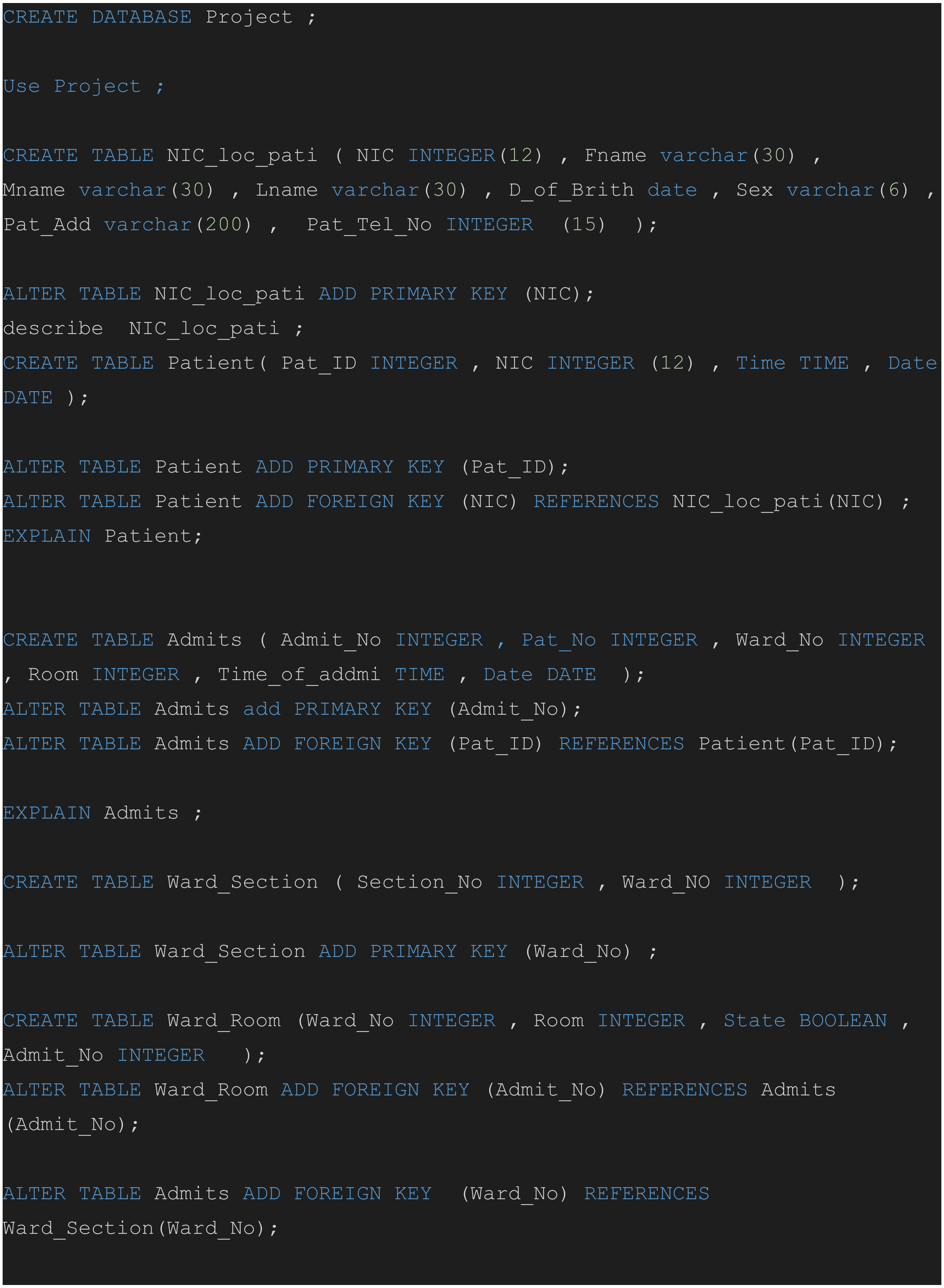


```
CREATE DATABASE Project ;

Use Project ;

CREATE TABLE NIC_loc_pati ( NIC INTEGER(12) , Fname varchar(30) ,
Mname varchar(30) , Lname varchar(30) , D_of_Brith date , Sex varchar(6) ,
Pat_Add varchar(200) ,  Pat_Tel_No INTEGER  (15)  );

ALTER TABLE NIC_loc_pati ADD PRIMARY KEY (NIC);
describe  NIC_loc_pati ;
CREATE TABLE Patient( Pat_ID INTEGER , NIC INTEGER (12) , Time TIME , Date
DATE );

ALTER TABLE Patient ADD PRIMARY KEY (Pat_ID);
ALTER TABLE Patient ADD FOREIGN KEY (NIC) REFERENCES NIC_loc_pati(NIC) ;
EXPLAIN Patient;


CREATE TABLE Admits ( Admit_No INTEGER , Pat_No INTEGER , Ward_No INTEGER
, Room INTEGER , Time_of_addmi TIME , Date DATE  );
ALTER TABLE Admits add PRIMARY KEY (Admit_No);
ALTER TABLE Admits ADD FOREIGN KEY (Pat_ID) REFERENCES Patient(Pat_ID);

EXPLAIN Admits ;

CREATE TABLE Ward_Section ( Section_No INTEGER , Ward_NO INTEGER  );

ALTER TABLE Ward_Section ADD PRIMARY KEY (Ward_No) ;

CREATE TABLE Ward_Room (Ward_No INTEGER , Room INTEGER , State BOOLEAN ,
Admit_No INTEGER   );
ALTER TABLE Ward_Room ADD FOREIGN KEY (Admit_No) REFERENCES Admits
(Admit_No);

ALTER TABLE Admits ADD FOREIGN KEY  (Ward_No) REFERENCES
Ward_Section(Ward_No);
```

```
CREATE TABLE Section (Section_No INTEGER  , Type VARCHAR (30));

ALTER TABLE Section ADD PRIMARY KEY (Section_No) ;
ALTER TABLE Ward_Section ADD FOREIGN KEY (Section_No) REFERENCES
Section(Section_No);

CREATE TABLE Emplo_Sect( Section_No INTEGER , E_ID INTEGER  );
ALTER  TABLE Emplo_Sect ADD FOREIGN KEY (Section_No) REFERENCES
Section(Section_No);

CREATE TABLE Employee ( E_ID INTEGER , NIC INTEGER(12) , Fname VARCHAR
(30) , Mname VARCHAR (30) , Lname VARCHAR (30) , D_of_Brith DATE , Address
VARCHAR (200) , Sex VARCHAR (6) , Contact_No INTEGER (15) );

ALTER  TABLE Employee ADD PRIMARY KEY  (E_ID)  ;
ALTER TABLE Emplo_Sect ADD FOREIGN KEY ( E_ID ) REFERENCES Employee( E_ID
);

CREATE TABLE Doctor( D_ID INTEGER , Spcieciality CHAR (50) ) ;
ALTER TABLE Doctor ADD FOREIGN KEY (D_ID) REFERENCES Employee (E_ID);

CREATE TABLE Nurse( N_ID INTEGER , Grade CHAR (10) ) ;
ALTER TABLE Nurse ADD FOREIGN KEY (N_ID) REFERENCES Employee (E_ID);

CREATE TABLE Security( Sec_ID INTEGER , Grade CHAR (10) ) ;
ALTER TABLE Security ADD FOREIGN KEY (Sec_ID) REFERENCES Employee (E_ID);

CREATE TABLE Clerical( Cler_ID INTEGER , Type CHAR (10) ) ;
ALTER TABLE Clerical ADD FOREIGN KEY (Cler_ID) REFERENCES Employee (E_ID);

CREATE TABLE Clearing( Clear_ID INTEGER , Type CHAR (10) ) ;
ALTER TABLE Clearing ADD FOREIGN KEY (Clear_ID) REFERENCES Employee
(E_ID);


CREATE TABLE Bill ( Bill_No INTEGER  , Pat_ID INTEGER  , Date DATE  , Time
TIME  , Payment FLOAT (8,2) ,  Paid BOOLEAN );
```

```
ALTER TABLE Bill ADD PRIMARY KEY (Bill_No) ;
ALTER TABLE Bill ADD FOREIGN KEY (Pat_ID) REFERENCES Patient(Pat_ID);

CREATE TABLE Bill_Medici (Bill_No INTEGER , Product_ID INTEGER(20) ,
Quantity INTEGER (4) , Price DECIMAL (8,2) );
ALTER TABLE Bill_Medici ADD FOREIGN KEY (Bill_No) REFERENCES
Bill(Bill_No);

CREATE TABLE Medicine (Product_ID INTEGER (20) , Product_code VARCHAR (50)
, Price DECIMAL  (8,2) , MFG DATE , EXP DATE , Quantity INTEGER  );
ALTER TABLE Medicine ADD PRIMARY KEY (Product_ID);
ALTER TABLE Bill_Medici ADD FOREIGN KEY (Product_ID) REFERENCES
Medicine(Product_ID);

ALTER TABLE Admits ADD COLUMN Description varchar(200);
ALTER TABLE Admits ADD COLUMN Leave_date Date ;

ALTER TABLE Ward_Room ADD FOREIGN KEY (Ward_No) REFERENCES
Ward_Section(Ward_No);



ALTER TABLE Patient  AUTO_INCREMENT  ;
ALTER TABLE Patient ALTER COLUMN Pat_ID SET DEFAULT 1;

Alter TABLE Ward_Room ADD PRIMARY KEY (Ward_No,Room);
```

