PES UNIVERSITY

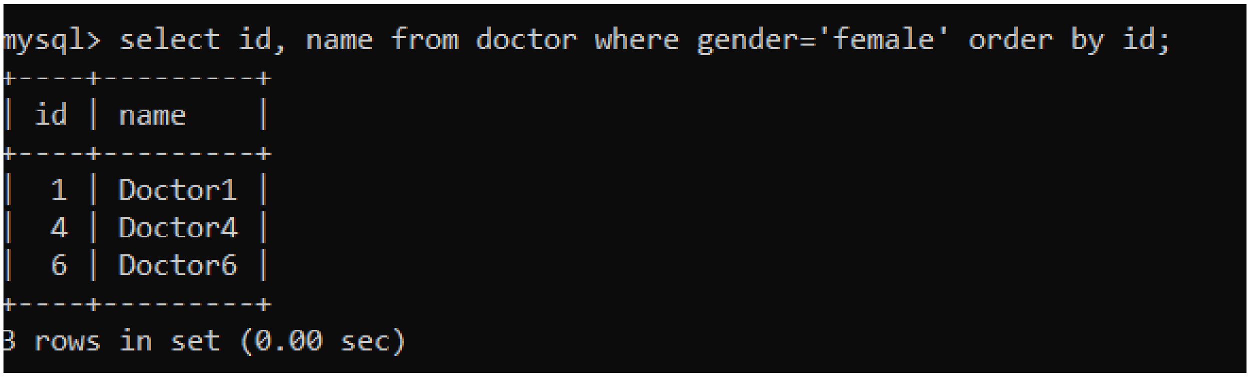
SEMESTER – 5
Assignment - 3

NAME	SRN
MEELA DEEPTI	PES2UG19CS227
M SATVIKA	PES2UG19CS207
LAKSHMI NARAYAN P	PES2UG19CS200

Simple queries

Query No.1 - Display name of female doctors and their ids

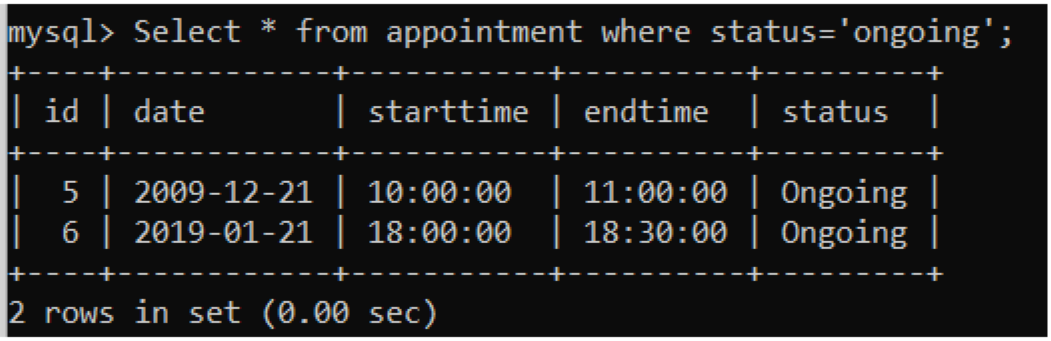
Select id,name from doctor where gender='female' order by id;



```
mysql> select id, name from doctor where gender='female' order by id;
+----+---------+
| id | name    |
+----+---------+
|  1 | Doctor1 |
|  4 | Doctor4 |
|  6 | Doctor6 |
+----+---------+
3 rows in set (0.00 sec)
```

Query No.2 - Display all the appointments whose status is ongoing

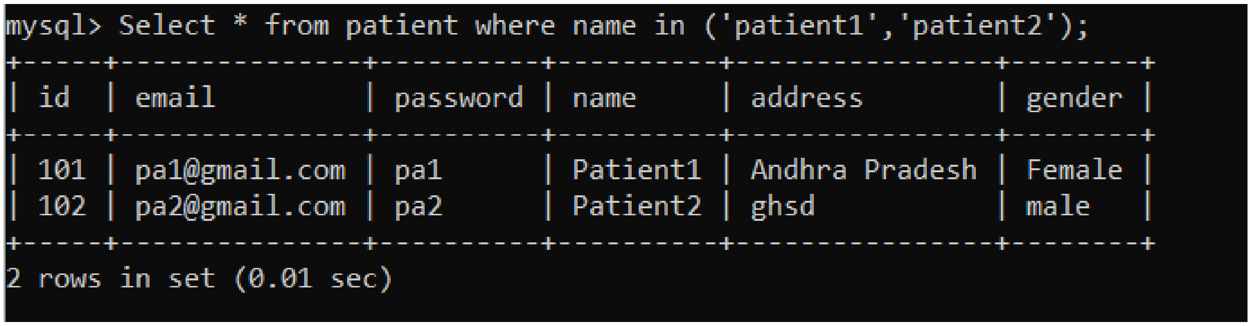
Select * from appointment where status=’ongoing’;



```
mysql> Select * from appointment where status='ongoing';
+----+------------+-----------+----------+---------+
| id | date       | starttime | endtime  | status  |
+----+------------+-----------+----------+---------+
|  5 | 2009-12-21 | 10:00:00  | 11:00:00 | Ongoing |
|  6 | 2019-01-21 | 18:00:00  | 18:30:00 | Ongoing |
+----+------------+-----------+----------+---------+
2 rows in set (0.00 sec)
```

Query No.3 - Display all attributes of patients if name of the patient is present in the tuple

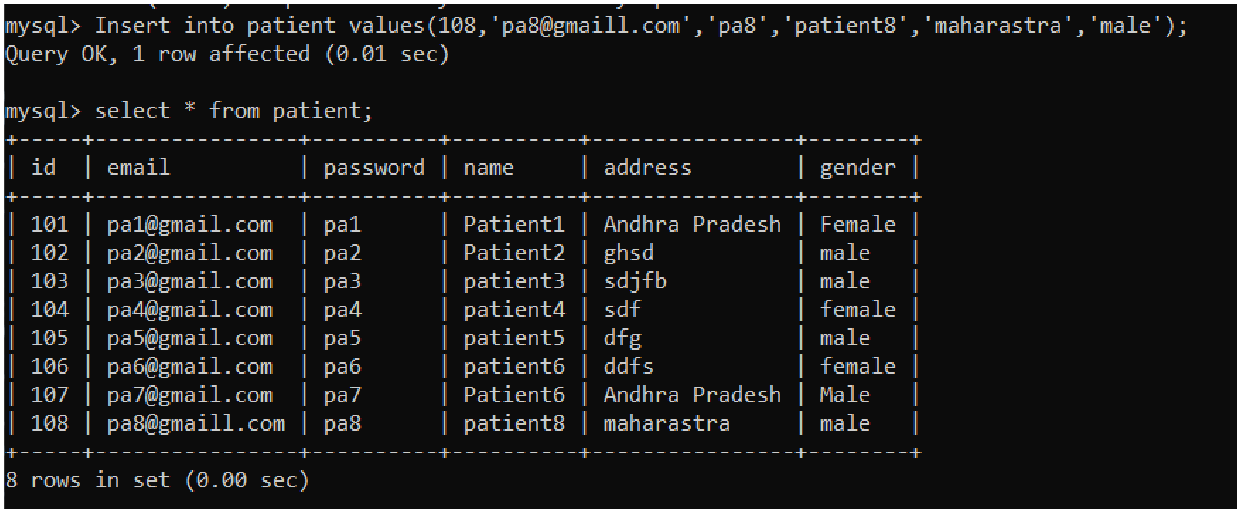
Select * from patient where name in (‘patient1’,’patient2’);



```
mysql> Select * from patient where name in ('patient1','patient2');
+-----+---------------+----------+----------+----------------+--------+
| id  | email         | password | name     | address        | gender |
+-----+---------------+----------+----------+----------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Andhra Pradesh | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | ghsd           | male   |
+-----+---------------+----------+----------+----------------+--------+
2 rows in set (0.01 sec)
```

Query No.4 - Insert a new patient into the Patients table and display the changes.

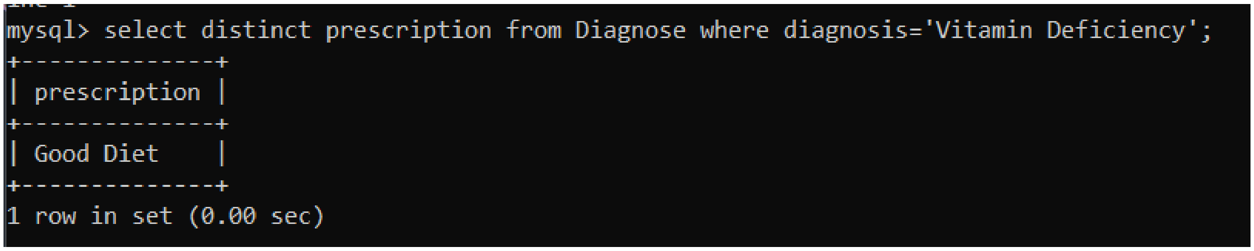
Insert into patient values(108,’pa8@gmaill.com’,’pa8’,’patient8’,’maharastra’,’male’);



```
mysql> Insert into patient values(108,'pa8@gmaill.com','pa8','patient8','maharastra','male');
Query OK, 1 row affected (0.01 sec)

mysql> select * from patient;
+-----+----------------+----------+----------+----------------+--------+
| id  | email          | password | name     | address        | gender |
+-----+----------------+----------+----------+----------------+--------+
| 101 | pa1@gmail.com  | pa1      | Patient1 | Andhra Pradesh | Female |
| 102 | pa2@gmail.com  | pa2      | Patient2 | ghsd           | male   |
| 103 | pa3@gmail.com  | pa3      | patient3 | sdjfb          | male   |
| 104 | pa4@gmail.com  | pa4      | patient4 | sdf            | female |
| 105 | pa5@gmail.com  | pa5      | patient5 | dfg            | male   |
| 106 | pa6@gmail.com  | pa6      | patient6 | ddfs           | female |
| 107 | pa7@gmail.com  | pa7      | Patient6 | Andhra Pradesh | Male   |
| 108 | pa8@gmaill.com | pa8      | patient8 | maharastra     | male   |
+-----+----------------+----------+----------+----------------+--------+
8 rows in set (0.00 sec)
```

Query No.5 - Display the prescription of the doctor for vitamin deficiency.

select distinct prescription from Diagnose where diagnosis='Vitamin Deficiency';



```
mysql> select distinct prescription from Diagnose where diagnosis='Vitamin Deficiency';
+--------------+
| prescription |
+--------------+
| Good Diet    |
+--------------+
1 row in set (0.00 sec)
```

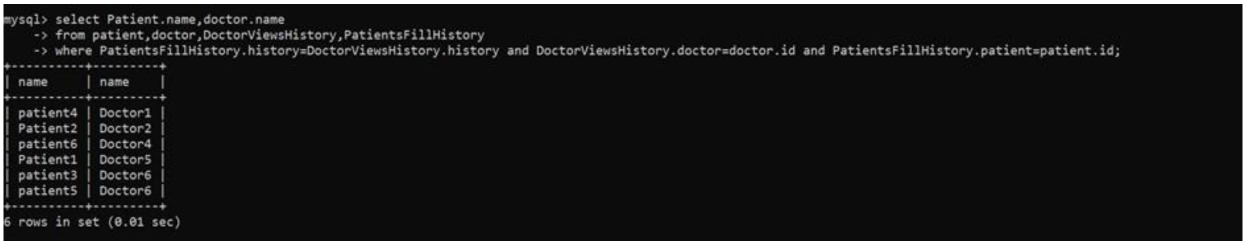
Complex queries

Query No.1- Display the details of the patient and respective doctor assigned to them.

