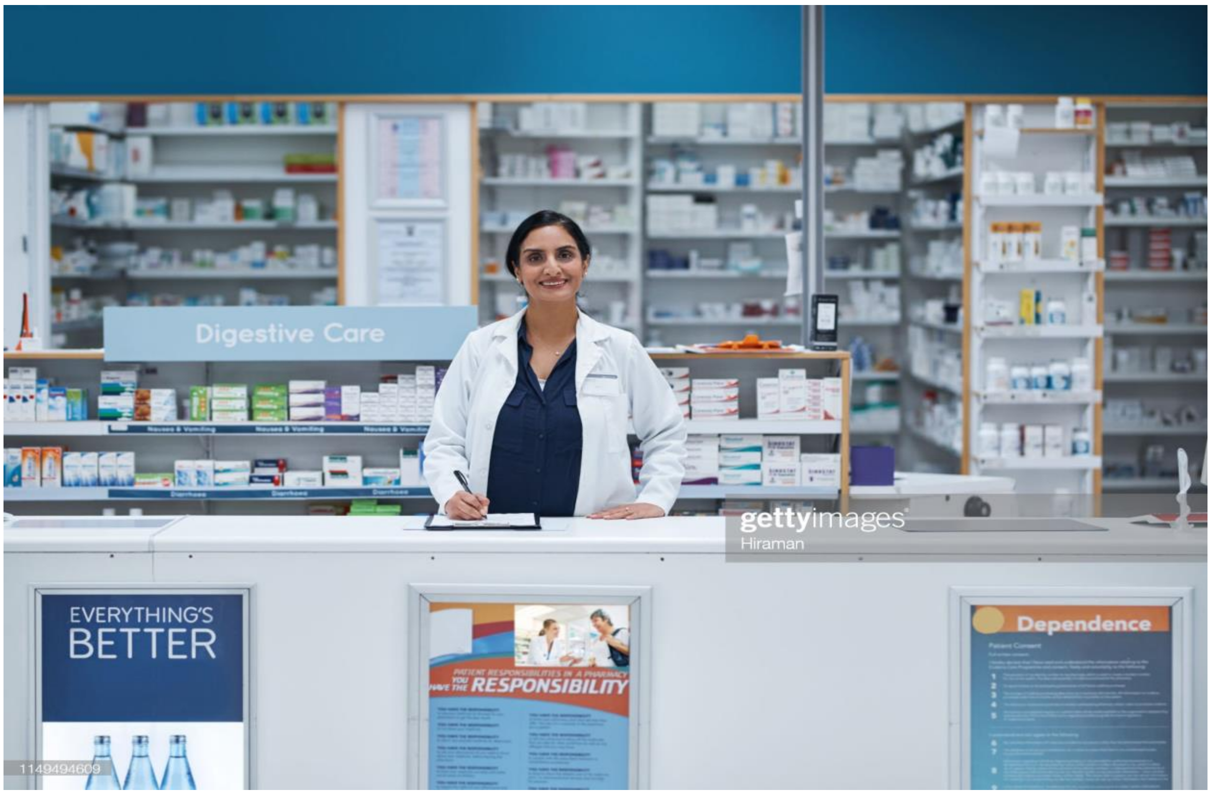
SQL PROJECT

Dhanvantari Medicals

Medical Store Management



Abstract:

The Medical Store Management System is a comprehensive database project designed to streamline and enhance the operations of a Medical store. This project aims to create an efficient and organized platform for managing customer information, medication, Patient & doctor records, and Bill transactions. By implementing this system, the medical store can improve customer service, optimize inventory control, and maintain accurate records of its activities.

Aim:

The aim of this project is to develop a robust SQL-based medical Store Management System that addresses the specific needs and challenges of a medical store. This system will serve as a central repository for storing, managing, and retrieving critical data related to customers, medications, prescriptions, Doctors Employees and Bill etc. The primary goal is to improve the overall efficiency, accuracy, and customer satisfaction of the Medical store.

Objectives:

Patient Management: Develop a feature for adding, updating, and deleting customer information, including personal details and information.

Medicines: Create a module to manage medication inventory, including details such as name, description, Vendors, unit price, and Expire and Manufacturing Date.

Doctors Records: Implement a system for recording and managing prescription details for customers, including prescription dates, doctor names, and prescribed medications.

Bill Generation: Generate invoices or receipts for customers after completing a sale transaction, providing a clear breakdown of the purchased items and their costs.

Pharmacist Management : If employees are part of the system, monitor their information and assess their sales performance.

Vendors : vendors also main part of medicals full fill the medical requirement as per order .

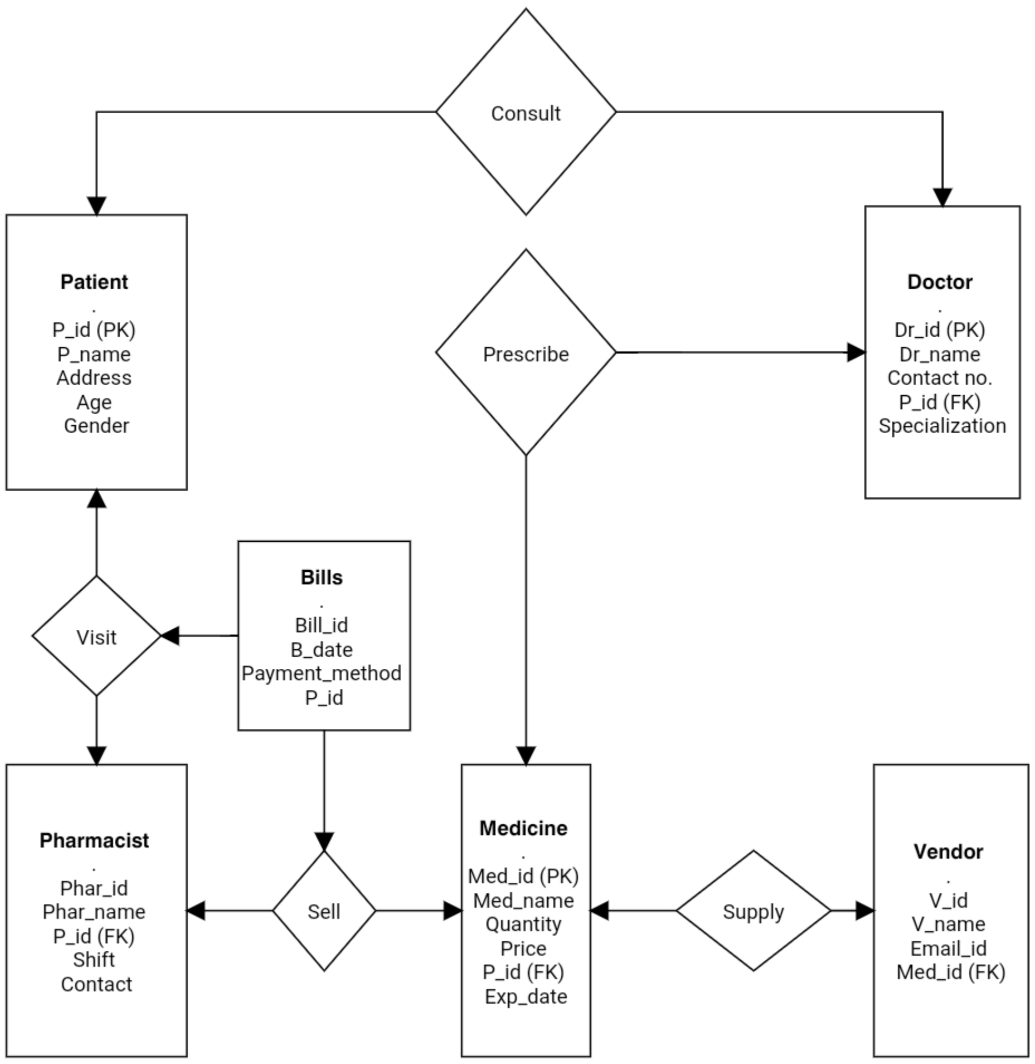
Introduction:

The modern medical store faces the challenge of efficiently managing its operations while providing quality service to customers. To address these challenges, we present the medical Store Management System, a database project designed to streamline the store's daily activities. This system aims to create an organized and efficient environment for managing customer information, medication inventory, prescription records, and sales transactions. By centralizing data and automating key processes, the pharmacy store can enhance customer service, optimize inventory control, and maintain accurate records of its business activities.

