the online recorded videos for common problems faced by the citizens during this covid time.
- The scope can be increased to the inclusion of lab tests and health checkups.
- Offers and packages for the health checkups and lab tests can also be included.

11) Work Distribution and Contribution

RDB Mini Project: Check List and Team Member Contribution Form

Team Number:								
Project Name:								
	Criteria	WEIGHT	DESCRIPTION	EXCELLENT (10 PTS) ALL	SATISFACTORY (7 PTS) MOST	BORDERLINE (4 PTS) SOME	INSUFFICIENT (1 PT) NONE	SCORE * WEIGHT
INTERESTING Project Idea								
1.	Project Idea: <i>Project Topic, Description & Requirements</i>	10%	<p>The project idea shows the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> interesting, challenging, creative and reflecting a real-world scenario. <input type="checkbox"/> The project description is clearly identified. <input type="checkbox"/> Application, data requirements, data constraints and business rules are clearly defined, realistic and well-research. <input type="checkbox"/> Important usage scenarios and queries are properly defined: <ul style="list-style-type: none"> <input type="checkbox"/> Important insert, update, delete operations and transactions (minimum 10 operations each); <input type="checkbox"/> Identifying important data inquiries and reports (minimum 20 inquiries or reports); 					
DB DESIGN and DEVELOPMENT								

2.	Conceptual and Logical Design: <i>Appropriateness and accuracy of Design</i>	35%	<p>The conceptual design (ER model) has the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> 100% accuracy <input type="checkbox"/> Capture all data requirements explained in the proposal <input type="checkbox"/> Easy to understand with a nice diagram layout <input type="checkbox"/> Follow proper naming convention for the entity names, attribute names, relationship names. <p>The logical design (relational schema, data dictionary and SQL scripts for table creation and queries) has the following:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Schema and Data dictionary is accurate (corresponding to the conceptual design) <input type="checkbox"/> Properly define data types for all attributes (with careful domain analysis) <input type="checkbox"/> Properly define the PKs for all tables <input type="checkbox"/> Properly define the FKs for all tables <input type="checkbox"/> Properly define all integrity constraints, and business rules, as defined in the Project Proposal <p>The design is tested with practical, sample data.</p>					
3.	Implementation & Demonstration (Individually evaluated)	35%	<ul style="list-style-type: none"> <input type="checkbox"/> The implemented operations, transactions and queries completely fulfils the project requirements (set at the proposal stage), is practical and useful in a real-world scenario. <input type="checkbox"/> The implementation is well demonstrated. It is interesting and effective in conveying ideas. 					

G2 Tindoc Project

			<input type="checkbox"/> Selected usage scenarios and queries are correctly written in SQL statements and are tested.					
PRESENTATION								
4.	Presentation & Communication (Individually evaluated) <i>The oral presentation does not exceed the time allotment. It is tight, focused, and clearly explains the project.</i>	20%	<input type="checkbox"/> Oral presentation keeps to the time limit and is focused. <input type="checkbox"/> Main ideas are expressed clearly and convincingly. <input type="checkbox"/> Q&A session is well managed and questions can be properly addressed.					

Work Distribution and Contribution

Task	Member#1 [Saratoon]	Member#2 [Pasit]	Member#3 [Harold]	Member#4 [Vineela]	Member#5 [Noppawan]
1. Project Proposal: <i>Project Topic, Description & Requirements</i>	Propose the idea and brainstorm (Original idea: Prison Database)	Propose some ideas. And brainstorming about project scope and features in the application.	Participated in the team brainstorming.	Proposed an idea (community management system) and brainstorming about purpose, features and usage of the application.	Proposed an idea (about hospital), brainstorm in the team.
2. Conceptual and Logical Design	Try to draw my own version of ER-diagram and discuss with others	Draw an ER-diagram and discuss it with everyone.	Participated in the conception.	Draw the workflow, E-r diagram and discuss it with everyone.	Participated in detail discussion and drawn some diagrams
3. Final Presentation	A part of conceptual design, Database population and Data dictionary	Put everything together and design a use case diagram to make the system more understandable.	Bring my contribution	Part of conceptual, logical, sample data creation, data dictionary and description of all parts in the content.	Share some ideas and review and revise some parts that were not correct.

For each data operation/query, specify its type regarding the syntactic dimension and semantic dimension as well as the members who implement it.