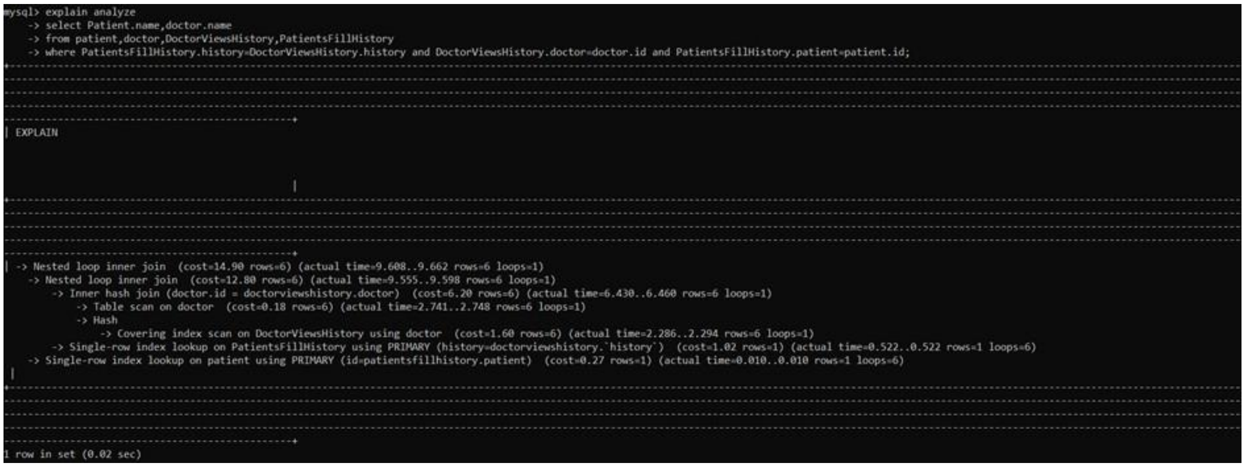
select Patient.name,doctor.name

from patient,doctor,DoctorViewsHistory,PatientsFillHistory

where PatientsFillHistory.history=DoctorViewsHistory.history and DoctorViewsHistory.doctor=doctor.id and PatientsFillHistory.patient=patient.id;



```
mysql> select Patient.name,doctor.name
    -> from patient,doctor,DoctorViewsHistory,PatientsFillHistory
    -> where PatientsFillHistory.history=DoctorViewsHistory.history and DoctorViewsHistory.doctor=doctor.id and PatientsFillHistory.patient=patient.id;
+----------+---------+
| name     | name    |
+----------+---------+
| patient4 | Doctor1 |
| Patient2 | Doctor2 |
| patient6 | Doctor4 |
| Patient1 | Doctor5 |
| patient3 | Doctor6 |
| patient5 | Doctor6 |
+----------+---------+
6 rows in set (0.01 sec)
```



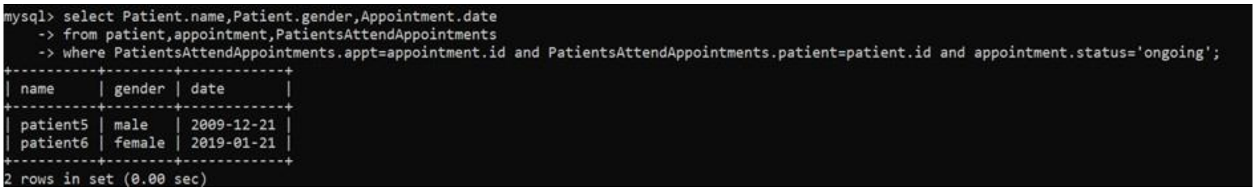
```
mysql> explain analyze
    -> select Patient.name,doctor.name
    -> from patient,doctor,DoctorViewsHistory,PatientsFillHistory
    -> where PatientsFillHistory.history=DoctorViewsHistory.history and DoctorViewsHistory.doctor=doctor.id and PatientsFillHistory.patient=patient.id;
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------+
| EXPLAIN



                                                    |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------+
| -> Nested loop inner join  (cost=14.90 rows=6) (actual time=9.608..9.662 rows=6 loops=1)
    -> Nested loop inner join  (cost=12.80 rows=6) (actual time=9.555..9.598 rows=6 loops=1)
        -> Inner hash join (doctor.id = doctorviewshistory.doctor)  (cost=6.20 rows=6) (actual time=6.430..6.460 rows=6 loops=1)
            -> Table scan on doctor  (cost=0.18 rows=6) (actual time=2.741..2.748 rows=6 loops=1)
            -> Hash
                -> Covering index scan on DoctorViewsHistory using doctor  (cost=1.60 rows=6) (actual time=2.286..2.294 rows=6 loops=1)
        -> Single-row index lookup on PatientsFillHistory using PRIMARY (history=doctorviewshistory.`history`)  (cost=1.02 rows=1) (actual time=0.522..0.522 rows=1 loops=6)
    -> Single-row index lookup on patient using PRIMARY (id=patientsfillhistory.patient)  (cost=0.27 rows=1) (actual time=0.010..0.010 rows=1 loops=6)
 |
+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------+
1 row in set (0.02 sec)
```

Query No.2 - Display name, gender, appointment date of patients whose appointment status is ongoing.

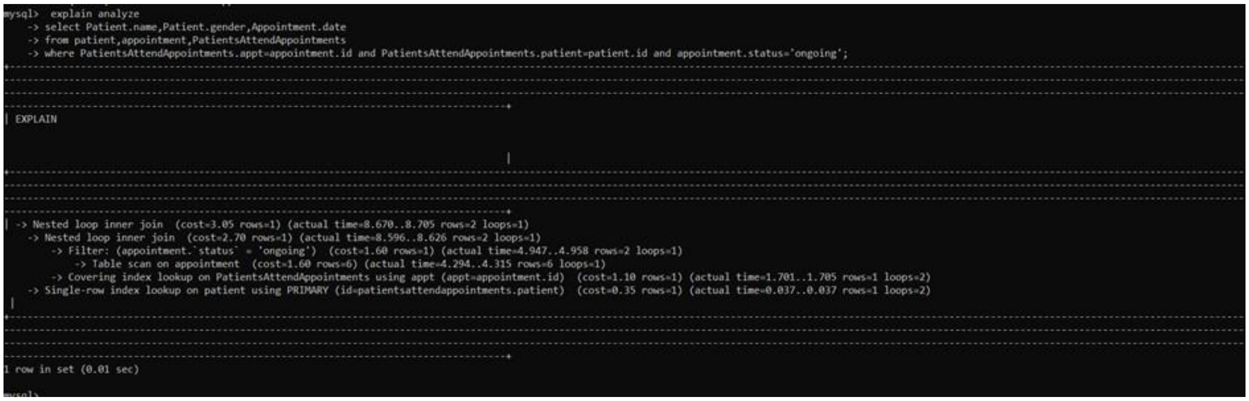
select Patient.name,Patient.gender,Appointment.date

from patient,appointment,PatientsAttendAppointments

where PatientsAttendAppointments.appt=appointment.id and PatientsAttendAppointments.patient=patient.id and appointment.status='ongoing';



```
mysql> select Patient.name,Patient.gender,Appointment.date
    -> from patient,appointment,PatientsAttendAppointments
    -> where PatientsAttendAppointments.appt=appointment.id and PatientsAttendAppointments.patient=patient.id and appointment.status='ongoing';
+----------+--------+------------+
| name     | gender | date       |
+----------+--------+------------+
| patient5 | male   | 2009-12-21 |
| patient6 | female | 2019-01-21 |
+----------+--------+------------+
2 rows in set (0.00 sec)
```



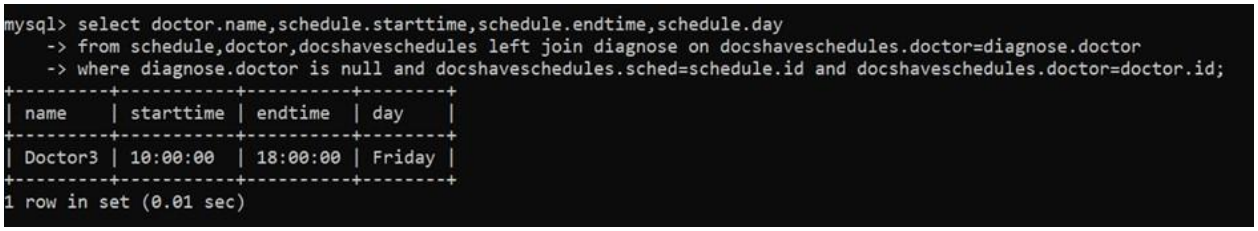
```
mysql>  explain analyze
    -> select Patient.name,Patient.gender,Appointment.date
    -> from patient,appointment,PatientsAttendAppointments
    -> where PatientsAttendAppointments.appt=appointment.id and PatientsAttendAppointments.patient=patient.id and appointment.status='ongoing';
+------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------+
| EXPLAIN


                                                                                 |
+------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------+
| -> Nested loop inner join  (cost=3.05 rows=1) (actual time=8.670..8.705 rows=2 loops=1)
    -> Nested loop inner join  (cost=2.70 rows=1) (actual time=8.596..8.626 rows=2 loops=1)
        -> Filter: (appointment.`status` = 'ongoing')  (cost=1.60 rows=1) (actual time=4.947..4.958 rows=2 loops=1)
            -> Table scan on appointment  (cost=1.60 rows=6) (actual time=4.294..4.315 rows=6 loops=1)
        -> Covering index lookup on PatientsAttendAppointments using appt (appt=appointment.id)  (cost=1.10 rows=1) (actual time=1.701..1.705 rows=1 loops=2)
    -> Single-row index lookup on patient using PRIMARY (id=patientsattendappointments.patient)  (cost=0.35 rows=1) (actual time=0.037..0.037 rows=1 loops=2)
 |
+------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------+
1 row in set (0.01 sec)

mysql>
```