In this project, we will detail the design and implementation of the database schema, the SQL queries and statements used for various functions, and the user interface that allows store employees to interact with the system. The project's objectives include improving customer management, medicine ,consulting doctors , sales processing, vendors and reporting. Additionally, we provide an optional feature for employee management to enhance the store's operational transparency.

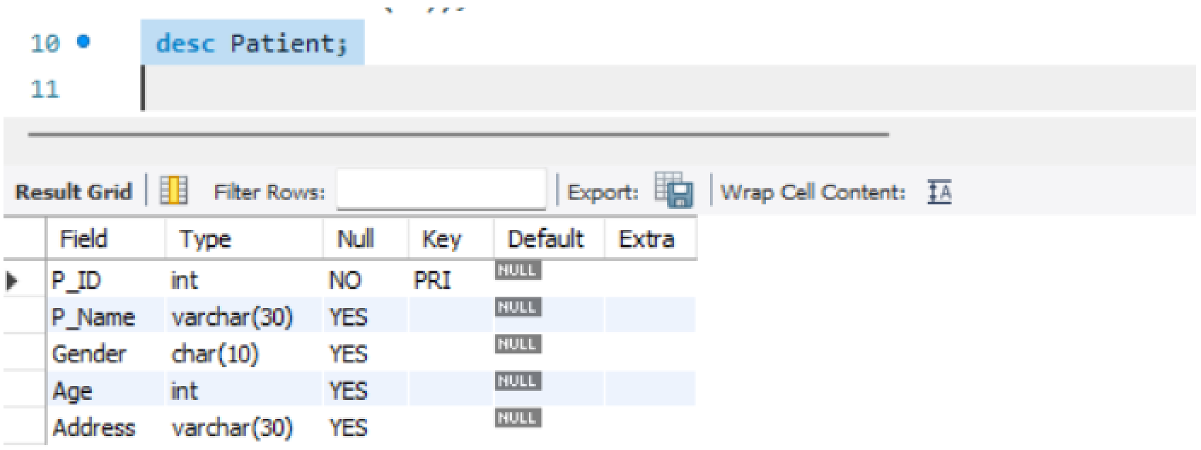
The medical Store Management System is a crucial tool for modernizing and optimizing the operations of a medical store, ultimately leading to improved customer satisfaction and business efficiency.

ER Diagram:


Consult
Patient
.
P_id (PK)
P_name
Address
Age
Gender
Prescribe
Doctor
.
Dr_id (PK)
Dr_name
Contact no.
P_id (FK)
Specialization
Visit
Bills
.
Bill_id
B_date
Payment_method
P_id
Pharmacist
.
Phar_id
Phar_name
P_id (FK)
Shift
Contact
Sell
Medicine
.
Med_id (PK)
Med_name
Quantity
Price
P_id (FK)
Exp_date
Supply
Vendor
.
V_id
V_name
Email_id
Med_id (FK)

Structure & Contents Of Tables:

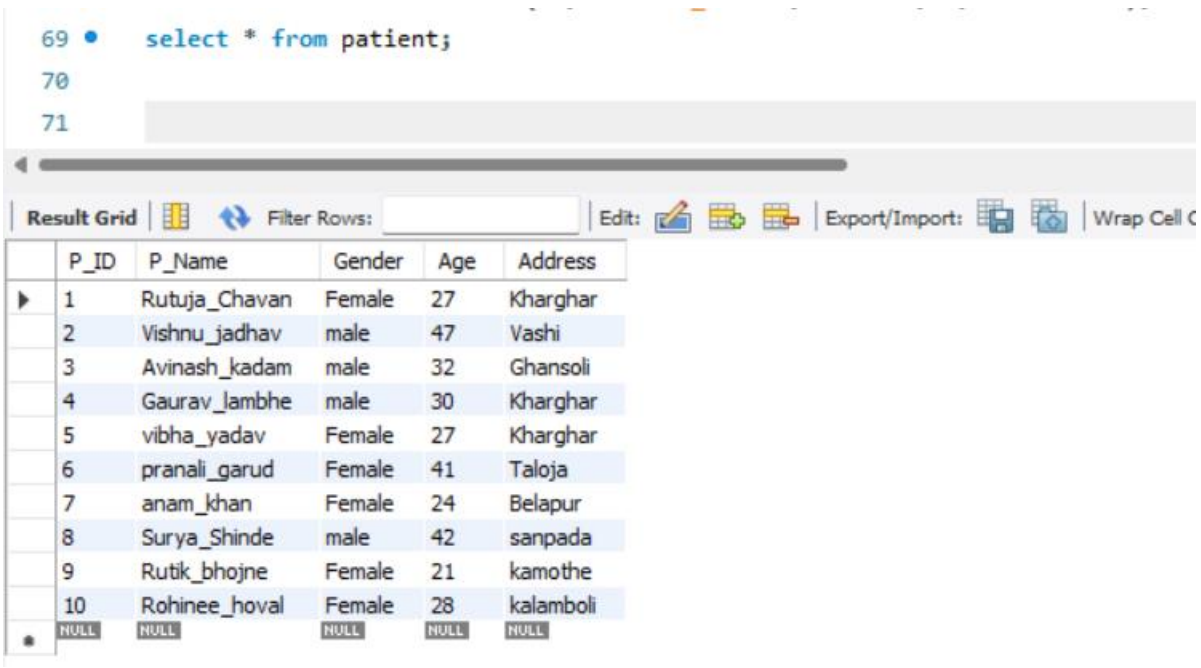
Patient -



```
desc Patient;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
P_ID	int	NO	PRI	NULL	
P_Name	varchar(30)	YES		NULL	
Gender	char(10)	YES		NULL	
Age	int	YES		NULL	
Address	varchar(30)	YES		NULL	

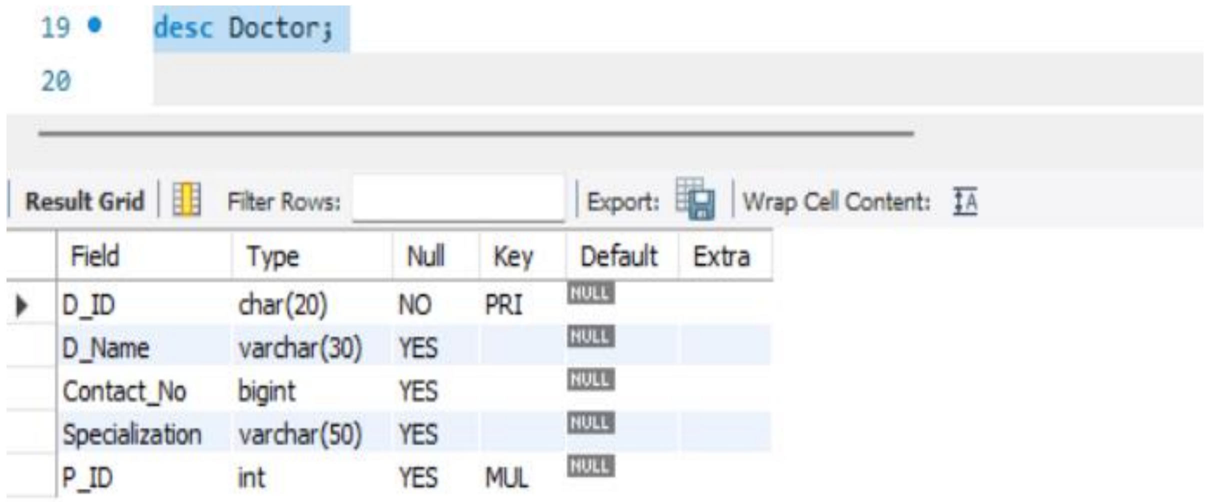