Patient adding procedure

```
DELIMITER //
DROP PROCEDURE IF EXISTS Patient_adding;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Patient_adding`( IN NIC_NO
INTEGER (12) )
BEGIN
```

```
   INSERT IGNORE INTO NIC_loc_pati (NIC) VALUES (NIC_NO)  ;
   DROP TEMPORARY TABLE IF EXISTS tmp ;
   CREATE TEMPORARY TABLE tmp SELECT  (Pat_ID+1) as 'my'   FROM   Patient
ORDER BY Date , time DESC LIMIT 1;
   IF ( SELECT * FROM tmp ) THEN
       BEGIN
       END ;
   ELSE
       BEGIN
       INSERT INTO tmp (my) VALUES(1);
       END ;
   END IF ;

   INSERT INTO Patient(  Pat_ID , NIC , Time , Date ) values( (Select *
from tmp )  , NIC_NO , NOW() , NOW() );

END//
```

Section adding

```
DELIMITER //
DROP PROCEDURE IF EXISTS Section_add;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Section_add`( IN typ VARCHAR
(30) )
BEGIN
   DROP TEMPORARY TABLE IF EXISTS tmp ;
   CREATE TEMPORARY TABLE tmp  SELECT  (Section_No+1) as 'my'  FROM
Section where SType != typ ORDER BY Section_No DESC LIMIT 1;
   IF ( SELECT * FROM tmp ) THEN
       BEGIN
       END ;
   ELSE
       BEGIN
           INSERT INTO tmp values(1);
       END ;

   END IF ;
   INSERT IGNORE  INTO Section ( Section_No , SType )  values((select *
from tmp),typ) ;

END //

```

Ward assign

```
DELIMITER //
DROP PROCEDURE IF EXISTS Ward_assign;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Ward_assign`( Section INTEGER
, Ward INTEGER  )
BEGIN

   INSERT INTO Ward_Section( Section_No , Ward_No ) values ( Section ,
Ward ) ON DUPLICATE KEY UPDATE Section_No = Section ;

END //
```

Ward Room assign

```
DELIMITER //
DROP PROCEDURE IF EXISTS Ward_Room_assign;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Ward_Room_assign`( IN ward
INTEGER  )
BEGIN
  DROP TEMPORARY TABLE IF EXISTS tmp ;
  CREATE TEMPORARY TABLE tmp SELECT  ( Room + 1 ) as 'Rm' FROM Ward_Room
WHERE Ward_No = ward  ORDER BY Room DESC LIMIT 1 ;
  IF ( SELECT * FROM tmp ) THEN
       BEGIN
       END;
   ELSE
       BEGIN
```

```
           INSERT INTO tmp values(1);
       END ;
   END IF ;
  INSERT IGNORE INTO Ward_Room(Ward_No , Room , State ) values ( ward ,
(Select * FROM tmp) , 0 );

END //
```