Query No.3- Display name of doctor who have no patients assigned to them

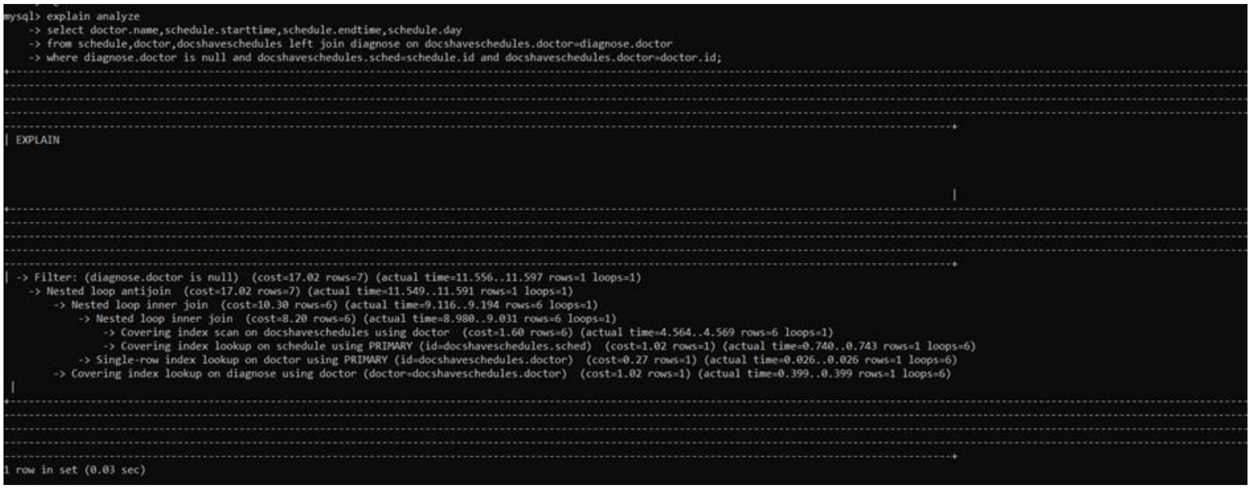
select doctor.name,schedule.starttime,schedule.endtime,schedule.day

from schedule,doctor,docshaveschedules left join diagnose on docshaveschedules.doctor=diagnose.doctor

where diagnose.doctor is null and docshaveschedules.sched=schedule.id and docshaveschedules.doctor=doctor.id;



```
mysql> select doctor.name,schedule.starttime,schedule.endtime,schedule.day
    -> from schedule,doctor,docshaveschedules left join diagnose on docshaveschedules.doctor=diagnose.doctor
    -> where diagnose.doctor is null and docshaveschedules.sched=schedule.id and docshaveschedules.doctor=doctor.id;
+---------+-----------+----------+--------+
| name    | starttime | endtime  | day    |
+---------+-----------+----------+--------+
| Doctor3 | 10:00:00  | 18:00:00 | Friday |
+---------+-----------+----------+--------+
1 row in set (0.01 sec)
```



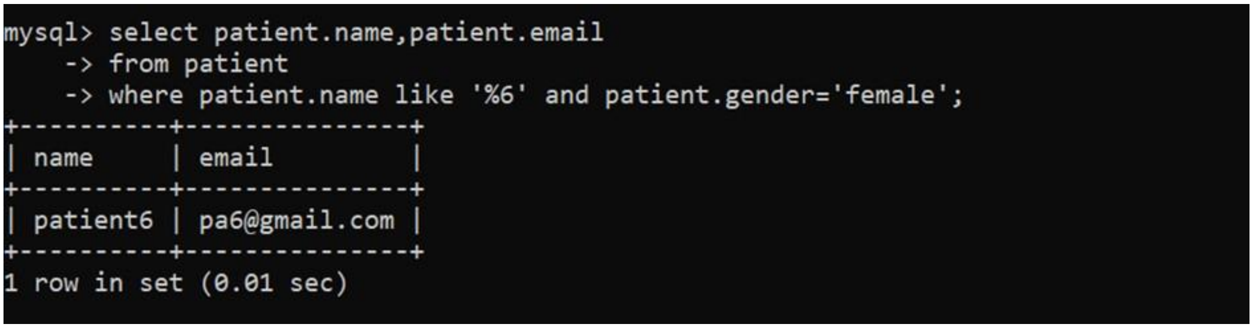
```
mysql> explain analyze
    -> select doctor.name,schedule.starttime,schedule.endtime,schedule.day
    -> from schedule,doctor,docshaveschedules left join diagnose on docshaveschedules.doctor=diagnose.doctor
    -> where diagnose.doctor is null and docshaveschedules.sched=schedule.id and docshaveschedules.doctor=doctor.id;
+------------------------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                                    |
+------------------------------------------------------------------------------------------------------------+
| -> Filter: (diagnose.doctor is null)  (cost=17.02 rows=7) (actual time=11.556..11.597 rows=1 loops=1)
    -> Nested loop antijoin  (cost=17.02 rows=7) (actual time=11.549..11.591 rows=1 loops=1)
        -> Nested loop inner join  (cost=10.30 rows=6) (actual time=9.116..9.194 rows=6 loops=1)
            -> Nested loop inner join  (cost=8.20 rows=6) (actual time=8.980..9.031 rows=6 loops=1)
                -> Covering index scan on docshaveschedules using doctor  (cost=1.60 rows=6) (actual time=4.564..4.569 rows=6 loops=1)
                -> Covering index lookup on schedule using PRIMARY (id=docshaveschedules.sched)  (cost=1.02 rows=1) (actual time=0.740..0.743 rows=1 loops=6)
            -> Single-row index lookup on doctor using PRIMARY (id=docshaveschedules.doctor)  (cost=0.27 rows=1) (actual time=0.026..0.026 rows=1 loops=6)
        -> Covering index lookup on diagnose using doctor (doctor=docshaveschedules.doctor)  (cost=1.02 rows=1) (actual time=0.399..0.399 rows=1 loops=6)
 |
+------------------------------------------------------------------------------------------------------------+
1 row in set (0.03 sec)
```

Query No.4- Display name of patients whose name ends with 6 and gender is female

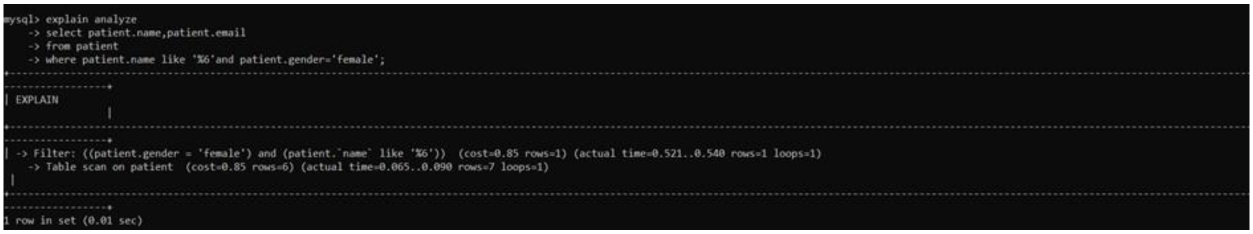
select patient.name,patient.email

from patient

where patient.name like '%6'and patient.gender='female';



```
mysql> select patient.name,patient.email
    -> from patient
    -> where patient.name like '%6' and patient.gender='female';
+----------+---------------+
| name     | email         |
+----------+---------------+
| patient6 | pa6@gmail.com |
+----------+---------------+
1 row in set (0.01 sec)
```



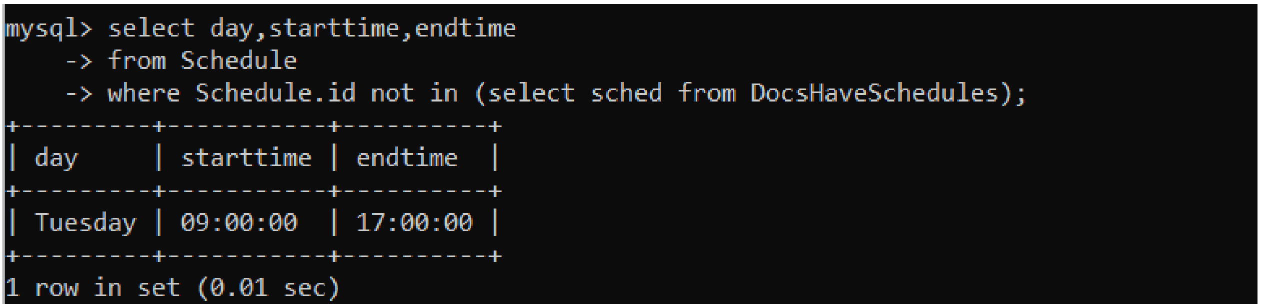
```
mysql> explain analyze
    -> select patient.name,patient.email
    -> from patient
    -> where patient.name like '%6'and patient.gender='female';
+--------------------------------------------------------------------------------------------------------------------------------------
----------------+
| EXPLAIN
                |
+--------------------------------------------------------------------------------------------------------------------------------------
----------------+
| -> Filter: ((patient.gender = 'female') and (patient.`name` like '%6'))  (cost=0.85 rows=1) (actual time=0.521..0.540 rows=1 loops=1)
    -> Table scan on patient  (cost=0.85 rows=6) (actual time=0.065..0.090 rows=7 loops=1)
 |
+--------------------------------------------------------------------------------------------------------------------------------------
----------------+
1 row in set (0.01 sec)
```

Query No.5 - Display schedules which are not scheduled to any doctor.

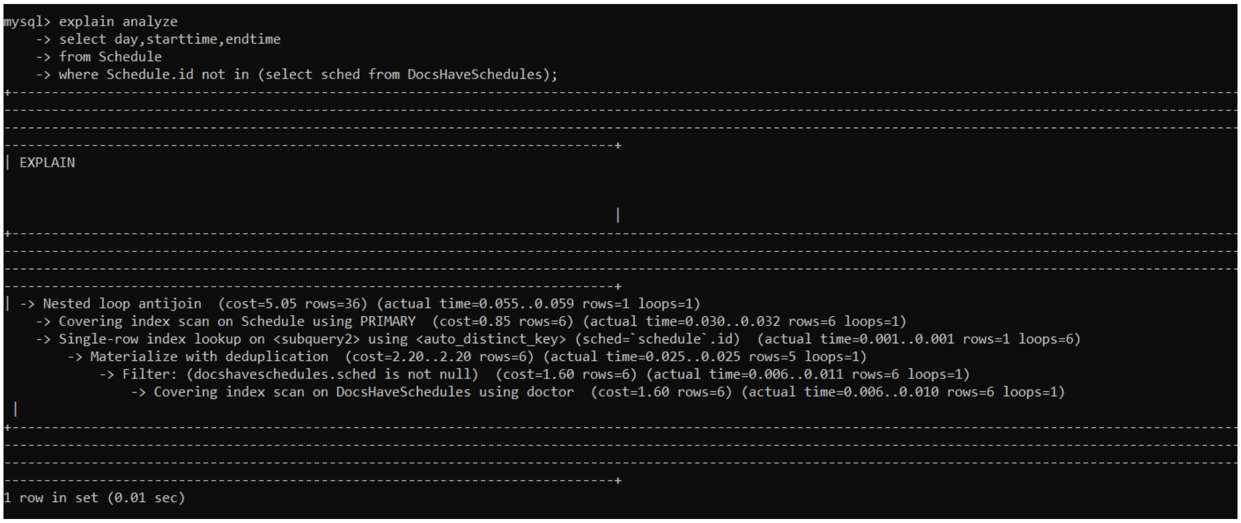
select day,starttime,endtime

from Schedule

where Schedule.id not in (select sched from DocsHaveSchedules);



```
mysql> select day,starttime,endtime
    -> from Schedule
    -> where Schedule.id not in (select sched from DocsHaveSchedules);
+---------+-----------+----------+
| day     | starttime | endtime  |
+---------+-----------+----------+
| Tuesday | 09:00:00  | 17:00:00 |
+---------+-----------+----------+
1 row in set (0.01 sec)
```