```
select * from patient;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell C

P_ID	P_Name	Gender	Age	Address
1	Rutuja_Chavan	Female	27	Kharghar
2	Vishnu_jadhav	male	47	Vashi
3	Avinash_kadam	male	32	Ghansoli
4	Gaurav_lambhe	male	30	Kharghar
5	vibha_yadav	Female	27	Kharghar
6	pranali_garud	Female	41	Taloja
7	anam_khan	Female	24	Belapur
8	Surya_Shinde	male	42	sanpada
9	Rutik_bhojne	Female	21	kamothe
10	Rohinee_hoval	Female	28	kalamboli
NULL	NULL	NULL	NULL	NULL

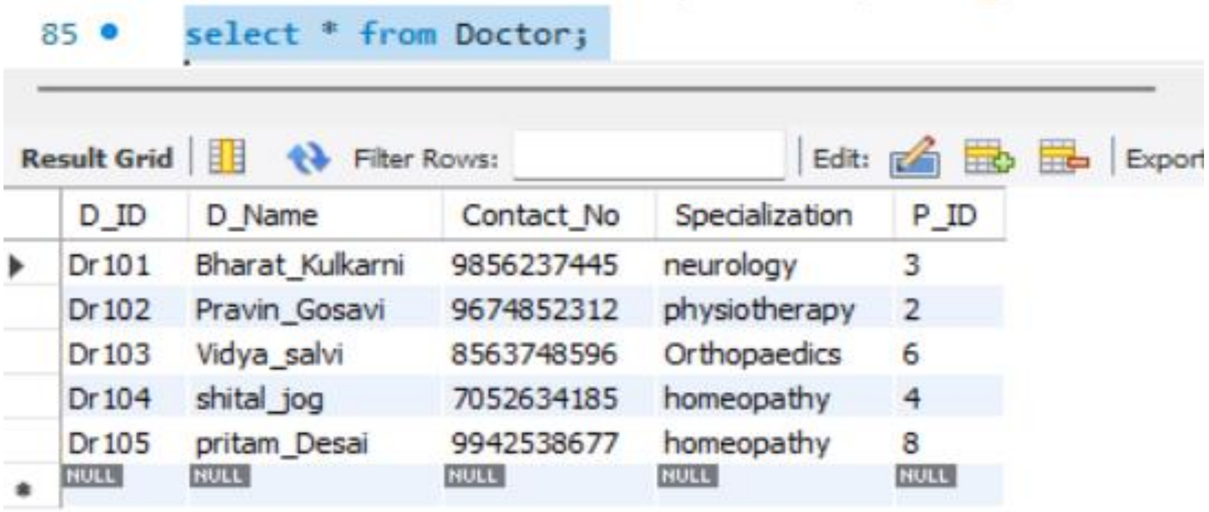
Doctor –



```
desc Doctor;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
D_ID	char(20)	NO	PRI	NULL	
D_Name	varchar(30)	YES		NULL	
Contact_No	bigint	YES		NULL	
Specialization	varchar(50)	YES		NULL	
P_ID	int	YES	MUL	NULL	

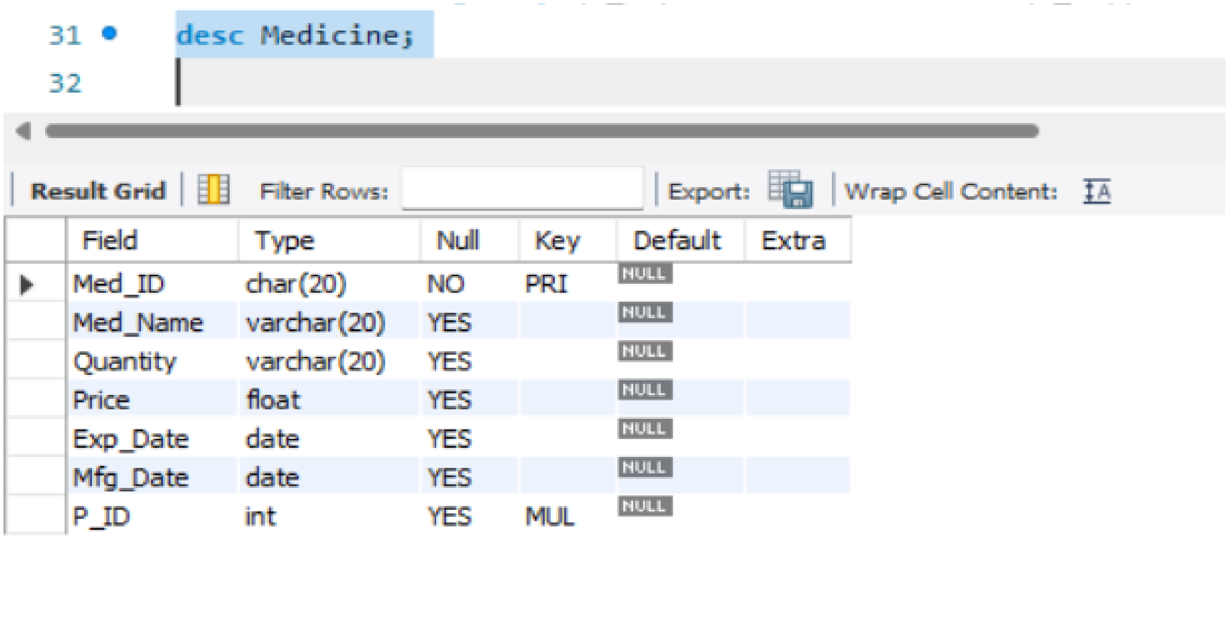