Admit patient

```
DELIMITER //
DROP PROCEDURE IF EXISTS Admit_Patient;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Admit_Patient`( IN PatID
INTEGER , IN typ INTEGER , IN Descr VARCHAR (200)    )
BEGIN
  DROP TEMPORARY TABLE IF EXISTS tmp ;
  CREATE TEMPORARY TABLE tmp SELECT (admit_No+1) as 'ad' FROM Admits ORDER
BY Date , Time_of_admi DESC LIMIT 1 ;
   IF ( SELECT * FROM tmp ) THEN
       BEGIN
       END ;
   ELSE
       BEGIN
       INSERT INTO tmp (ad) VALUES(1);
       END ;
   END IF ;

  DROP TEMPORARY TABLE IF EXISTS tmp1 ;
  CREATE TEMPORARY TABLE tmp1 SELECT  Ward_Room.Ward_No , Ward_Room.Room
FROM Ward_Room , Ward_Section  WHERE Ward_Section.Section_No = typ and
Ward_Section.Ward_No = Ward_Room.Ward_No and Ward_Room.State = 0 ORDER BY
Room ASC  LIMIT 1 ;
  INSERT IGNORE INTO Admits(Admit_No  , Pat_ID , Time_of_admi , Date ,
Description  ) Value ( (select * from tmp) , PatID , NOW() , NOW() , Descr
) ;
```

```
  UPDATE Ward_Room set State = 1 , Admit_No = (SELECT * FROM tmp) WHERE
Ward_No =  (SELECT Ward_No FROM tmp1) and Room = ( SELECT Room FROM  tmp1
) and Ward_Room.State = 0 ORDER BY Room ASC LIMIT 1 ;
  UPDATE Admits set Ward_No = ( SELECT Ward_No FROM  tmp1 ) , Room =
(SELECT Room FROM tmp1) where Admit_No = ( SELECT * FROM tmp ) ;
END//
```

Medicine adding

```
DELIMITER //
DROP PROCEDURE IF EXISTS medicine_add;

CREATE DEFINER=`root`@`localhost` PROCEDURE `medicine_add`( IN Pro_Code
VARCHAR (50) , IN Prc DECIMAL (8,2) , IN mfg DATE , IN exp DATE , IN Qut
INTEGER  )
BEGIN
   DROP TEMPORARY TABLE IF EXISTS tmp ;
   CREATE TEMPORARY TABLE tmp  SELECT  (Product_ID+1) as 'my'  FROM
Medicine  ORDER BY Product_ID DESC LIMIT 1;
   INSERT IGNORE  INTO Medicine ( Product_ID , Product_code , Price , MFG
, EXP , Quantity )  values((select * from tmp) , Pro_Code  , Prc , mfg ,
exp , Qut) ;

END //
```

Expired remove

```

DELIMITER //
CREATE EVENT IF NOT EXISTS expired
ON SCHEDULE
EVERY 1 DAY
DO
BEGIN

DELETE  FROM Medicine WHERE EXP < NOW();

END//
```

Medicine used by the patient

```
DELIMITER //
DROP PROCEDURE IF EXISTS Medicine_used_on_patient;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Medicine_used_on_patient`( IN
PatID INTEGER  , IN Pro_Code VARCHAR  (50) , IN Qnt INTEGER  )
BEGIN


   INSERT IGNORE  INTO Bill(Bill_No,Pat_ID,Date,Time)
value(1,PatID,Now(),Now());

   DROP TEMPORARY TABLE  IF EXISTS tmpmed ;
   DROP TEMPORARY TABLE  IF EXISTS tembill ;
   CREATE    TEMPORARY TABLE tmpmed SELECT * FROM Medicine WHERE
Product_code = Pro_Code and Quantity >=Qnt LIMIT 1 ;
   CREATE    TEMPORARY TABLE tembill SELECT Bill_No FROM Bill WHERE Pat_ID
= PatID and Paid is Null LIMIT 1 ;

   IF ( SELECT * FROM tembill ) THEN
       BEGIN
       END ;
   ELSE
       BEGIN
           INSERT INTO  tembill value( (SELECT (Bill_No+1) as 'bill' FROM
Bill ORDER BY Date , Time DESC LIMIT 1)  );
           INSERT INTO Bill(Bill_No,Pat_ID,Date,Time) value ((Select *
From tembill),PatID,Now(),Now());
       END ;
   END IF ;

   INSERT INTO Bill_Medici( Bill_No , Product_ID ,  Quantity , Price )
values ( (select * from tembill) , (select Product_ID from tmpmed) , Qnt ,
( select (Price * Qnt) as 'price'  from tmpmed  ) );
```

```

END //
```

Total bill count

```
DELIMITER //
DROP PROCEDURE IF EXISTS Billing_patient;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Billing_patient`( IN PatID
INTEGER   )
BEGIN

    SELECT sum(Price) as 'Total' , Paid FROM Bill_Medici , Bill WHERE
Bill_Medici.Bill_No = Bill.Bill_No and Bill.Pat_ID = PatID ;

END


END //
```