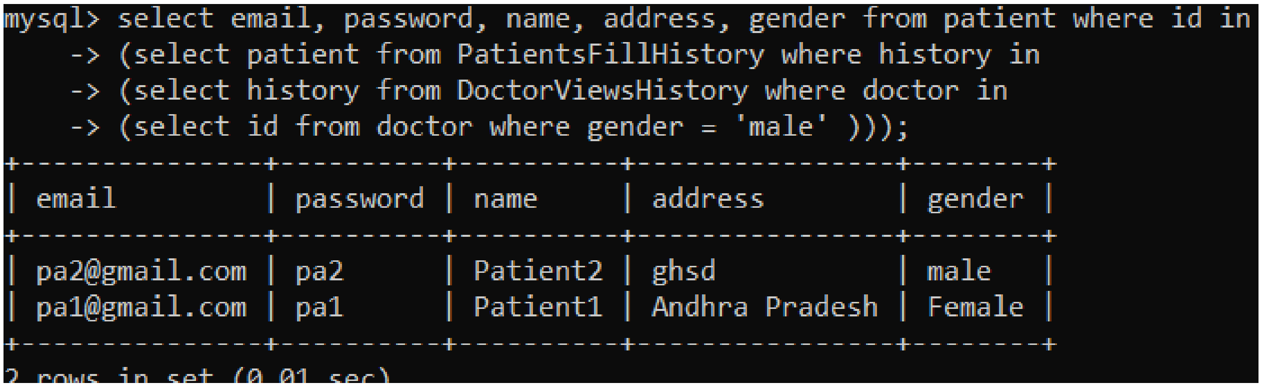


```
mysql> explain analyze
    -> select day,starttime,endtime
    -> from Schedule
    -> where Schedule.id not in (select sched from DocsHaveSchedules);
+--------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------+
| EXPLAIN



                                                                       |
+--------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------+
| -> Nested loop antijoin  (cost=5.05 rows=36) (actual time=0.055..0.059 rows=1 loops=1)
    -> Covering index scan on Schedule using PRIMARY  (cost=0.85 rows=6) (actual time=0.030..0.032 rows=6 loops=1)
    -> Single-row index lookup on <subquery2> using <auto_distinct_key> (sched=`schedule`.id)  (actual time=0.001..0.001 rows=1 loops=6)
        -> Materialize with deduplication  (cost=2.20..2.20 rows=6) (actual time=0.025..0.025 rows=5 loops=1)
            -> Filter: (docshaveschedules.sched is not null)  (cost=1.60 rows=6) (actual time=0.006..0.011 rows=6 loops=1)
                -> Covering index scan on DocsHaveSchedules using doctor  (cost=1.60 rows=6) (actual time=0.006..0.010 rows=6 loops=1)
 |
+--------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------+
1 row in set (0.01 sec)
```

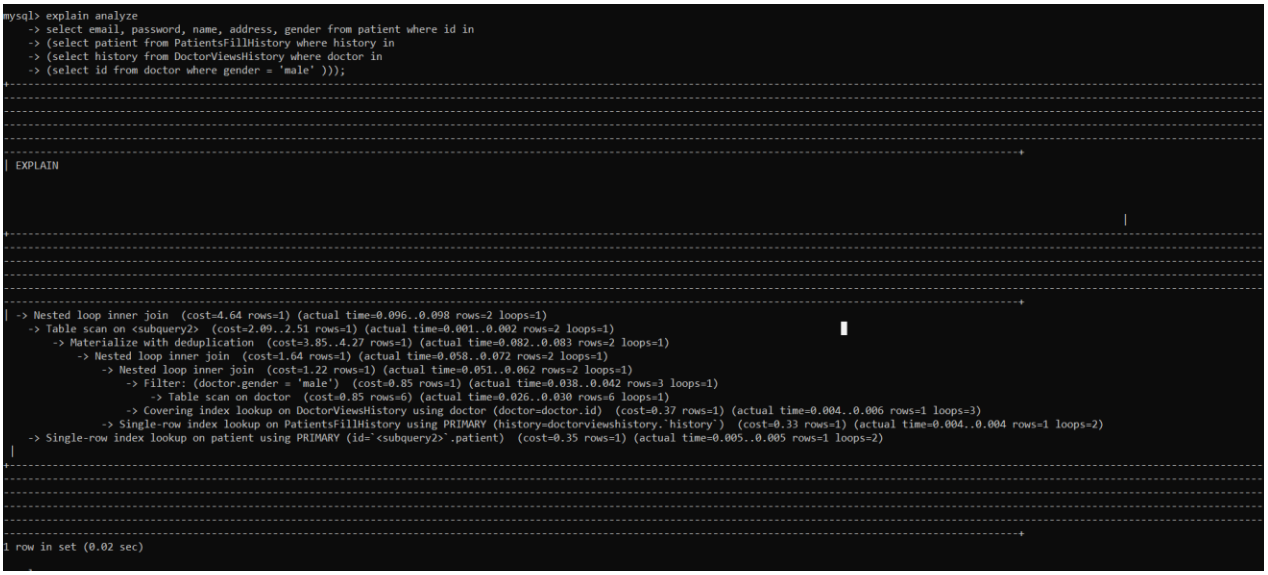
Nested Queries :

Query No.1 - Display email, password, name, address and gender of patient who has been attended by a male doctor.

select email, password, name, address, gender from patient where id in
(select patient from PatientsFillHistory where history in
(select history from DoctorViewsHistory where doctor in
(select id from doctor where gender = 'male')));



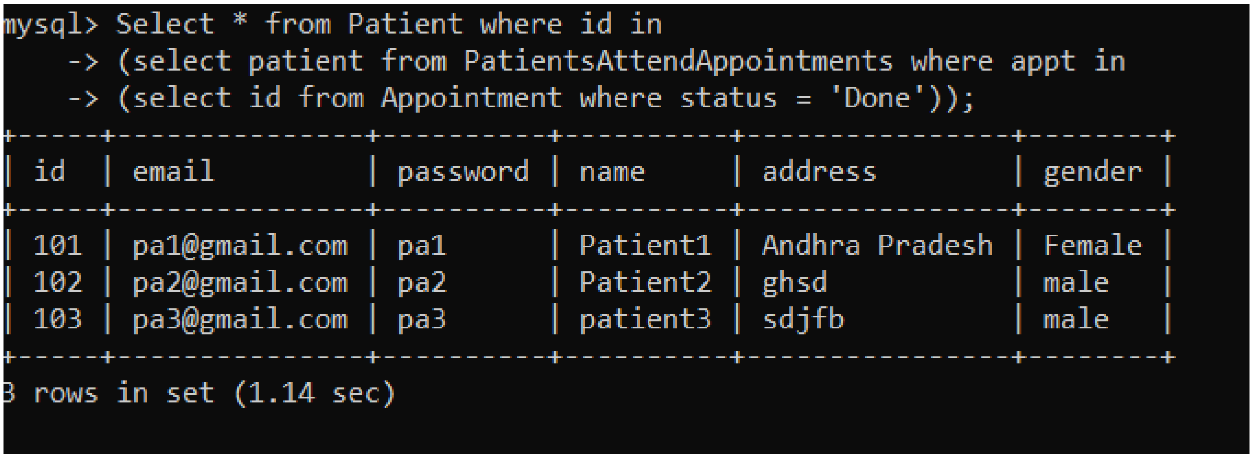
```
mysql> select email, password, name, address, gender from patient where id in
    -> (select patient from PatientsFillHistory where history in
    -> (select history from DoctorViewsHistory where doctor in
    -> (select id from doctor where gender = 'male' )));
+---------------+----------+----------+----------------+--------+
| email         | password | name     | address        | gender |
+---------------+----------+----------+----------------+--------+
| pa2@gmail.com | pa2      | Patient2 | ghsd           | male   |
| pa1@gmail.com | pa1      | Patient1 | Andhra Pradesh | Female |
+---------------+----------+----------+----------------+--------+
2 rows in set (0.01 sec)
```



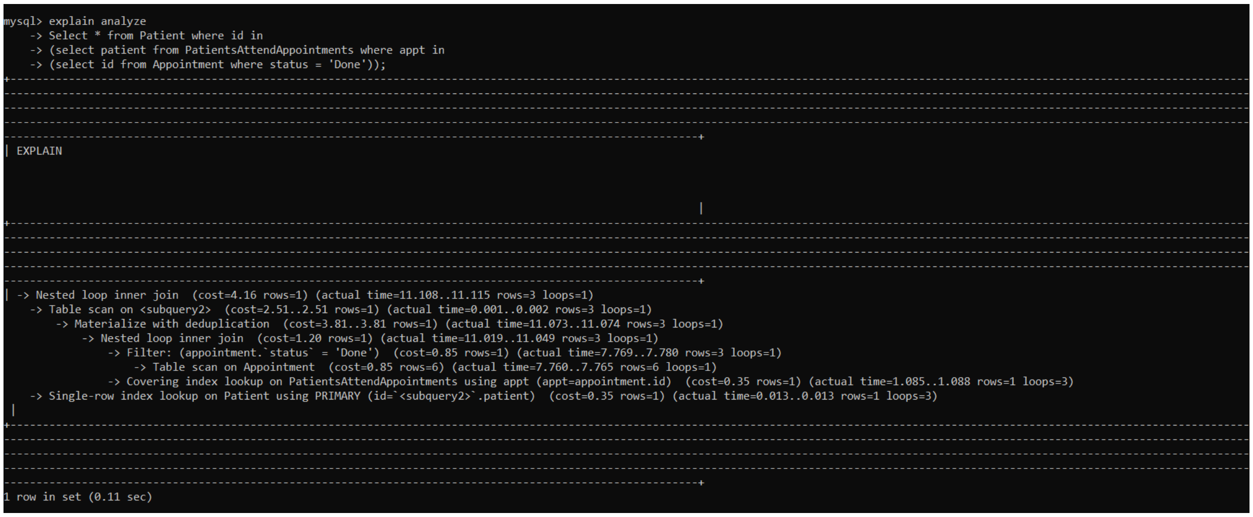
```
mysql> explain analyze
    -> select email, password, name, address, gender from patient where id in
    -> (select patient from PatientsFillHistory where history in
    -> (select history from DoctorViewsHistory where doctor in
    -> (select id from doctor where gender = 'male' )));
+------------------------------------------------------------------------------------------------------------------+
| EXPLAIN                                                                                                          |
+------------------------------------------------------------------------------------------------------------------+
| -> Nested loop inner join  (cost=4.64 rows=1) (actual time=0.096..0.098 rows=2 loops=1)
    -> Table scan on <subquery2>  (cost=2.09..2.51 rows=1) (actual time=0.001..0.002 rows=2 loops=1)
        -> Materialize with deduplication  (cost=3.85..4.27 rows=1) (actual time=0.082..0.083 rows=2 loops=1)
            -> Nested loop inner join  (cost=1.64 rows=1) (actual time=0.058..0.072 rows=2 loops=1)
                -> Nested loop inner join  (cost=1.22 rows=1) (actual time=0.051..0.062 rows=2 loops=1)
                    -> Filter: (doctor.gender = 'male')  (cost=0.85 rows=1) (actual time=0.038..0.042 rows=3 loops=1)
                        -> Table scan on doctor  (cost=0.85 rows=6) (actual time=0.026..0.030 rows=6 loops=1)
                    -> Covering index lookup on DoctorViewsHistory using doctor (doctor=doctor.id)  (cost=0.37 rows=1) (actual time=0.004..0.006 rows=1 loops=3)
                -> Single-row index lookup on PatientsFillHistory using PRIMARY (history=doctorviewshistory.`history`)  (cost=0.33 rows=1) (actual time=0.004..0.004 rows=1 loops=2)
    -> Single-row index lookup on patient using PRIMARY (id=`<subquery2>`.patient)  (cost=0.35 rows=1) (actual time=0.005..0.005 rows=1 loops=2)
 |
+------------------------------------------------------------------------------------------------------------------+
1 row in set (0.02 sec)
```

Query NO.2 - Display all details of patients whose treatment is done.