```
select * from Doctor;
```

Result Grid | Filter Rows: | Edit: | Export

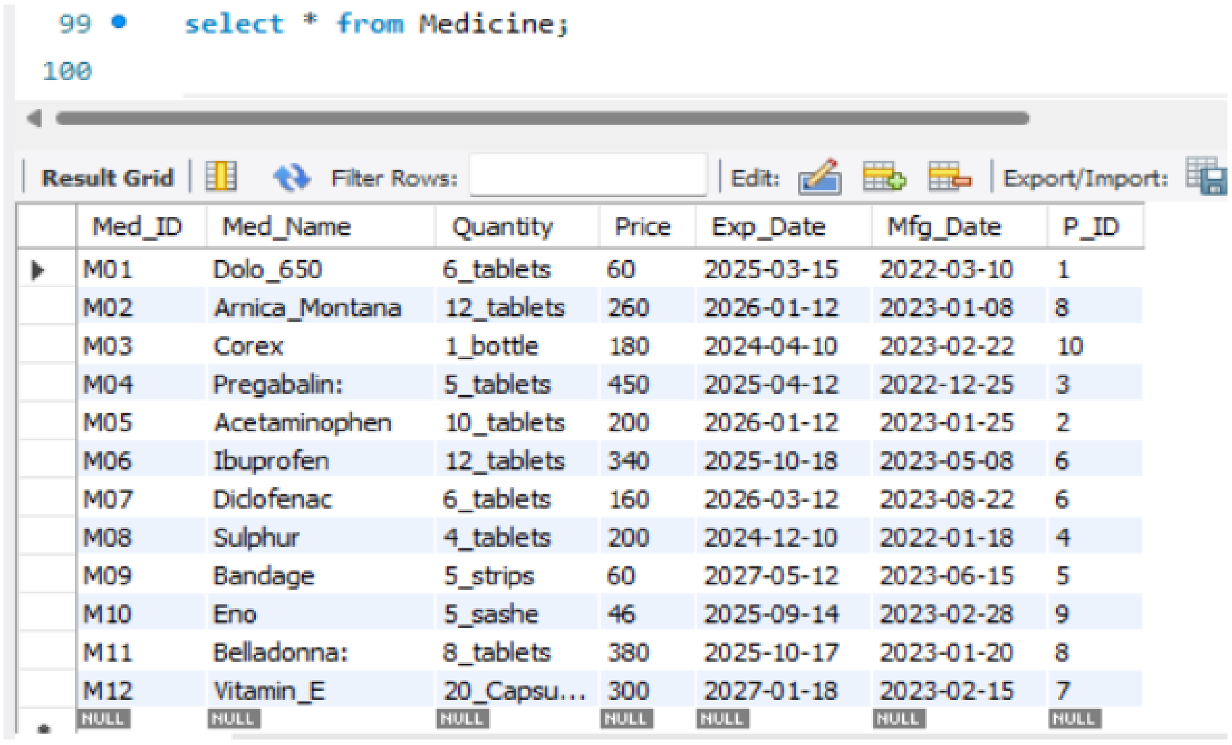
D_ID	D_Name	Contact_No	Specialization	P_ID
Dr101	Bharat_Kulkarni	9856237445	neurology	3
Dr102	Pravin_Gosavi	9674852312	physiotherapy	2
Dr103	Vidya_salvi	8563748596	Orthopaedics	6
Dr104	shital_jog	7052634185	homeopathy	4
Dr105	pritam_Desai	9942538677	homeopathy	8
NULL	NULL	NULL	NULL	NULL

Medicine -



```
desc Medicine;
```

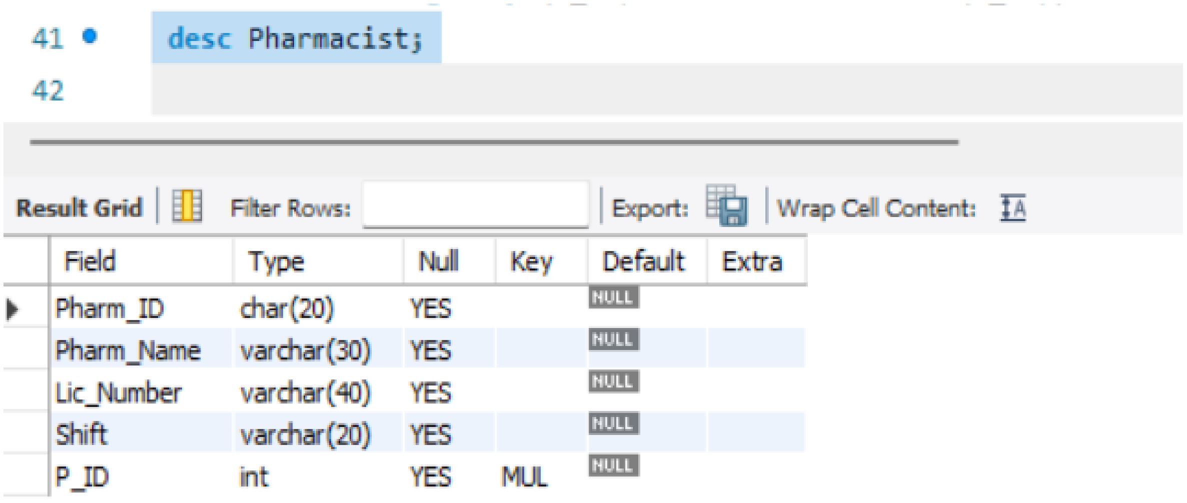
Field	Type	Null	Key	Default	Extra
Med_ID	char(20)	NO	PRI	NULL	
Med_Name	varchar(20)	YES		NULL	
Quantity	varchar(20)	YES		NULL	
Price	float	YES		NULL	
Exp_Date	date	YES		NULL	
Mfg_Date	date	YES		NULL	
P_ID	int	YES	MUL	NULL	



```
select * from Medicine;
```

Med_ID	Med_Name	Quantity	Price	Exp_Date	Mfg_Date	P_ID
M01	Dolo_650	6_tablets	60	2025-03-15	2022-03-10	1
M02	Arnica_Montana	12_tablets	260	2026-01-12	2023-01-08	8
M03	Corex	1_bottle	180	2024-04-10	2023-02-22	10
M04	Pregabalin:	5_tablets	450	2025-04-12	2022-12-25	3
M05	Acetaminophen	10_tablets	200	2026-01-12	2023-01-25	2
M06	Ibuprofen	12_tablets	340	2025-10-18	2023-05-08	6
M07	Diclofenac	6_tablets	160	2026-03-12	2023-08-22	6
M08	Sulphur	4_tablets	200	2024-12-10	2022-01-18	4
M09	Bandage	5_strips	60	2027-05-12	2023-06-15	5
M10	Eno	5_sashe	46	2025-09-14	2023-02-28	9
M11	Belladonna:	8_tablets	380	2025-10-17	2023-01-20	8
M12	Vitamin_E	20_Capsu...	300	2027-01-18	2023-02-15	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL

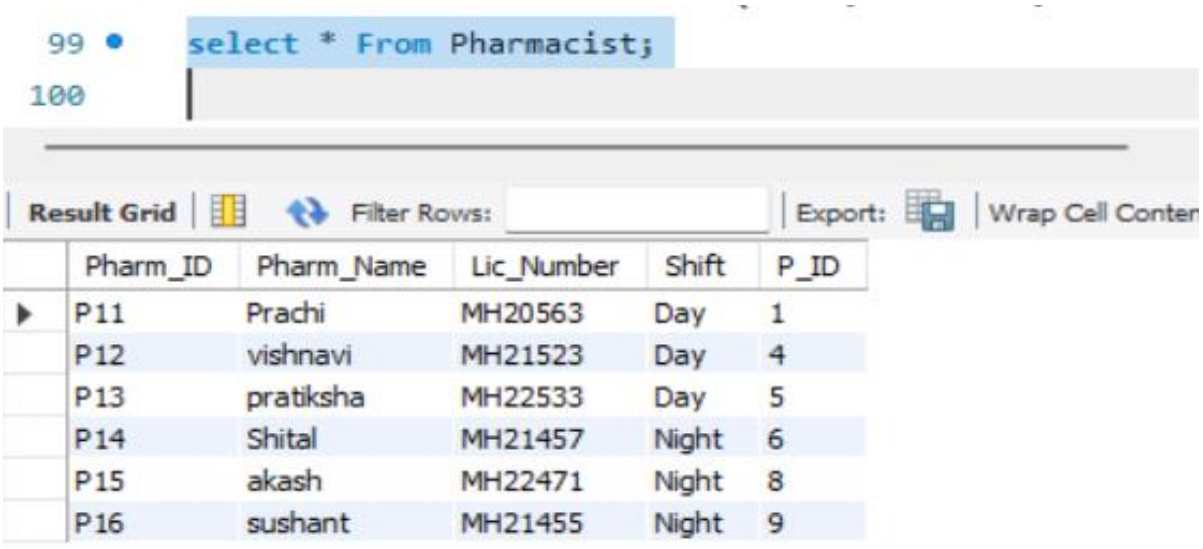
Pharmacist -



```
desc Pharmacist;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Pharm_ID	char(20)	YES		NULL	
Pharm_Name	varchar(30)	YES		NULL	
Lic_Number	varchar(40)	YES		NULL	
Shift	varchar(20)	YES		NULL	
P_ID	int	YES	MUL	NULL	