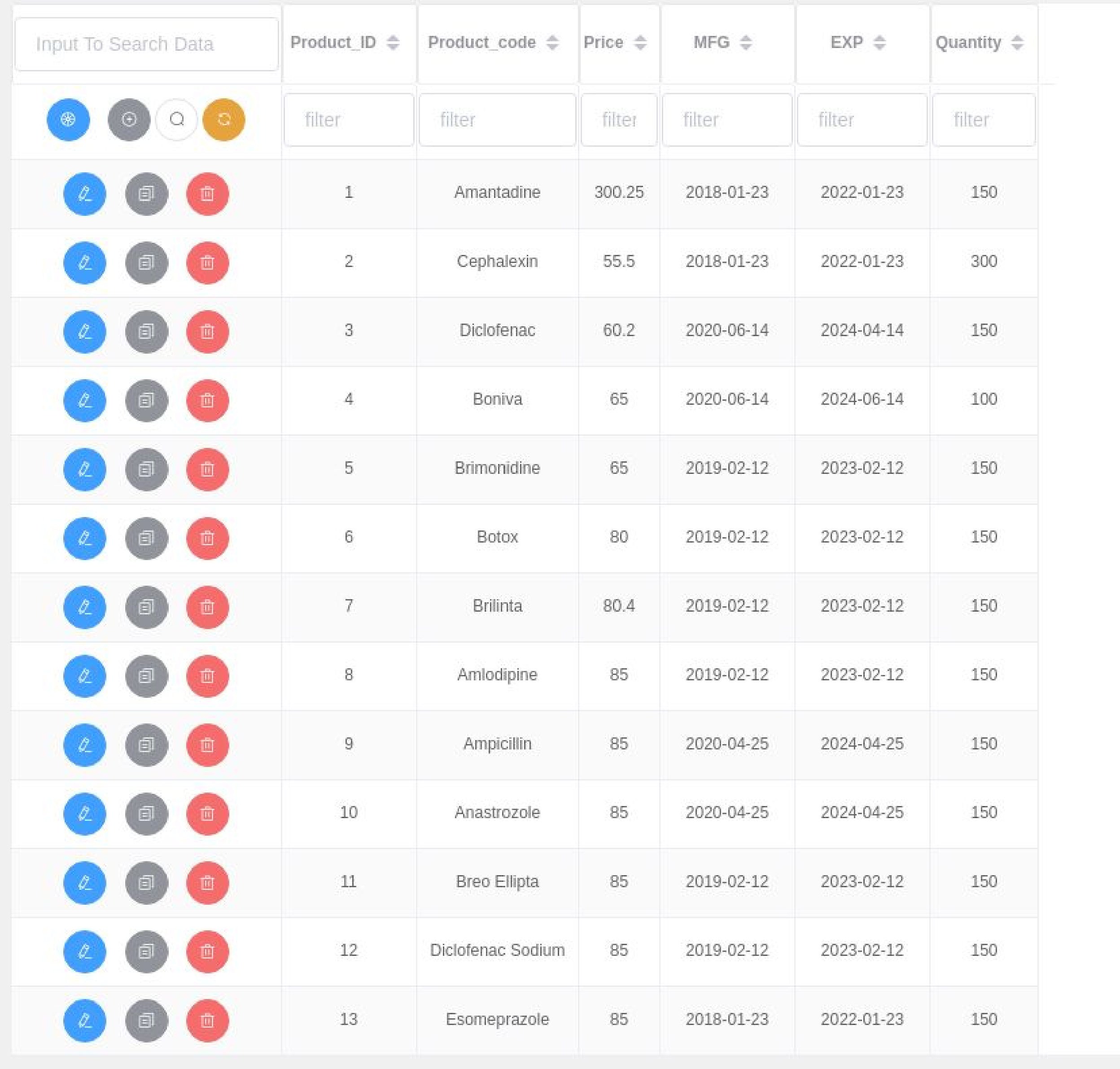
Patient Leaving

```
DELIMITER //
DROP PROCEDURE IF EXISTS Leave_Patien;

CREATE DEFINER=`root`@`localhost` PROCEDURE `Leave_Patien`( IN Adm_No
INTEGER )
BEGIN
   UPDATE Admits SET Leave_date = NOW() WHERE Admit_No = Adm_No ORDER BY
DATE , Time_of_admi DESC LIMIT 1 ;



   DROP TEMPORARY TABLE IF EXISTS tmp ;
   CREATE TEMPORARY TABLE tmp SELECT * FROM Admits WHERE Admit_No = Adm_No
ORDER BY Date , Time_of_admi DESC LIMIT 1 ;

```

```
   UPDATE Ward_Room,tmp SET State = 0 , Ward_Room.Admit_No = Null  WHERE
Ward_Room.Ward_No = tmp.Ward_No
   and Ward_Room.Room = tmp.Room  ;
END //
```