Select * from Patient where id in
(select patient from PatientsAttendAppointments where appt in
(select id from Appointment where status = 'Done'));



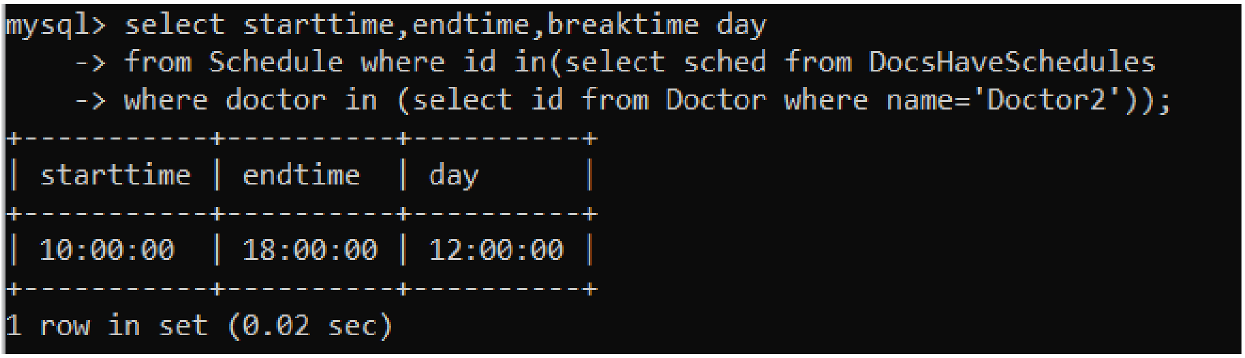
```
mysql> Select * from Patient where id in
    -> (select patient from PatientsAttendAppointments where appt in
    -> (select id from Appointment where status = 'Done'));
+-----+---------------+----------+----------+----------------+--------+
| id  | email         | password | name     | address        | gender |
+-----+---------------+----------+----------+----------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Andhra Pradesh | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | ghsd           | male   |
| 103 | pa3@gmail.com | pa3      | patient3 | sdjfb          | male   |
+-----+---------------+----------+----------+----------------+--------+
3 rows in set (1.14 sec)
```



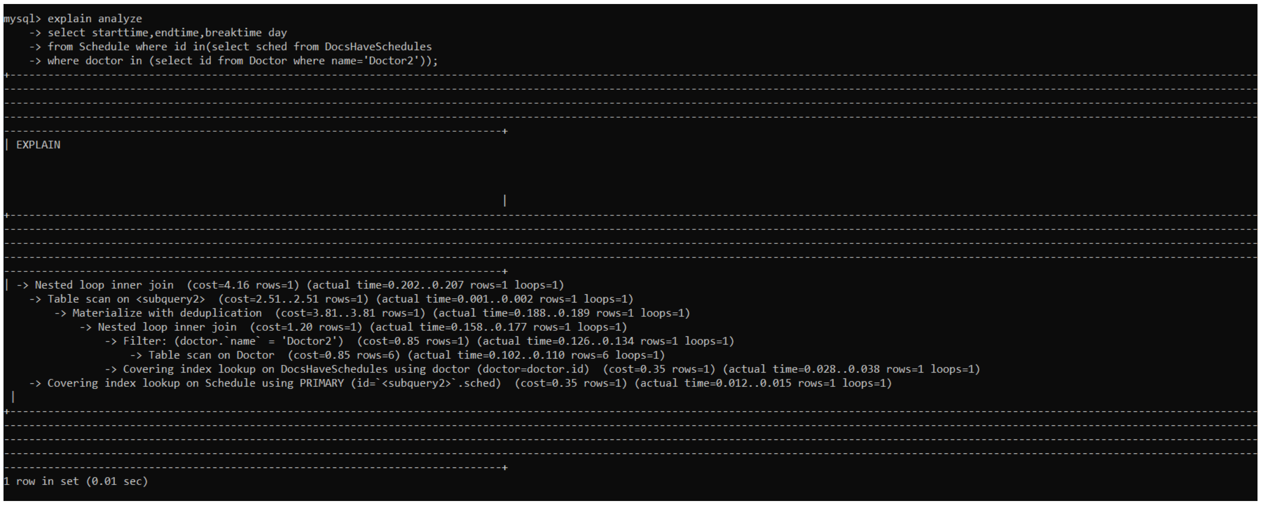
```
mysql> explain analyze
    -> Select * from Patient where id in
    -> (select patient from PatientsAttendAppointments where appt in
    -> (select id from Appointment where status = 'Done'));
+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-----------------------------------------------------------------------+
| EXPLAIN
                                                                       |
+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-----------------------------------------------------------------------+
| -> Nested loop inner join  (cost=4.16 rows=1) (actual time=11.108..11.115 rows=3 loops=1)
    -> Table scan on <subquery2>  (cost=2.51..2.51 rows=1) (actual time=0.001..0.002 rows=3 loops=1)
        -> Materialize with deduplication  (cost=3.81..3.81 rows=1) (actual time=11.073..11.074 rows=3 loops=1)
            -> Nested loop inner join  (cost=1.20 rows=1) (actual time=11.019..11.049 rows=3 loops=1)
                -> Filter: (appointment.`status` = 'Done')  (cost=0.85 rows=1) (actual time=7.769..7.780 rows=3 loops=1)
                    -> Table scan on Appointment  (cost=0.85 rows=6) (actual time=7.760..7.765 rows=6 loops=1)
                -> Covering index lookup on PatientsAttendAppointments using appt (appt=appointment.id)  (cost=0.35 rows=1) (actual time=1.085..1.088 rows=1 loops=3)
    -> Single-row index lookup on Patient using PRIMARY (id=`<subquery2>`.patient)  (cost=0.35 rows=1) (actual time=0.013..0.013 rows=1 loops=3)
 |
+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-----------------------------------------------------------------------+
1 row in set (0.11 sec)
```

Query No.3 - DIsplay the starttime and endtime of ‘Doctor 2’

select starttime,endtime,breaktime day
from Schedule where id in(select sched from DocsHaveSchedules
where doctor in (select id from Doctor where name='Doctor2'));

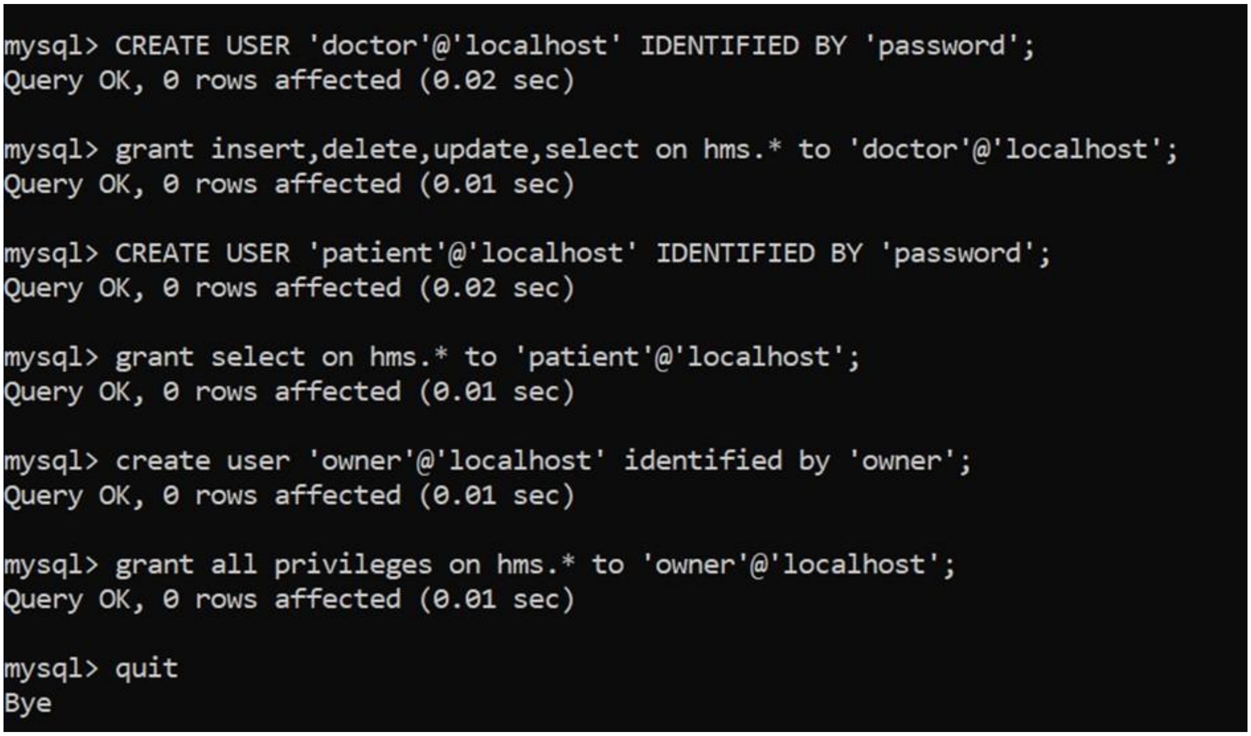


```
mysql> select starttime,endtime,breaktime day
    -> from Schedule where id in(select sched from DocsHaveSchedules
    -> where doctor in (select id from Doctor where name='Doctor2'));
+-----------+----------+----------+
| starttime | endtime  | day      |
+-----------+----------+----------+
| 10:00:00  | 18:00:00 | 12:00:00 |
+-----------+----------+----------+
1 row in set (0.02 sec)
```