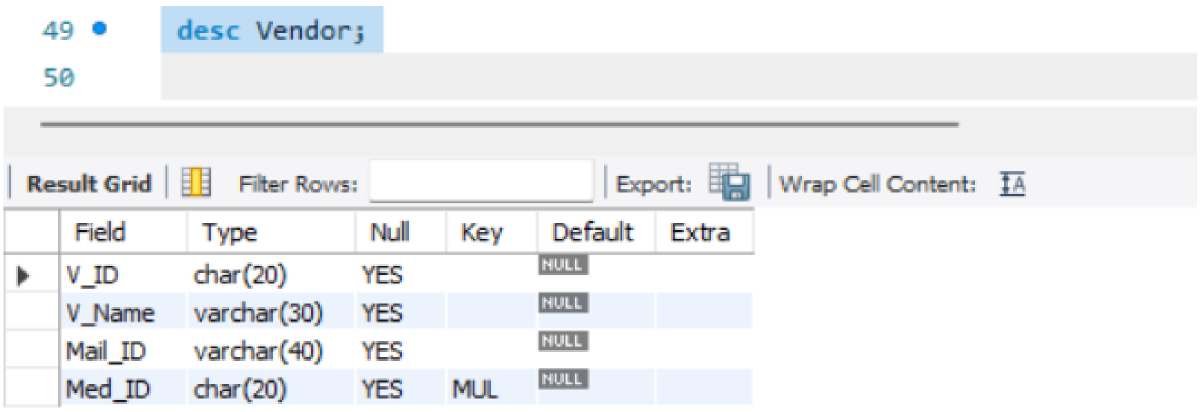


```
select * From Pharmacist;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conten

Pharm_ID	Pharm_Name	Lic_Number	Shift	P_ID
P11	Prachi	MH20563	Day	1
P12	vishnavi	MH21523	Day	4
P13	pratiksha	MH22533	Day	5
P14	Shital	MH21457	Night	6
P15	akash	MH22471	Night	8
P16	sushant	MH21455	Night	9

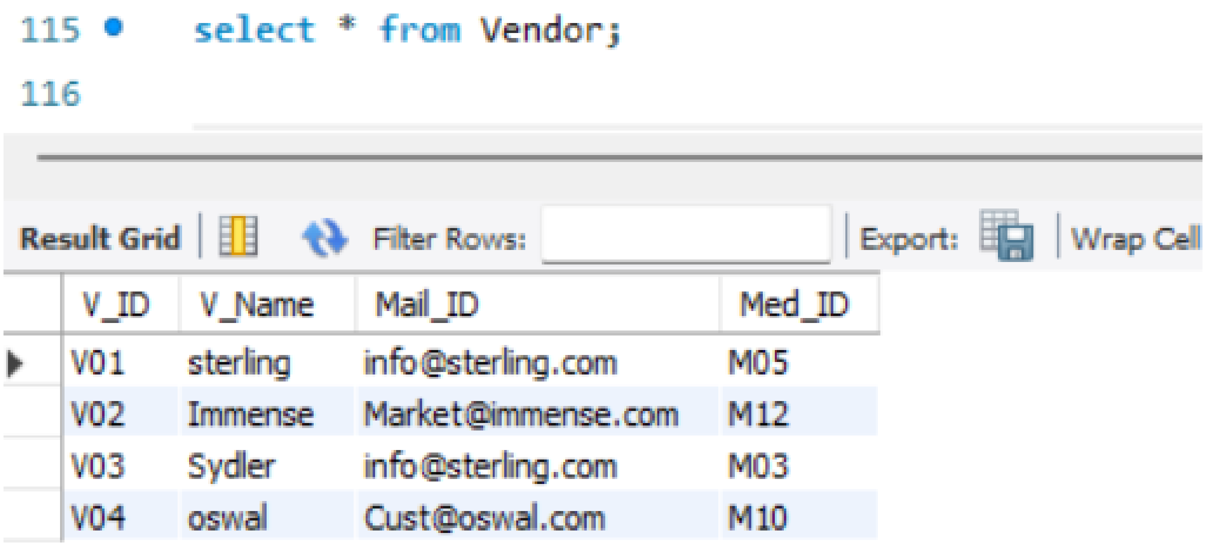
Vendor -



```
desc Vendor;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
V_ID	char(20)	YES		NULL	
V_Name	varchar(30)	YES		NULL	
Mail_ID	varchar(40)	YES		NULL	
Med_ID	char(20)	YES	MUL	NULL	

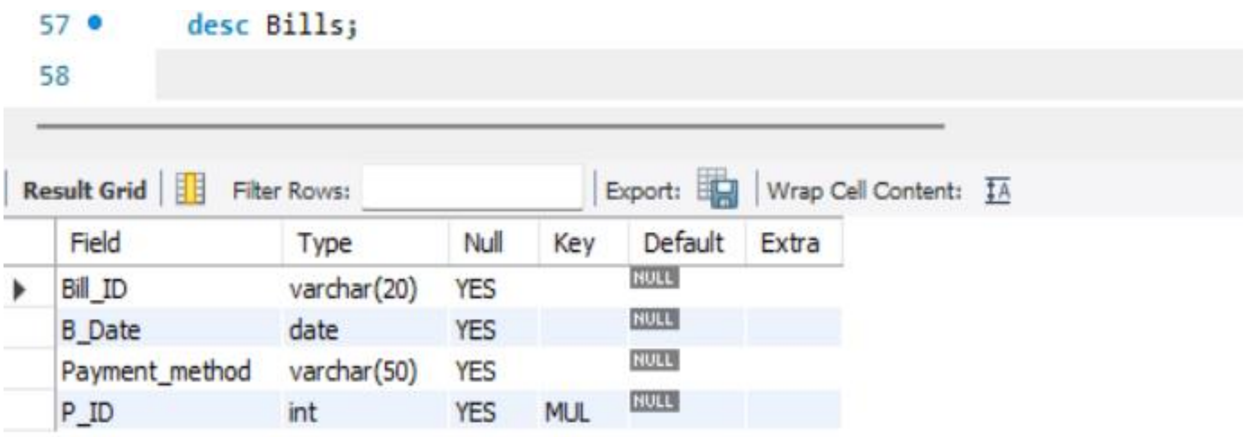


```
select * from Vendor;
```

Result Grid | Filter Rows: | Export: | Wrap Cell

V_ID	V_Name	Mail_ID	Med_ID
V01	sterling	info@sterling.com	M05
V02	Immense	Market@immense.com	M12
V03	Sydler	info@sterling.com	M03
V04	oswal	Cust@oswal.com	M10