Medicine data input

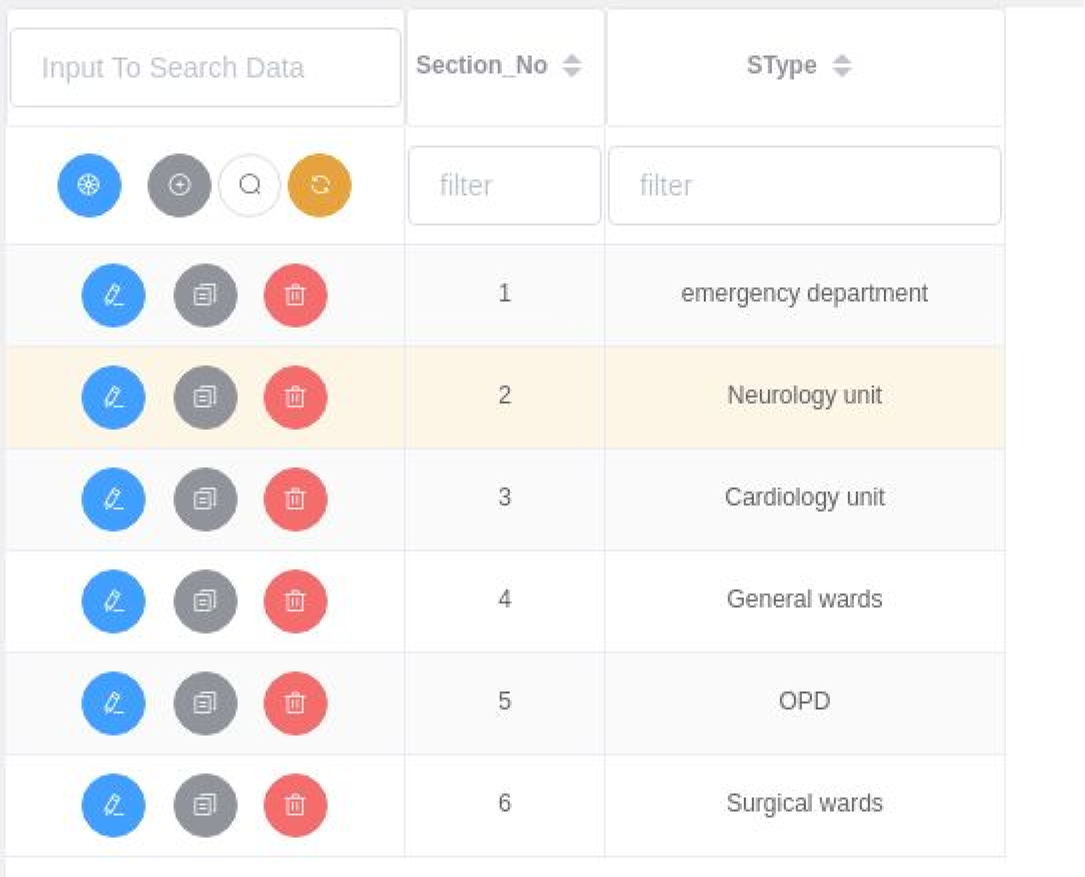
```
CALL medicine_add("Amantadine" ,300.25  ,"2018-01-23",  "2022-01-23",
150);
CALL medicine_add( "Cephalexin",    55.5    ,"2018-01-23",  "2022-01-23",
300);
CALL medicine_add( "Diclofenac" ,60.2   ,"2020-06-14"   ,"2024-04-14",
150);
CALL medicine_add( "Boniva" ,65,"2020-06-14",   "2024-06-14",   100);
CALL medicine_add( "Brimonidine",   65, "2019-02-12",   "2023-02-12",
150);
CALL medicine_add( "Botox", 80, "2019-02-12",   "2023-02-12",   150);
CALL medicine_add( "Brilinta",  80.4,   "2019-02-12",   "2023-02-12",
150);
CALL medicine_add( "Amlodipine",    85, "2019-02-12",   "2023-02-12",
150);
CALL medicine_add( "Ampicillin",    85, "2020-04-25",   "2024-04-25",
150);
CALL medicine_add( "Anastrozole",   85, "2020-04-25",   "2024-04-25",
150);
CALL medicine_add( "Breo Ellipta",  85, "2019-02-12",   "2023-02-12",
150);
CALL medicine_add( "Diclofenac Sodium", 85, "2019-02-12",   "2023-02-12",
150);
CALL medicine_add( "Esomeprazole",  85, "2018-01-23",   "2022-01-23",
150);
```



Input To Search Data	Product_ID	Product_code	Price	MFG	EXP	Quantity
	filter	filter	filter	filter	filter	filter
	1	Amantadine	300.25	2018-01-23	2022-01-23	150
	2	Cephalexin	55.5	2018-01-23	2022-01-23	300
	3	Diclofenac	60.2	2020-06-14	2024-04-14	150
	4	Boniva	65	2020-06-14	2024-06-14	100
	5	Brimonidine	65	2019-02-12	2023-02-12	150
	6	Botox	80	2019-02-12	2023-02-12	150
	7	Brilinta	80.4	2019-02-12	2023-02-12	150
	8	Amlodipine	85	2019-02-12	2023-02-12	150
	9	Ampicillin	85	2020-04-25	2024-04-25	150
	10	Anastrozole	85	2020-04-25	2024-04-25	150
	11	Breo Ellipta	85	2019-02-12	2023-02-12	150
	12	Diclofenac Sodium	85	2019-02-12	2023-02-12	150
	13	Esomeprazole	85	2018-01-23	2022-01-23	150

Section / department add

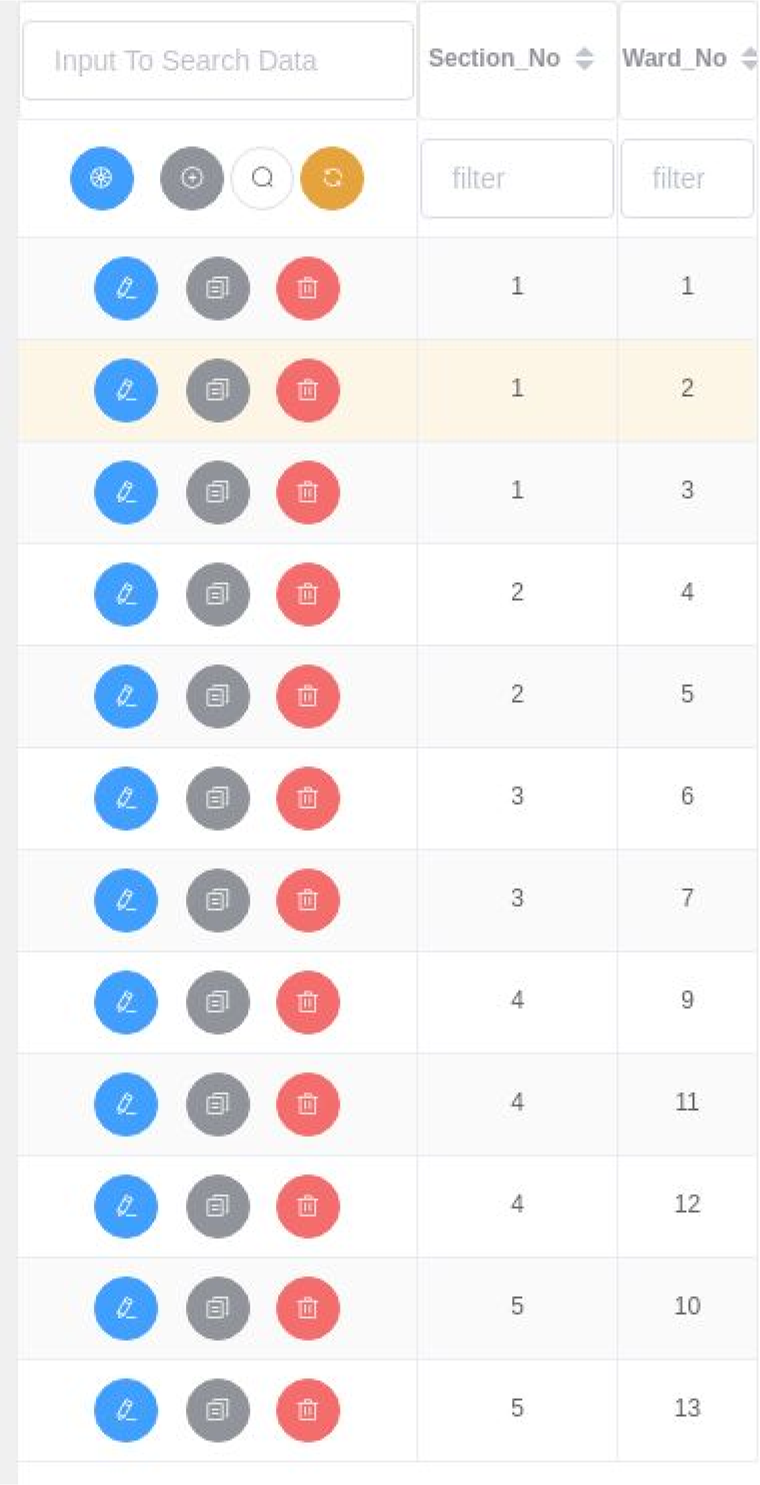
```
CALL Section_add("emergency department");
CALL Section_add("Neurology unit");
CALL Section_add("Cardiology unit");
CALL Section_add("General wards");
CALL Section_add("OPD");
CALL Section_add("Surgical wards");
```