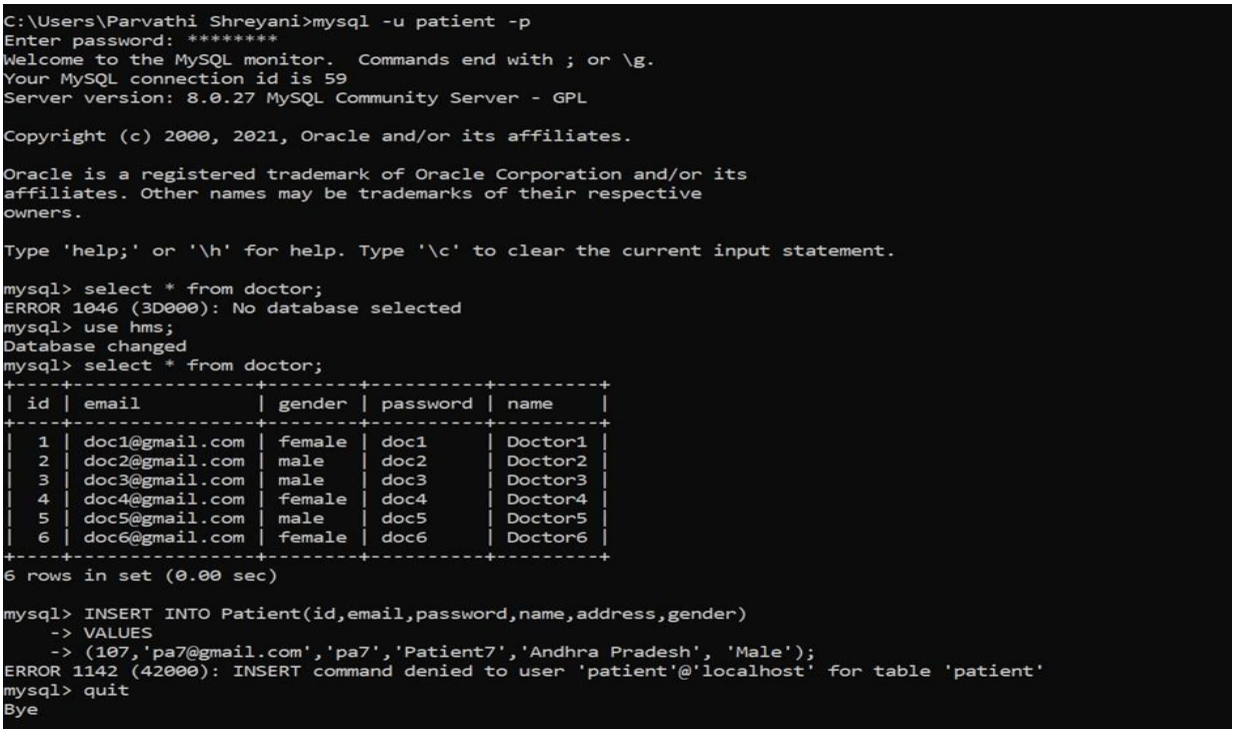


```
mysql> explain analyze
    -> select starttime,endtime,breaktime day
    -> from Schedule where id in(select sched from DocsHaveSchedules
    -> where doctor in (select id from Doctor where name='Doctor2'));
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------+
| EXPLAIN




                                                                                |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------+
| -> Nested loop inner join  (cost=4.16 rows=1) (actual time=0.202..0.207 rows=1 loops=1)
    -> Table scan on <subquery2>  (cost=2.51..2.51 rows=1) (actual time=0.001..0.002 rows=1 loops=1)
        -> Materialize with deduplication  (cost=3.81..3.81 rows=1) (actual time=0.188..0.189 rows=1 loops=1)
            -> Nested loop inner join  (cost=1.20 rows=1) (actual time=0.158..0.177 rows=1 loops=1)
                -> Filter: (doctor.`name` = 'Doctor2')  (cost=0.85 rows=1) (actual time=0.126..0.134 rows=1 loops=1)
                    -> Table scan on Doctor  (cost=0.85 rows=6) (actual time=0.102..0.110 rows=6 loops=1)
                -> Covering index lookup on DocsHaveSchedules using doctor (doctor=doctor.id)  (cost=0.35 rows=1) (actual time=0.028..0.038 rows=1 loops=1)
    -> Covering index lookup on Schedule using PRIMARY (id=`<subquery2>`.sched)  (cost=0.35 rows=1) (actual time=0.012..0.015 rows=1 loops=1)
 |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------+
1 row in set (0.01 sec)
```

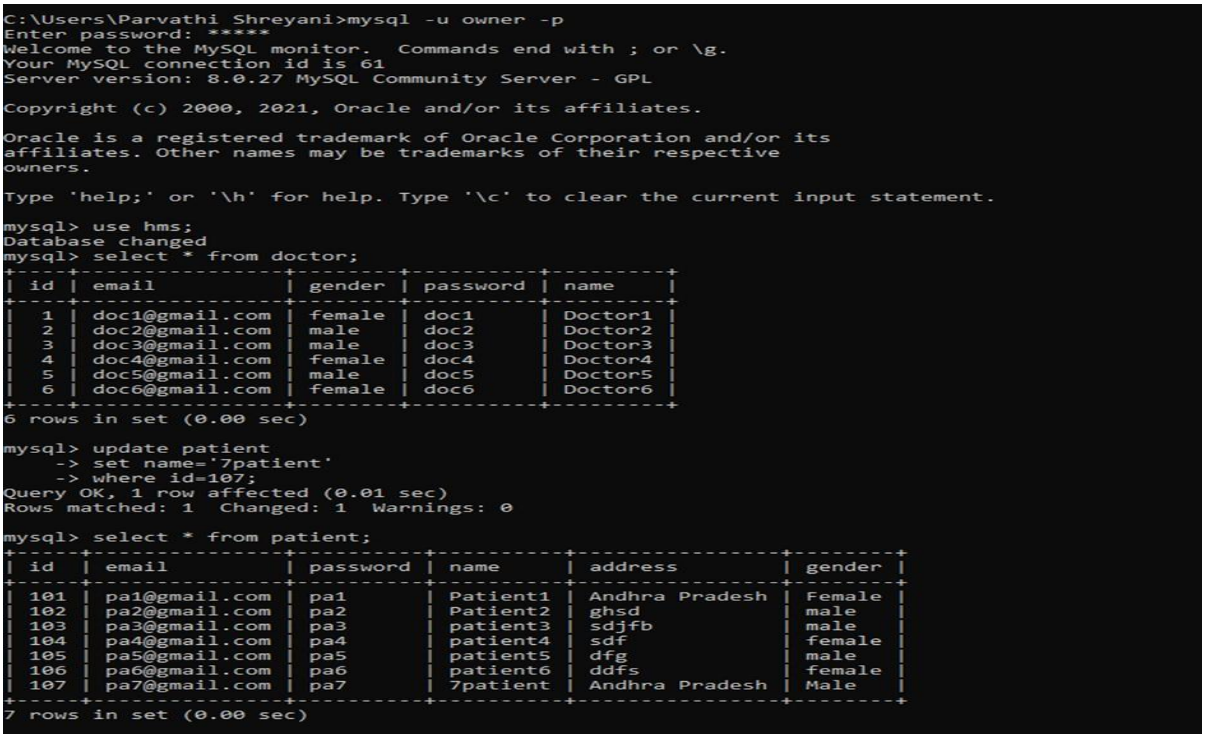
Multiple User Creation:



```
mysql> CREATE USER 'doctor'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.02 sec)

mysql> grant insert,delete,update,select on hms.* to 'doctor'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'patient'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.02 sec)

mysql> grant select on hms.* to 'patient'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> create user 'owner'@'localhost' identified by 'owner';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all privileges on hms.* to 'owner'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql> quit
Bye
```



```
C:\Users\Parvathi Shreyani>mysql -u patient -p
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from doctor;
ERROR 1046 (3D000): No database selected
mysql> use hms;
Database changed
mysql> select * from doctor;
+----+----------------+--------+----------+---------+
| id | email          | gender | password | name    |
+----+----------------+--------+----------+---------+
|  1 | doc1@gmail.com | female | doc1     | Doctor1 |
|  2 | doc2@gmail.com | male   | doc2     | Doctor2 |
|  3 | doc3@gmail.com | male   | doc3     | Doctor3 |
|  4 | doc4@gmail.com | female | doc4     | Doctor4 |
|  5 | doc5@gmail.com | male   | doc5     | Doctor5 |
|  6 | doc6@gmail.com | female | doc6     | Doctor6 |
+----+----------------+--------+----------+---------+
6 rows in set (0.00 sec)

mysql> INSERT INTO Patient(id,email,password,name,address,gender)
    -> VALUES
    -> (107,'pa7@gmail.com','pa7','Patient7','Andhra Pradesh', 'Male');
ERROR 1142 (42000): INSERT command denied to user 'patient'@'localhost' for table 'patient'
mysql> quit
Bye
```

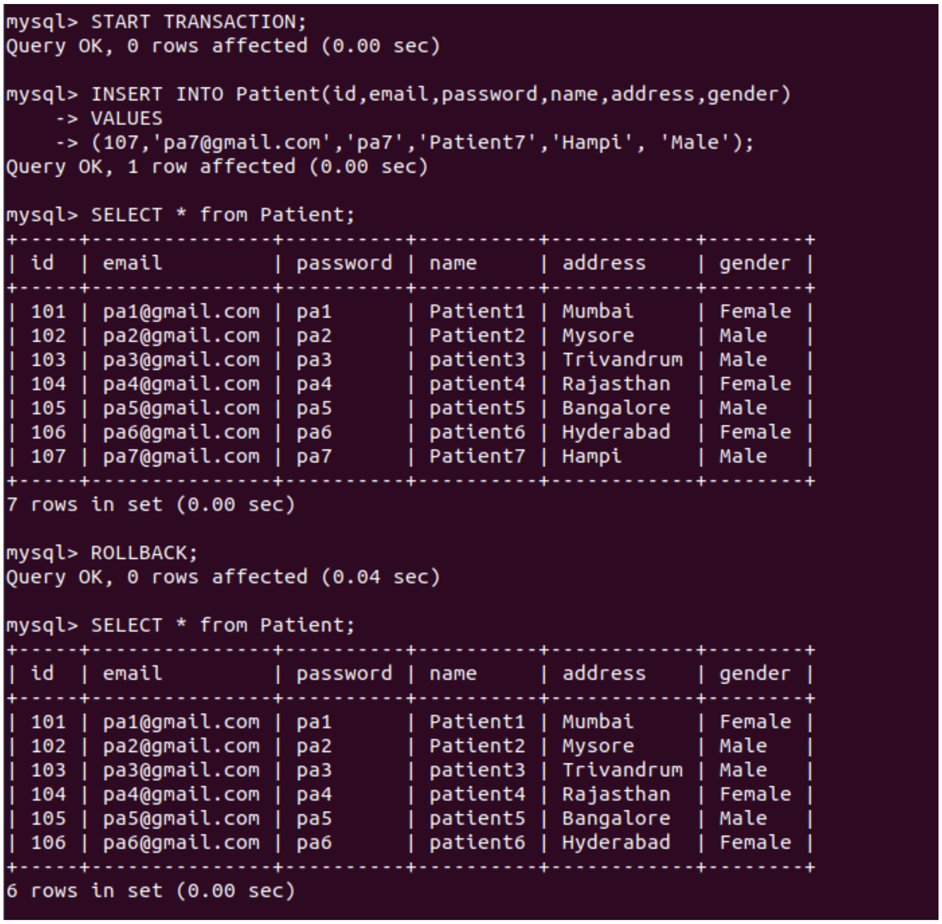


```
C:\Users\Parvathi Shreyani>mysql -u owner -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 61
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use hms;
Database changed
mysql> select * from doctor;
+----+----------------+--------+----------+---------+
| id | email          | gender | password | name    |
+----+----------------+--------+----------+---------+
|  1 | doc1@gmail.com | female | doc1     | Doctor1 |
|  2 | doc2@gmail.com | male   | doc2     | Doctor2 |
|  3 | doc3@gmail.com | male   | doc3     | Doctor3 |
|  4 | doc4@gmail.com | female | doc4     | Doctor4 |
|  5 | doc5@gmail.com | male   | doc5     | Doctor5 |
|  6 | doc6@gmail.com | female | doc6     | Doctor6 |
+----+----------------+--------+----------+---------+
6 rows in set (0.00 sec)

mysql> update patient
    -> set name='7patient'
    -> where id=107;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from patient;
+-----+---------------+----------+-----------+----------------+--------+
| id  | email         | password | name      | address        | gender |
+-----+---------------+----------+-----------+----------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1  | Andhra Pradesh | Female |
| 102 | pa2@gmail.com | pa2      | Patient2  | ghsd           | male   |
| 103 | pa3@gmail.com | pa3      | patient3  | sdjfb          | male   |
| 104 | pa4@gmail.com | pa4      | patient4  | sdf            | female |
| 105 | pa5@gmail.com | pa5      | patient5  | dfg            | male   |
| 106 | pa6@gmail.com | pa6      | patient6  | ddfs           | female |
| 107 | pa7@gmail.com | pa7      | 7patient  | Andhra Pradesh | Male   |
+-----+---------------+----------+-----------+----------------+--------+
7 rows in set (0.00 sec)
```