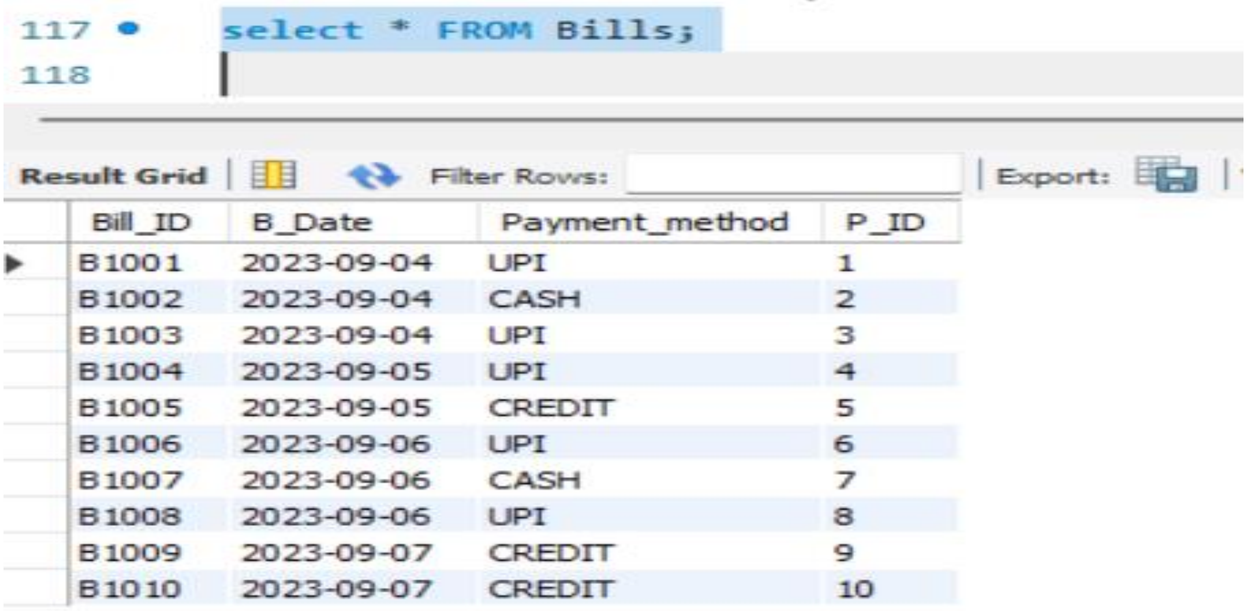
Bills-



```
desc Bills;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	Bill_ID	varchar(20)	YES		NULL	
	B_Date	date	YES		NULL	
	Payment_method	varchar(50)	YES		NULL	
	P_ID	int	YES	MUL	NULL	



```
select * FROM Bills;
```

Result Grid | Filter Rows: | Export:

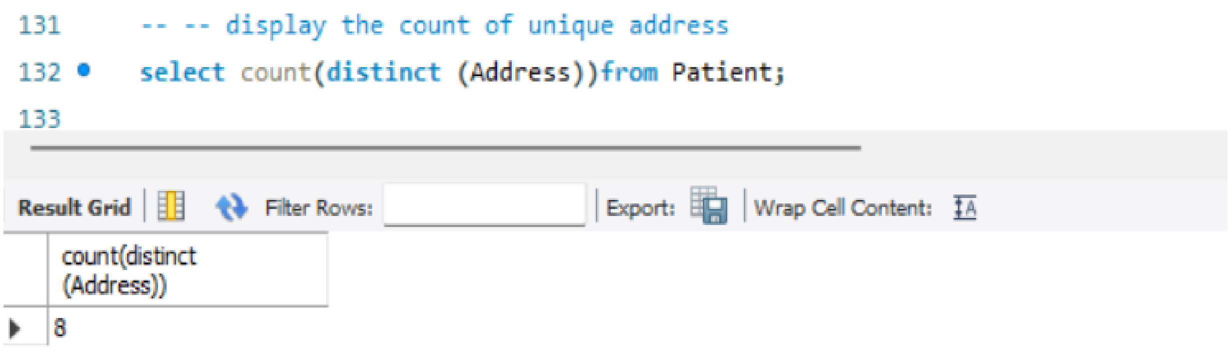
	Bill_ID	B_Date	Payment_method	P_ID
▶	B1001	2023-09-04	UPI	1
	B1002	2023-09-04	CASH	2
	B1003	2023-09-04	UPI	3
	B1004	2023-09-05	UPI	4
	B1005	2023-09-05	CREDIT	5
	B1006	2023-09-06	UPI	6
	B1007	2023-09-06	CASH	7
	B1008	2023-09-06	UPI	8
	B1009	2023-09-07	CREDIT	9
	B1010	2023-09-07	CREDIT	10

Que1. display the count of unique address

Query:

select count(distinct (Address))from Patient;

- DISTINCT to eliminate duplicate values from a column and useful for filtering out duplicate values



```
-- -- display the count of unique address
select count(distinct (Address))from Patient;

```

count(distinct (Address))
8

Que.2. display the medicine name whose medicine price between 300 to 500

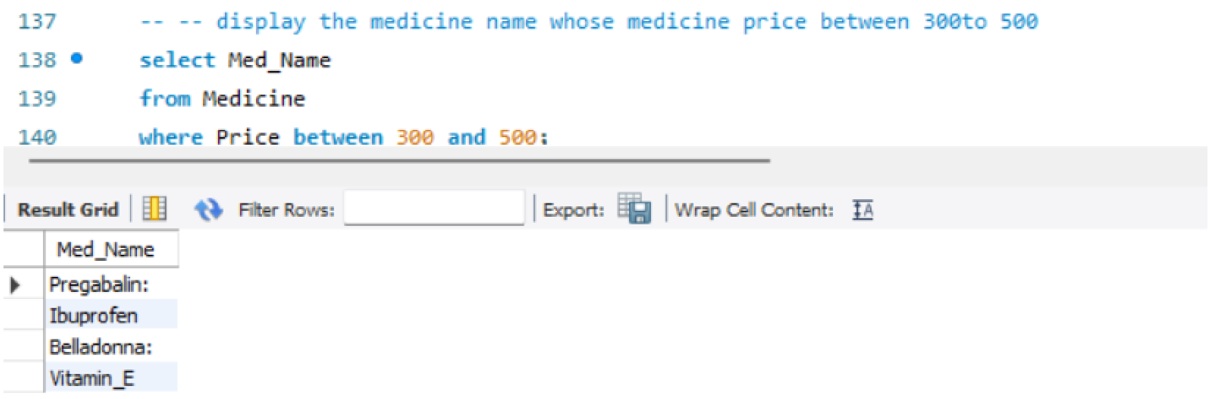
Query:

select Med_Name

from Medicine

where Price between 300 and 500;

- **WHERE clause with range operator (BETWEEN):** It can be used for range of dates, numeric values. Similar to AND condition but for single field.



```
-- -- display the medicine name whose medicine price between 300to 500
select Med_Name
from Medicine
where Price between 300 and 500;
```

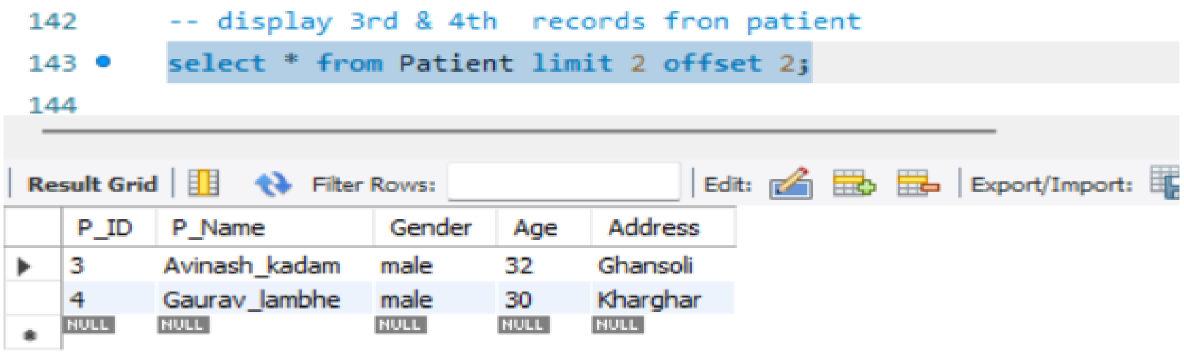
Med_Name
Pregabalin:
Ibuprofen
Belladonna:
Vitamin_E

Que.3. display 3rd & 4th records from patient

Query:

select * from Patient limit 2 offset 2;

- Limit and offset are clause used in conjunction with the select statement to control the number of rows returned by a query and skip a certain number of rows respectively.