	Section_No	SType
	1	emergency department
	2	Neurology unit
	3	Cardiology unit
	4	General wards
	5	OPD
	6	Surgical wards

Ward assign

```
CALL Ward_assign (1,1);
CALL Ward_assign (1,2);
CALL Ward_assign (1,3);
CALL Ward_assign (2,5);
CALL Ward_assign (2,4);
CALL Ward_assign (3,6);
CALL Ward_assign (3,7);
CALL Ward_assign (4,9);
CALL Ward_assign (5,10);
CALL Ward_assign (4,11);
CALL Ward_assign (4,12);
CALL Ward_assign (5,13);
```



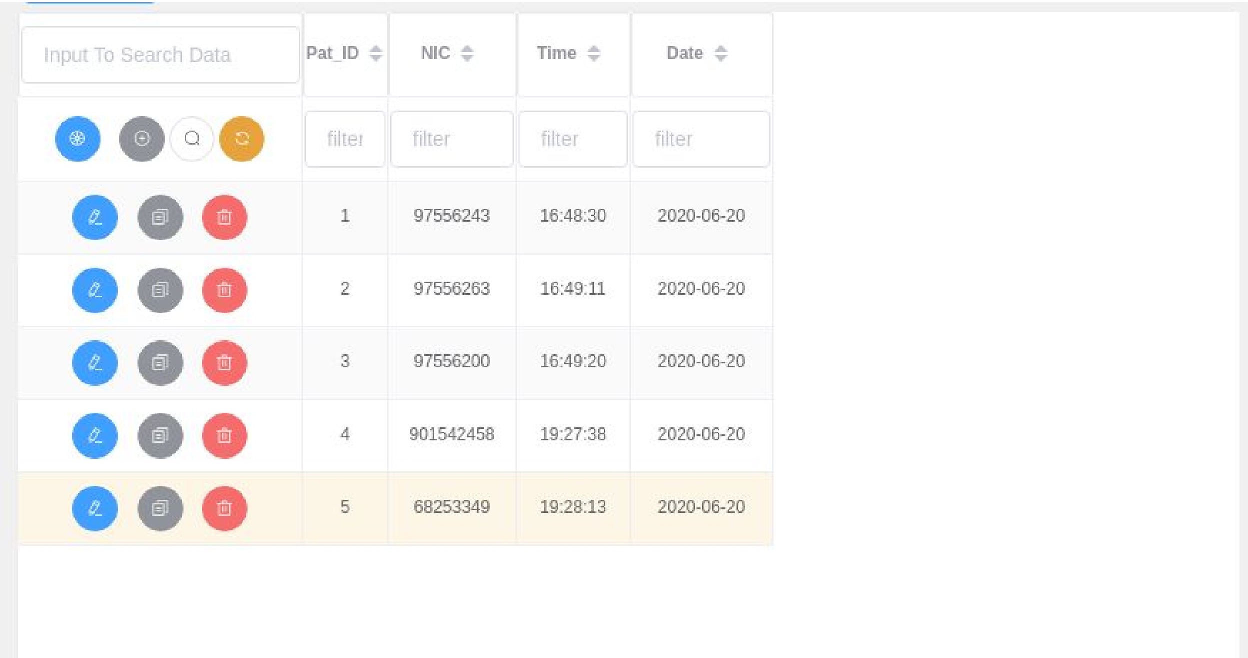
Section_No	Ward_No
1	1
1	2
1	3
2	4
2	5
3	6
3	7
4	9
4	11
4	12
5	10
5	13

Assign room for Ward

```
CALL Ward_Room_assign ( 2 ) ;
CALL Ward_Room_assign ( 2 ) ;
CALL Ward_Room_assign ( 3 ) ;
CALL Ward_Room_assign ( 3 ) ;
CALL Ward_Room_assign ( 3 ) ;
CALL Ward_Room_assign ( 3 ) ;
CALL Ward_Room_assign ( 3 ) ;
CALL Ward_Room_assign ( 5 ) ;
CALL Ward_Room_assign ( 6 ) ;
```