Transaction control:

Transaction without commit

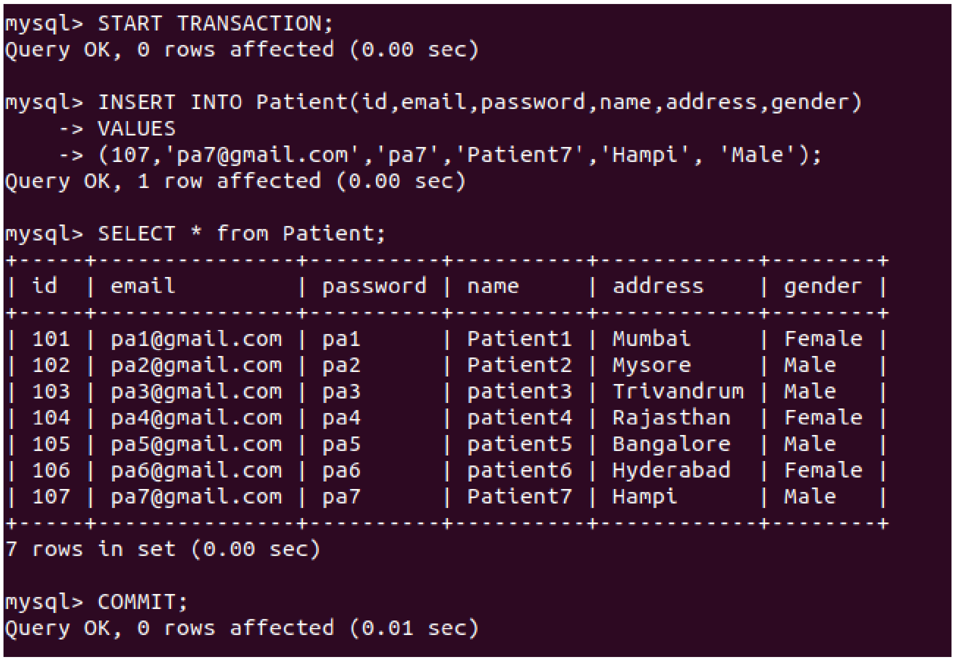
As seen below, when the changes have not been committed, the table goes back to the original when rollback operation is run and the modifications have not been implemented.



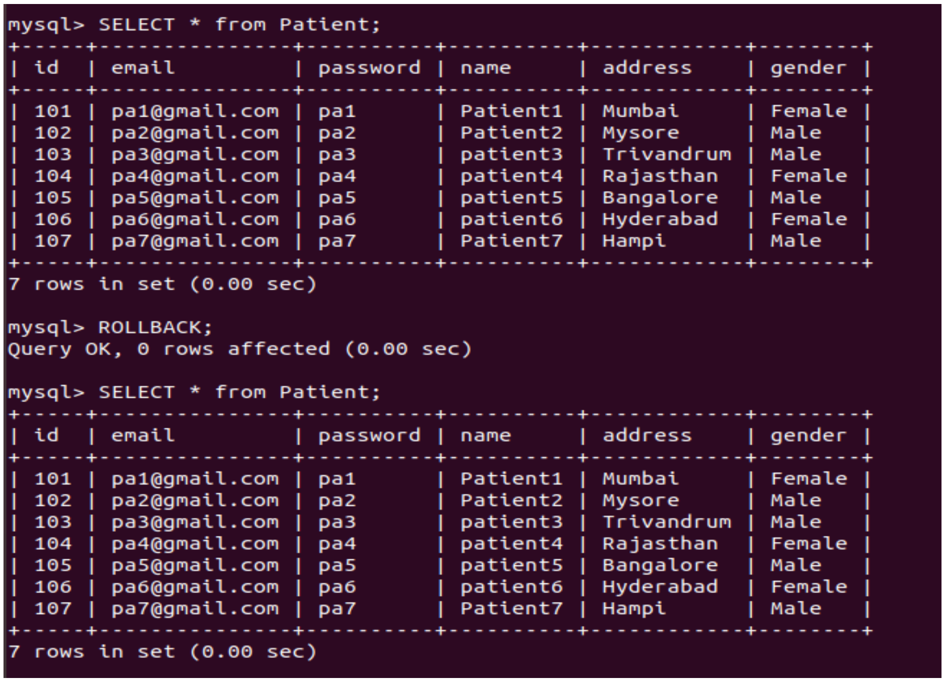
```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO Patient(id,email,password,name,address,gender)
    -> VALUES
    -> (107,'pa7@gmail.com','pa7','Patient7','Hampi', 'Male');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
| 107 | pa7@gmail.com | pa7      | Patient7 | Hampi      | Male   |
+-----+---------------+----------+----------+------------+--------+
7 rows in set (0.00 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0.04 sec)

mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
+-----+---------------+----------+----------+------------+--------+
6 rows in set (0.00 sec)
```

Transaction with commit

As seen below, after performing commit, the changes get saved permanently and on performing rollback, no change is made to the table.



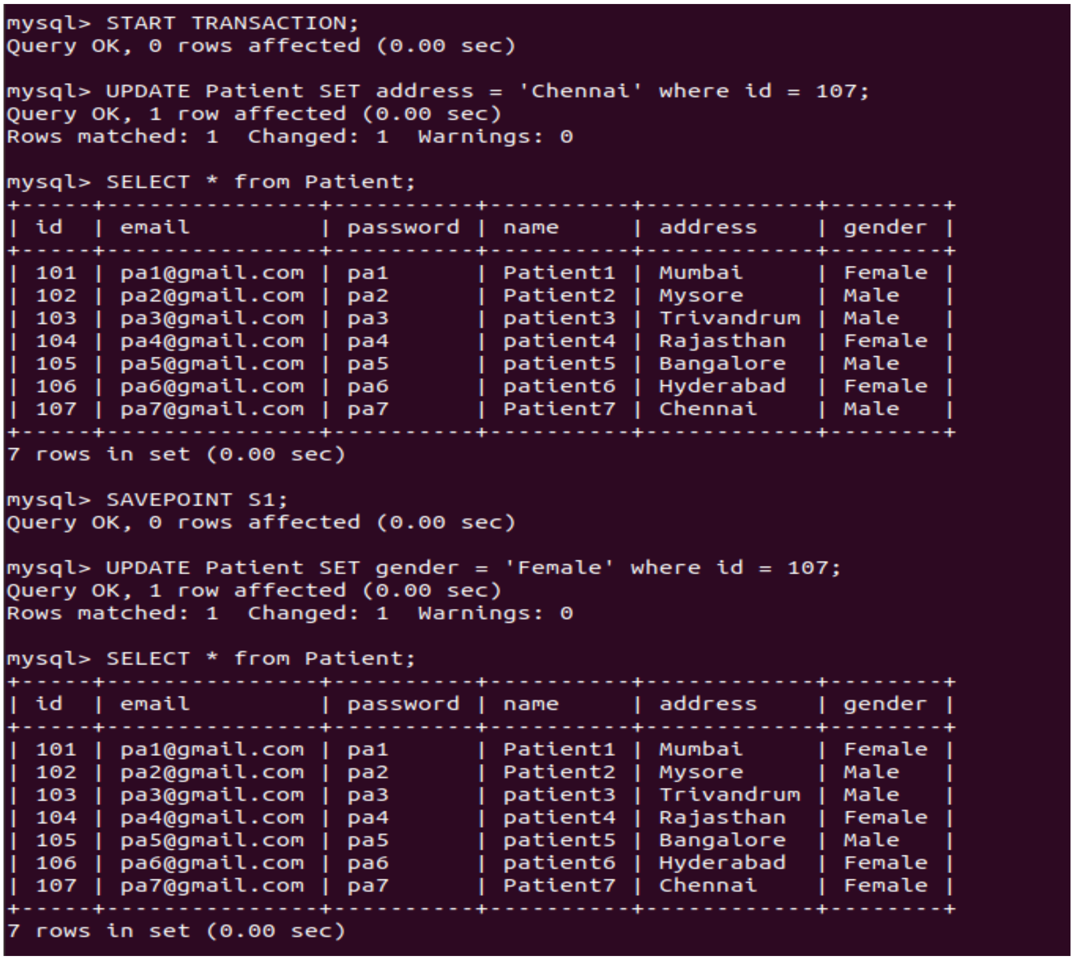
```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO Patient(id,email,password,name,address,gender)
    -> VALUES
    -> (107,'pa7@gmail.com','pa7','Patient7','Hampi', 'Male');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
| 107 | pa7@gmail.com | pa7      | Patient7 | Hampi      | Male   |
+-----+---------------+----------+----------+------------+--------+
7 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.01 sec)
```