```
-- display 3rd & 4th  records fron patient
select * from Patient limit 2 offset 2;
```

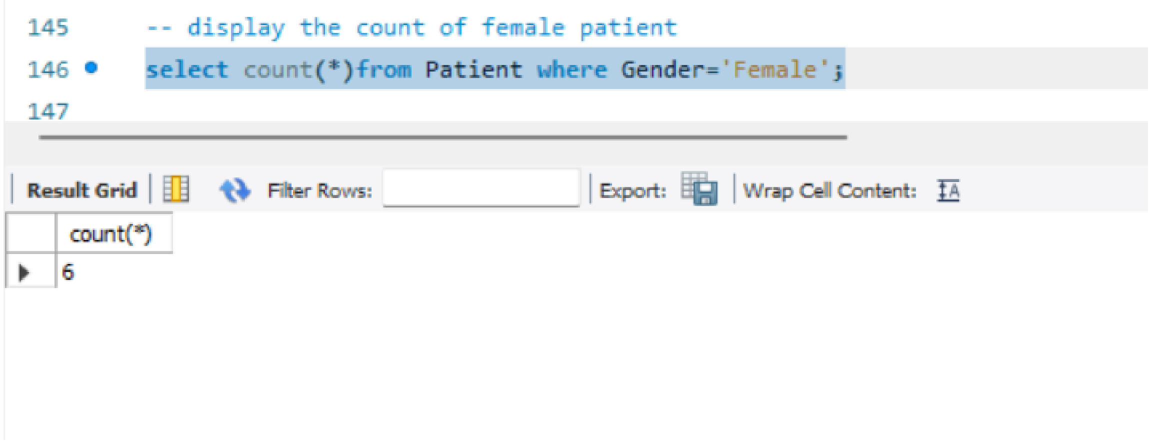
P_ID	P_Name	Gender	Age	Address
3	Avinash_kadam	male	32	Ghansoli
4	Gaurav_lambhe	male	30	Kharghar
NULL	NULL	NULL	NULL	NULL

Que.4. display the count of female patient

Query:

select count(*)from Patient where Gender='Female';

- Count query is a single value, which is the count of rows or values that match the specified criteria.



```
-- display the count of female patient
select count(*)from Patient where Gender='Female';
```

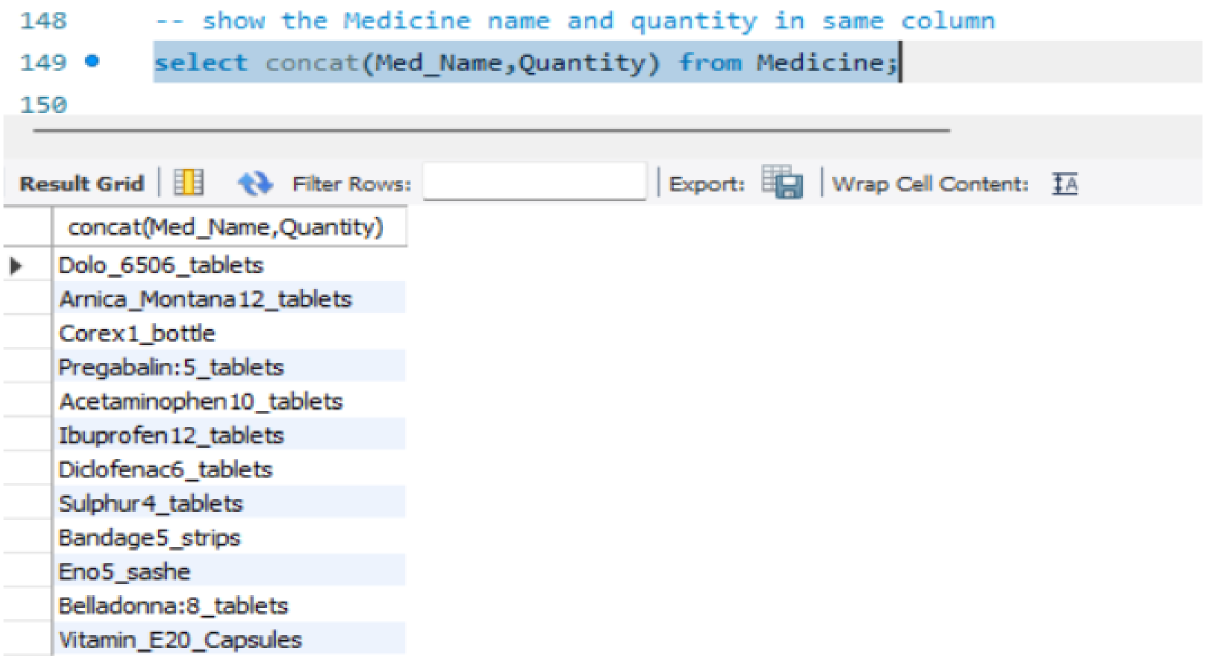
count(*)
6

Que.5. show the Medicine name and quantity in same column.

Query:

select concat(Med_Name,Quantity) from Medicine;

- concat function is used to combine two or more strings together into a single string.



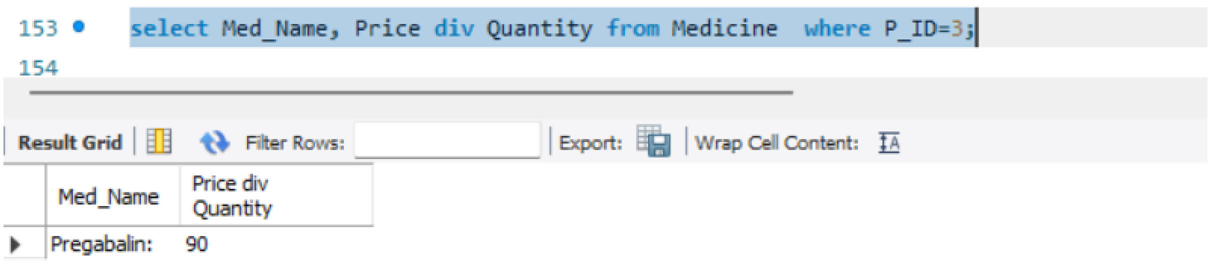
```
-- show the Medicine name and quantity in same column
select concat(Med_Name,Quantity) from Medicine;
```

concat(Med_Name,Quantity)
Dolo_6506_tablets
Arnica_Montana12_tablets
Corex1_bottle
Pregabalin:5_tablets
Acetaminophen10_tablets
Ibuprofen12_tablets
Diclofenac6_tablets
Sulphur4_tablets
Bandage5_strips
Eno5_sashe
Belladonna:8_tablets
Vitamin_E20_Capsules

Que.6.calculate the one pregabalin tablet price

Query:

select Med_Name, Price div Quantity from Medicine where P_ID=3;



```
select Med_Name, Price div Quantity from Medicine  where P_ID=3;
```

Med_Name	Price div Quantity
Pregabalin:	90

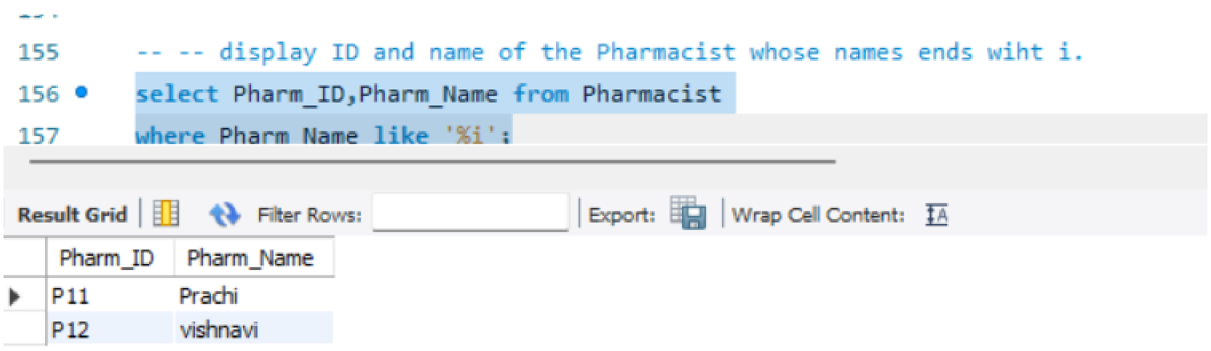
Que.7. display ID and name of the Pharmacist whose names ends with ‘I’

Query:

select Pharm_ID,Pharm_Name from Pharmacist

where Pharm_Name like '%i';

- **WHERE clause with LIKE operator:** It is used when we want to select rows to display that are similar to another field or constant. It is used with % (percentage) or _ (underscore) for character data types. % represent many, _ represent single character.
-