Patient_adding

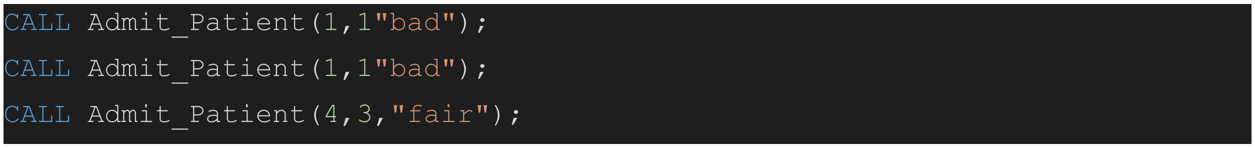
```
CALL Patient_adding(97556243);
CALL Patient_adding(97556243);
CALL Patient_adding(97556200);
CALL Patient_adding(901542458);
CALL Patient_adding(68253349);
```



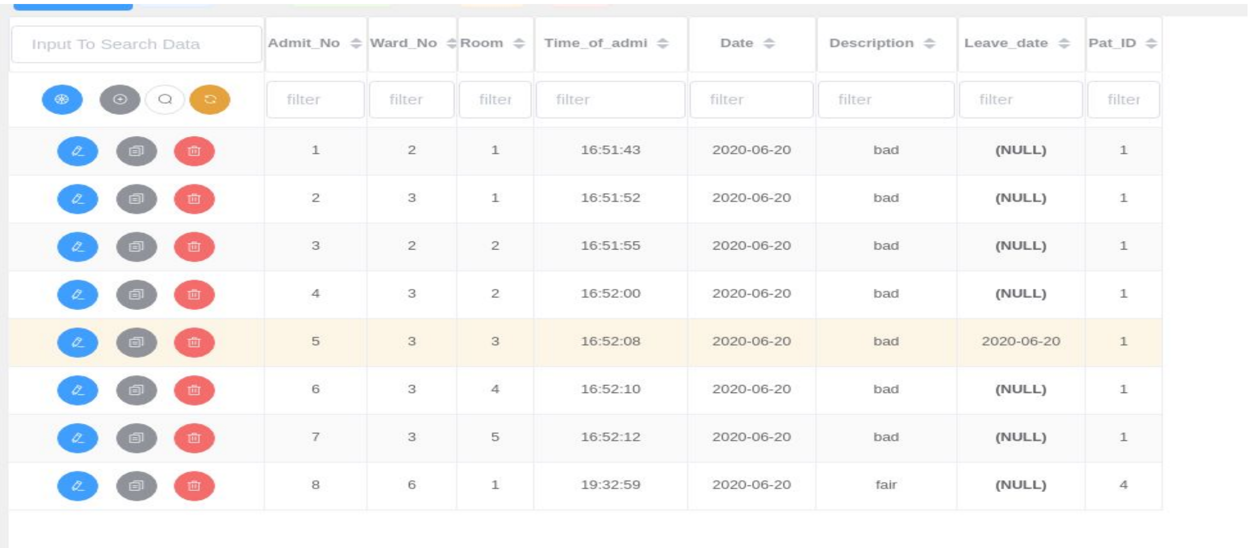
Input To Search Data	Pat_ID	NIC	Time	Date
	filter	filter	filter	filter
	1	97556243	16:48:30	2020-06-20
	2	97556263	16:49:11	2020-06-20
	3	97556200	16:49:20	2020-06-20
	4	901542458	19:27:38	2020-06-20
	5	68253349	19:28:13	2020-06-20

Admit Patient

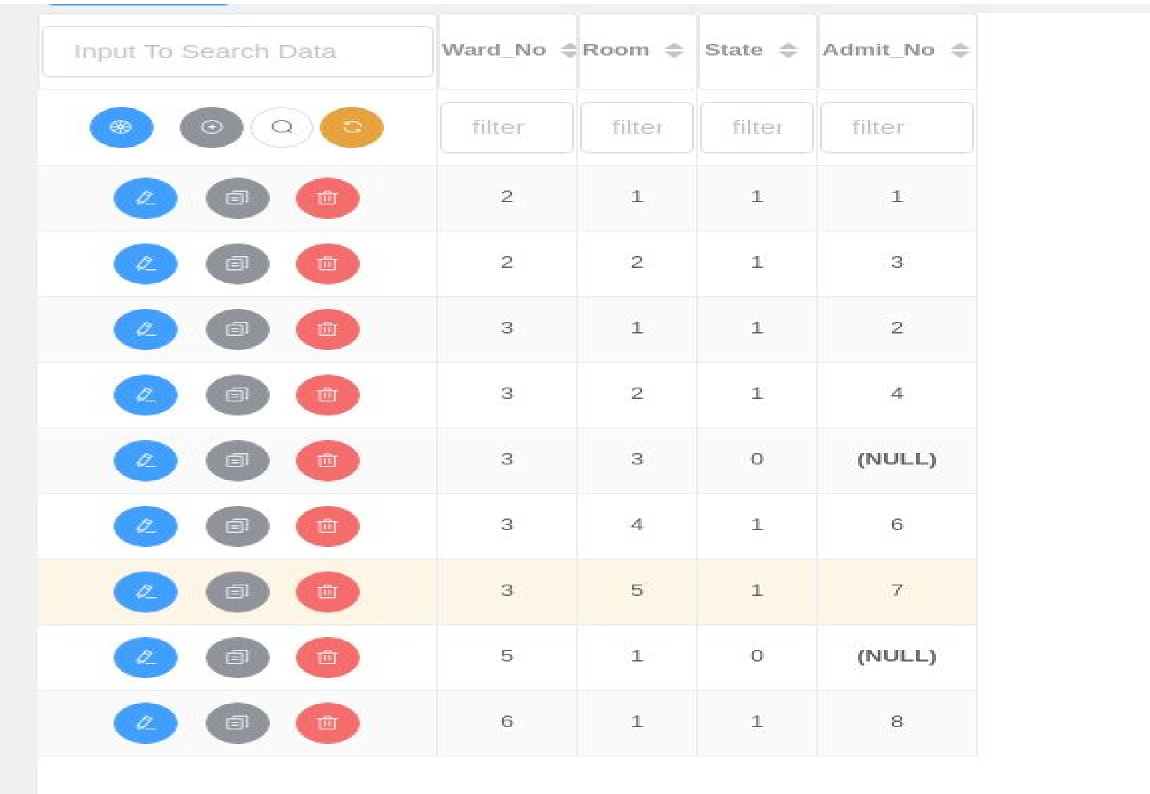
```
CALL Admit_Patient(1,1"bad");
CALL Admit_Patient(1,1"bad");
CALL Admit_Patient(1,1"bad");
CALL Admit_Patient(1,1"bad");
CALL Admit_Patient(1,1"bad");
```