Data Operation/Query	Query Type : Syntactic Dimension		Query Type : Semantic Dimension		Implementer
	I/U/D?	BR/JOIN/GROUP?	OTP?	BP/BT/CA/PA/REC	
1. Can see the appointment and prescription history of a particular patient.		JOIN	OTP		Noppawan Nurnuansuwan
2. Can see average days that each doctor accepts an appointment.*		JOIN, GROUP		REC	Noppawan Nurnuansuwan
3. Number of patients living in each state in the selected hospital.*		JOIN, GROUP		BT	Noppawan Nurnuansuwan

G2 Tindoc Project

4. Get the most common disease in the state (Max number of cases in a specific field in each state).**		JOIN, GROUP		BT	Nopphawan Nurnuansuwan
5. Show all patients who didn't have any appointments and bills.		JOIN		BP	Nopphawan Nurnuansuwan
6. Show the patient information with their current age, number of appointments that they have made, number of bills that they have created, total appointments and bills, and number of medicine that they have ordered.**		JOIN, GROUP		BP	Nopphawan Nurnuansuwan
7. Most needed field of expertise filter by age and gender**		JOIN/GROUP		BT	Saratoon Khantasima
8. How many patients per month**		JOIN/GROUP		CA	Saratoon Khantasima
9. See bill details		BR	OTP		Saratoon Khantasima
10. See the patient who is often sick*		GROUP		CA	Saratoon Khantasima
11. Number of case for each doctor*		JOIN/GROUP		BT	Saratoon Khantasima
12. Check the rating for doctor*		JOIN/GROUP		REC	Saratoon Khantasima
13. Patients can check the appointment time.		JOIN	OTP		Pasit Tiwawongrut
14. Number of hospitals which treat the medical problem near to the location of the patient. *		JOIN	OTP		Pasit Tiwawongrut
15. Find which hospitals have more doctors than average. **		JOIN/GROUP		BP	Pasit Tiwawongrut

G2 Tindoc Project

16. Doctors can check their work hours.		JOIN	OTP		Pasit Tiwawongrut
17. Doctors can see the number of patients that make an appointment between particular times. **		JOIN	OTP		Pasit Tiwawongrut
18. Hospitals can see how many doctors for each work shift. *		JOIN/GROUP		BP	Pasit Tiwawongrut
19. Total cases for each expert field on the specific date. **		JOIN/GROUP		BP/BT	Pasit Tiwawongrut
20. Prescription from each doctor		JOIN	OPT		Harold Popluhar
21. List of the drugs a patient can order		JOIN	OPT		Harold Popluhar
22. Typical disease by quarter		JOIN/GROUP		BT	Harold Popluhar
23. The average treatment duration per medical problem		JOIN/GROUP		BP	Harold Popluhar
24. Fast report of previous disease of a patient		JOIN		CA	Harold Popluhar
25. Good distribution of doctors per medical sector		JOIN,GROUP		REC	Harold Popluhar
26. Show doctor information with their name, email,expert field, hospital which they belong to, no of appointments, no of working hours, average rating for a doctor, maximum rating, min rating of a doctor, no of patients who visited the doctor. **		JOIN,GROUP		BP	Vineela Mukkamala

G2 Tindoc Project

27. Get the average duration spent by each doctor on patients and display his details along with the expert field.*		JOIN,GROUP		BT	Vineela Mukkamala
28. Total bill of the patient made till date. *		JOIN,GROUP		BP	Vineela Mukkamala
29. Show the most frequent appointment type(online or offline) with respect to age and gender. **		JOIN,GROUP		BT	Vineela Mukkamala
30. Patients can choose on which day he wants to have an appointment by seeing the total number of appointments for a particular doctor on a specific day. Show the number of appointments for a doctor on a specific day and probability of patients having appointments on that specific day(patients having appointment on a particular day with a specific doctor / number of patients having appointment with that specific doctor). **		JOIN,GROUP		REC	Vineela Mukkamala
31. Patients can see the latest appointment given by a particular doctor		JOIN,GROUP	OPT		Vineela Mukkamala
32. Show the total revenue generated by a particular year.		GROUP		BP,BT	Vineela Mukkamala
33. Patients can see the maximum consultation price, minimum consultation price, average consultation price of all doctors and can choose the doctor accordingly.*		JOIN,GROUP		REC	Vineela Mukkamala

G2 Tindoc Project

Note: * - medium level, ** - complex level queries, others- easy queries

Remark:

Types of SQL Statement: Syntactic Dimension

- (I) Insert, (U) Update, (D) Delete
- Retrieval
 - (BR) Basic Retrieval
 - (JOIN) Join Query / Nested or Subquery
 - (GROUP) Aggregate Query

Types of SQL Statement: Semantics Dimension

- (OPT) Basic Operation/Transaction Support
- Data Intelligence and Customer Insight Support
 - (BP) Business Performance Analysis
 - (BT) Basic Trend Analysis
 - (CA) Customer Analysis
 - (PA) Product Analysis
 - (REC) Suggestion or Recommendation of Products / Services to Users using history data