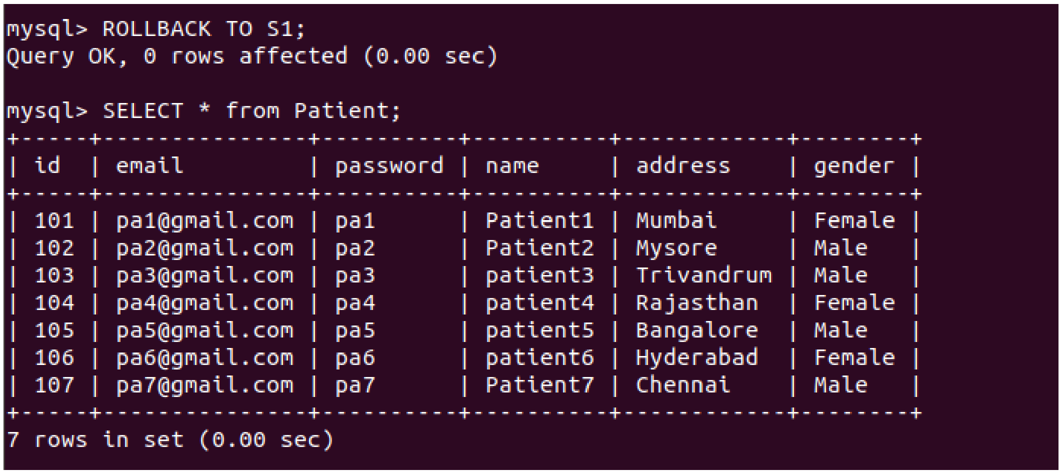
```
mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
| 107 | pa7@gmail.com | pa7      | Patient7 | Hampi      | Male   |
+-----+---------------+----------+----------+------------+--------+
7 rows in set (0.00 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
| 107 | pa7@gmail.com | pa7      | Patient7 | Hampi      | Male   |
+-----+---------------+----------+----------+------------+--------+
7 rows in set (0.00 sec)
```

Transactions with savepoint

In the example below, the first change is made to Patient 7 where their address is changed to Chennai and a savepoint is created. After this, Patient 7 is updated once more and gender is changed to female. On performing rollback now, the changes are rolled back to savepoint 1 as the address remains as Chennai.

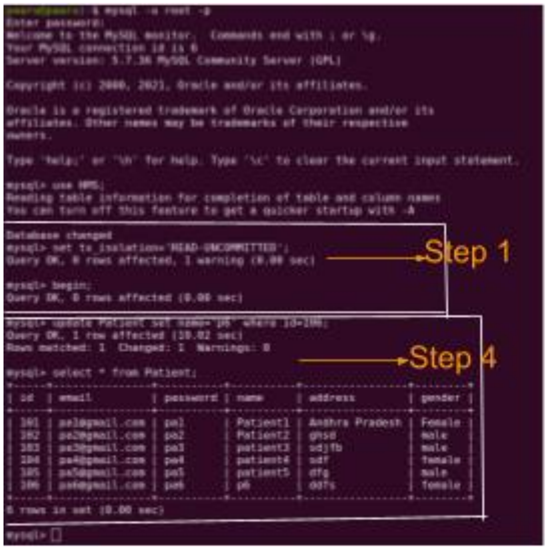


```
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE Patient SET address = 'Chennai' where id = 107;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
| 107 | pa7@gmail.com | pa7      | Patient7 | Chennai    | Male   |
+-----+---------------+----------+----------+------------+--------+
7 rows in set (0.00 sec)

mysql> SAVEPOINT S1;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE Patient SET gender = 'Female' where id = 107;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
| 107 | pa7@gmail.com | pa7      | Patient7 | Chennai    | Female |
+-----+---------------+----------+----------+------------+--------+
7 rows in set (0.00 sec)
```

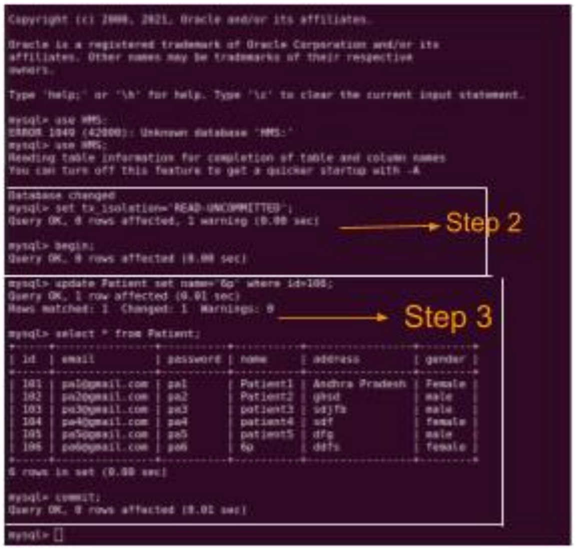


```
mysql> ROLLBACK TO S1;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * from Patient;
+-----+---------------+----------+----------+------------+--------+
| id  | email         | password | name     | address    | gender |
+-----+---------------+----------+----------+------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Mumbai     | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | Mysore     | Male   |
| 103 | pa3@gmail.com | pa3      | patient3 | Trivandrum | Male   |
| 104 | pa4@gmail.com | pa4      | patient4 | Rajasthan  | Female |
| 105 | pa5@gmail.com | pa5      | patient5 | Bangalore  | Male   |
| 106 | pa6@gmail.com | pa6      | patient6 | Hyderabad  | Female |
| 107 | pa7@gmail.com | pa7      | Patient7 | Chennai    | Male   |
+-----+---------------+----------+----------+------------+--------+
7 rows in set (0.00 sec)
```

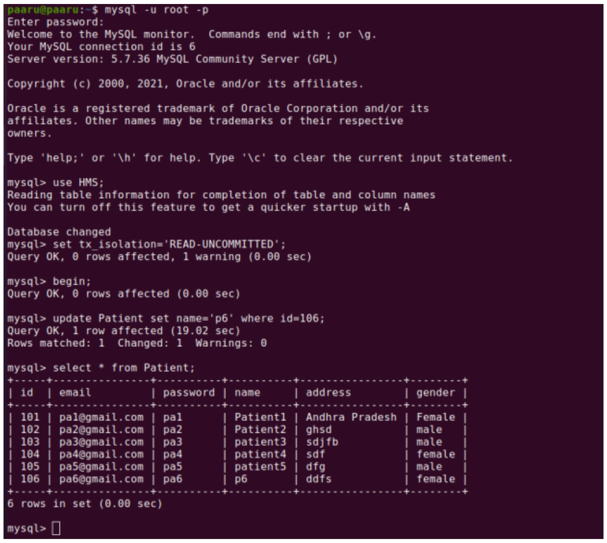
Concurrency transactions:



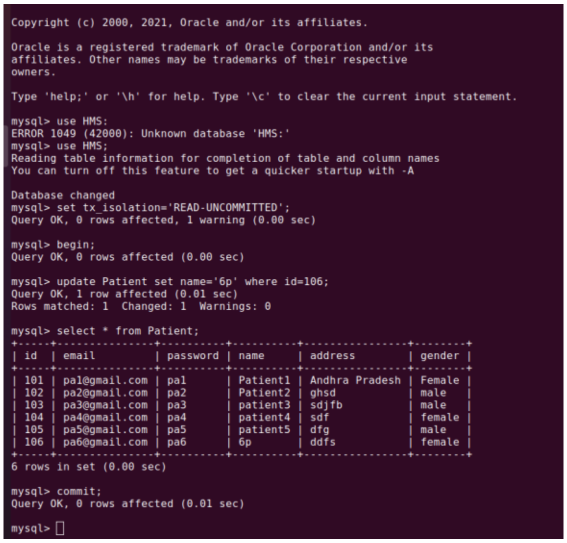
```
paaru@paaru:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.36 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use HMS;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> set tx_isolation='READ-UNCOMMITTED';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> update Patient set name='p6' where id=106;
Query OK, 1 row affected (19.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Patient;
+-----+--------------+----------+----------+----------------+--------+
| id  | email        | password | name     | address        | gender |
+-----+--------------+----------+----------+----------------+--------+
| 101 | pa1@gmail.com | pa1     | Patient1 | Andhra Pradesh | Female |
| 102 | pa2@gmail.com | pa2     | Patient2 | ghsd           | male   |
| 103 | pa3@gmail.com | pa3     | patient3 | sdjfb          | male   |
| 104 | pa4@gmail.com | pa4     | patient4 | sdf            | female |
| 105 | pa5@gmail.com | pa5     | patient5 | dfg            | male   |
| 106 | pa6@gmail.com | pa6     | p6       | ddfs           | female |
+-----+--------------+----------+----------+----------------+--------+
6 rows in set (0.00 sec)

mysql> 
```



```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use HMS:
ERROR 1049 (42000): Unknown database 'HMS:'
mysql> use HMS;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> set tx_isolation='READ-UNCOMMITTED';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> update Patient set name='6p' where id=106;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Patient;
+-----+---------------+----------+----------+----------------+--------+
| id  | email         | password | name     | address        | gender |
+-----+---------------+----------+----------+----------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Andhra Pradesh | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | ghsd           | male   |
| 103 | pa3@gmail.com | pa3      | patient3 | sdjfb          | male   |
| 104 | pa4@gmail.com | pa4      | patient4 | sdf            | female |
| 105 | pa5@gmail.com | pa5      | patient5 | dfg            | male   |
| 106 | pa6@gmail.com | pa6      | 6p       | ddfs           | female |
+-----+---------------+----------+----------+----------------+--------+
6 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.01 sec)

mysql> 
```



```
paaru@paaru:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.36 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use HMS;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> set tx_isolation='READ-UNCOMMITTED';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> update Patient set name='p6' where id=106;
Query OK, 1 row affected (19.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Patient;
+-----+---------------+----------+----------+----------------+--------+
| id  | email         | password | name     | address        | gender |
+-----+---------------+----------+----------+----------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Andhra Pradesh | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | ghsd           | male   |
| 103 | pa3@gmail.com | pa3      | patient3 | sdjfb          | male   |
| 104 | pa4@gmail.com | pa4      | patient4 | sdf            | female |
| 105 | pa5@gmail.com | pa5      | patient5 | dfg            | male   |
| 106 | pa6@gmail.com | pa6      | p6       | ddfs           | female |
+-----+---------------+----------+----------+----------------+--------+
6 rows in set (0.00 sec)

mysql> 
```



```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use HMS:
ERROR 1049 (42000): Unknown database 'HMS:'
mysql> use HMS;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> set tx_isolation='READ-UNCOMMITTED';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> update Patient set name='6p' where id=106;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Patient;
+-----+---------------+----------+----------+----------------+--------+
| id  | email         | password | name     | address        | gender |
+-----+---------------+----------+----------+----------------+--------+
| 101 | pa1@gmail.com | pa1      | Patient1 | Andhra Pradesh | Female |
| 102 | pa2@gmail.com | pa2      | Patient2 | ghsd           | male   |
| 103 | pa3@gmail.com | pa3      | patient3 | sdjfb          | male   |
| 104 | pa4@gmail.com | pa4      | patient4 | sdf            | female |
| 105 | pa5@gmail.com | pa5      | patient5 | dfg            | male   |
| 106 | pa6@gmail.com | pa6      | 6p       | ddfs           | female |
+-----+---------------+----------+----------+----------------+--------+
6 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.01 sec)

mysql> 
```

When two transactions update the same data at the same time, the right transaction is executed successfully, and the left transaction is blocked until the right transaction commits or rolls back, or the left transaction reports an error due to the blocking timeout, then the blocking can be ended.