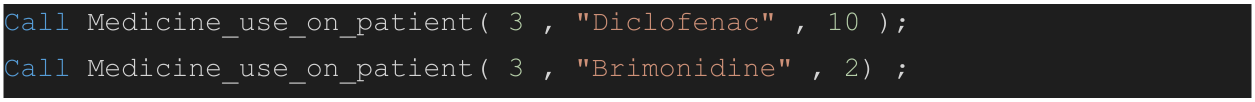
```
CALL Admit_Patient(1,1"bad");
CALL Admit_Patient(1,1"bad");
CALL Admit_Patient(4,3,"fair");
```



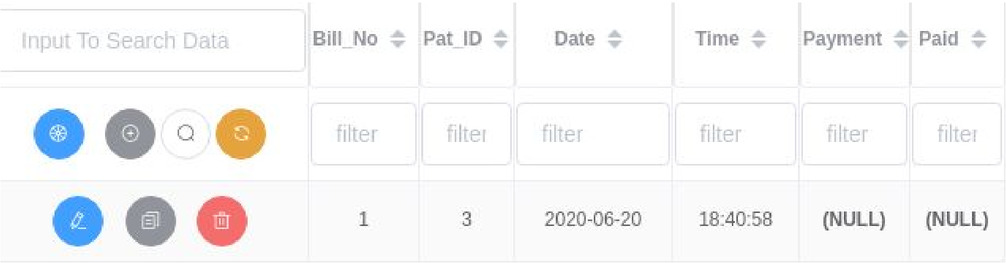
Admit_No	Ward_No	Room	Time_of_admi	Date	Description	Leave_date	Pat_ID
1	2	1	16:51:43	2020-06-20	bad	(NULL)	1
2	3	1	16:51:52	2020-06-20	bad	(NULL)	1
3	2	2	16:51:55	2020-06-20	bad	(NULL)	1
4	3	2	16:52:00	2020-06-20	bad	(NULL)	1
5	3	3	16:52:08	2020-06-20	bad	2020-06-20	1
6	3	4	16:52:10	2020-06-20	bad	(NULL)	1
7	3	5	16:52:12	2020-06-20	bad	(NULL)	1
8	6	1	19:32:59	2020-06-20	fair	(NULL)	4



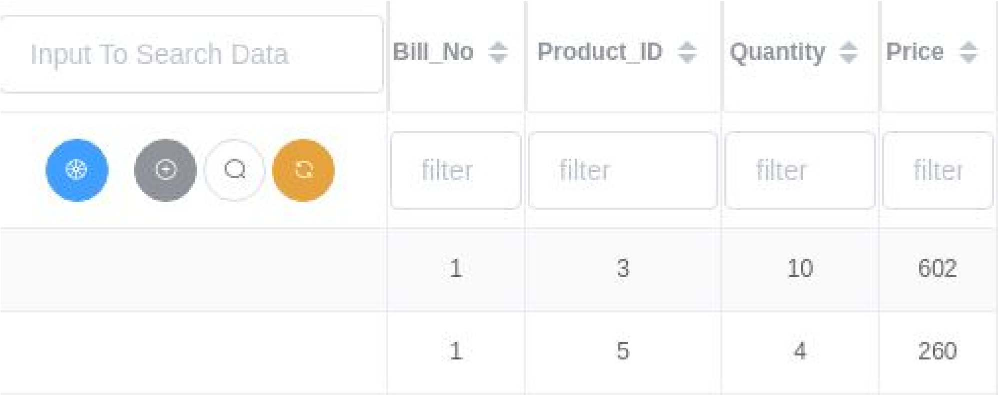
Ward_No	Room	State	Admit_No
2	1	1	1
2	2	1	3
3	1	1	2
3	2	1	4
3	3	0	(NULL)
3	4	1	6
3	5	1	7
5	1	0	(NULL)
6	1	1	8



```
Call Medicine_use_on_patient( 3 , "Diclofenac" , 10 );
Call Medicine_use_on_patient( 3 , "Brimonidine" , 2) ;
```

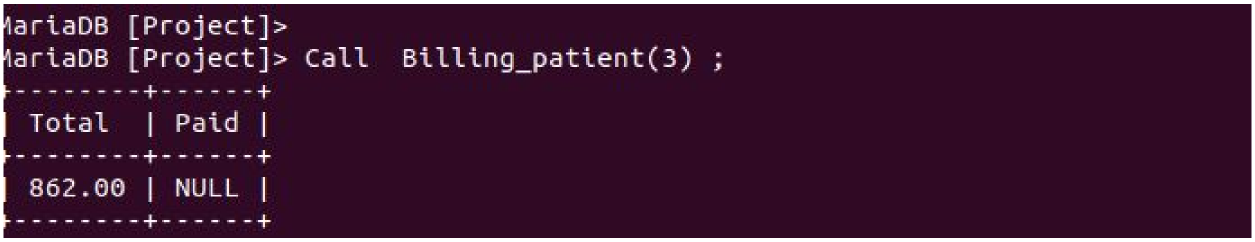


Bill_No	Pat_ID	Date	Time	Payment	Paid
1	3	2020-06-20	18:40:58	(NULL)	(NULL)



Bill_No	Product_ID	Quantity	Price
1	3	10	602
1	5	4	260

Billing patient



```
MariaDB [Project]>
MariaDB [Project]> Call  Billing_patient(3) ;
+--------+------+
| Total  | Paid |
+--------+------+
| 862.00 | NULL |
+--------+------+
```