```
-- -- display ID and name of the Pharmacist whose names ends wiht i.
select Pharm_ID,Pharm_Name from Pharmacist
where Pharm_Name like '%i';
```

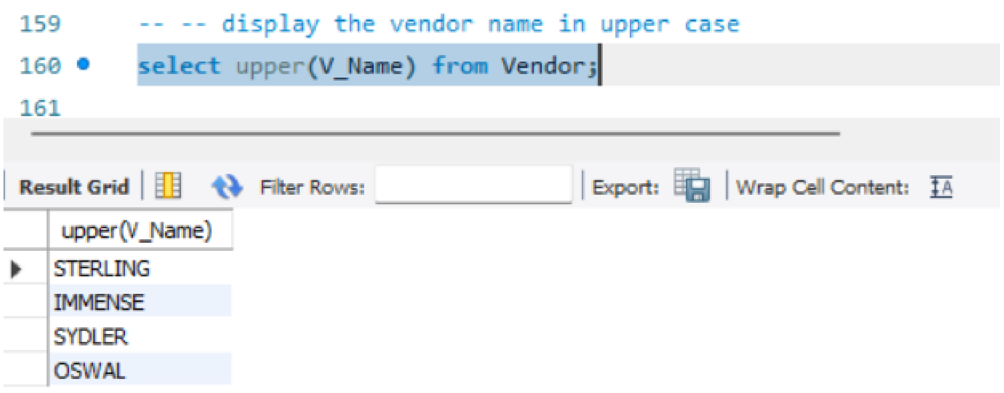
Pharm_ID	Pharm_Name
P11	Prachi
P12	vishnavi

Que.8. display the vendor name in upper case

Query

select upper(V_Name) from Vendor;

- **UPPER:** converts all the characters to Uppercase



```
-- -- display the vendor name in upper case
select upper(V_Name) from Vendor;
```

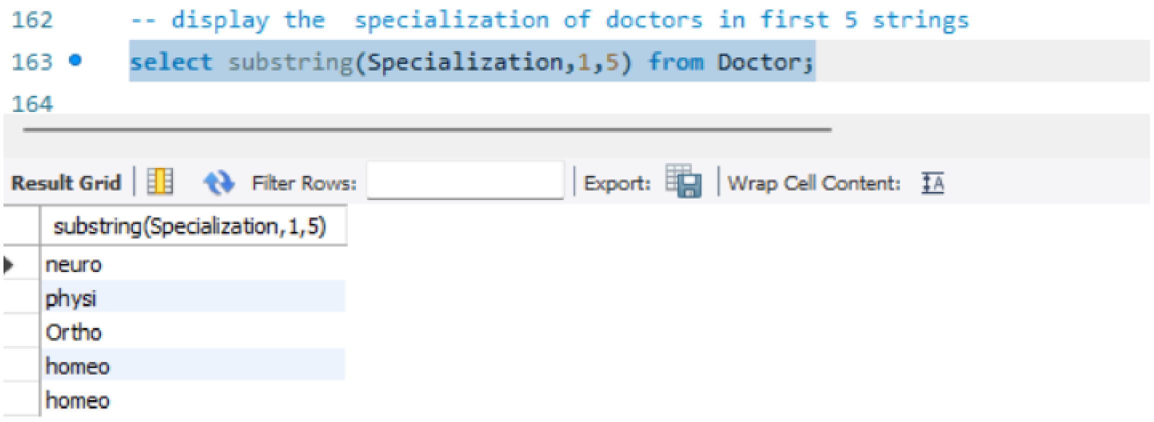
upper(V_Name)
STERLING
IMMENSE
SYDLER
OSWAL

Que.9. display the specialization of doctors in first 5 strings

Query:

select substring(Specialization,1,5) from Doctor;

- Substring is used to extrsct position of a string ,it allows to specify the position and length of substring to extract from given string.



```
-- display the  specialization of doctors in first 5 strings
select substring(Specialization,1,5) from Doctor;
```

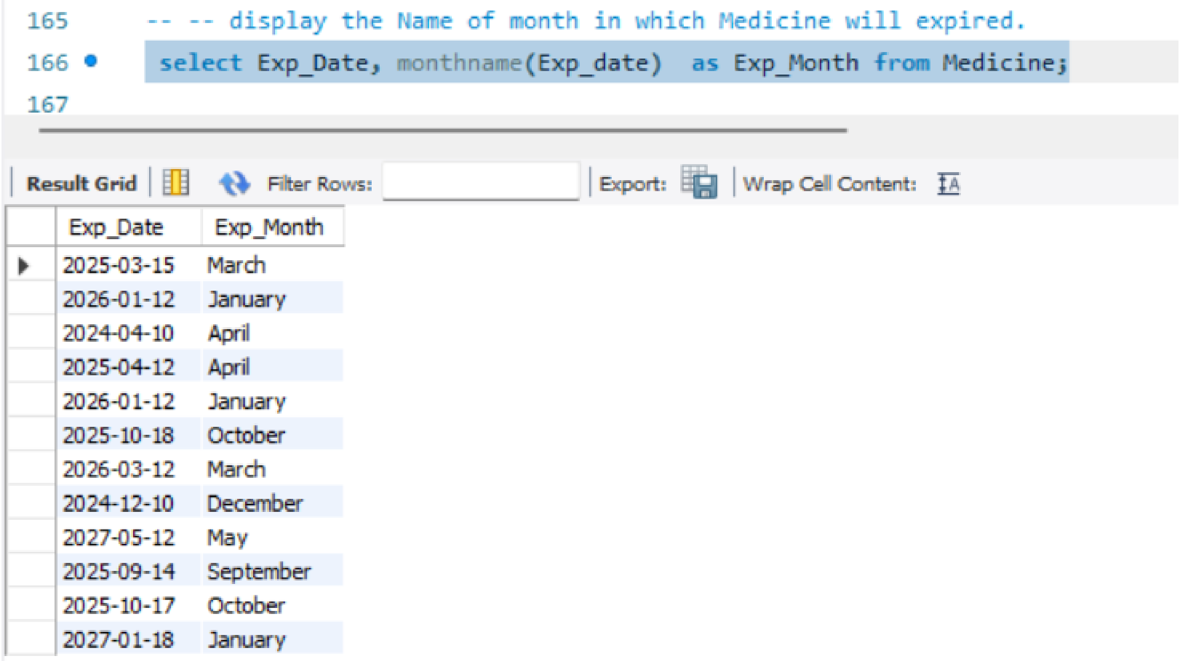
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

substring(Specialization,1,5)
neuro
physi
Ortho
homeo
homeo

Que.10. display the Name of month in which Medicine will expired.

Query: select Exp_Date, monthname(Exp_date) as Exp_Month from Medicine;

- Monthname function is used to extract name of month from date or timestamp.



```
-- -- display the Name of month in which Medicine will expired.
select Exp_Date, monthname(Exp_date)  as Exp_Month from Medicine;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

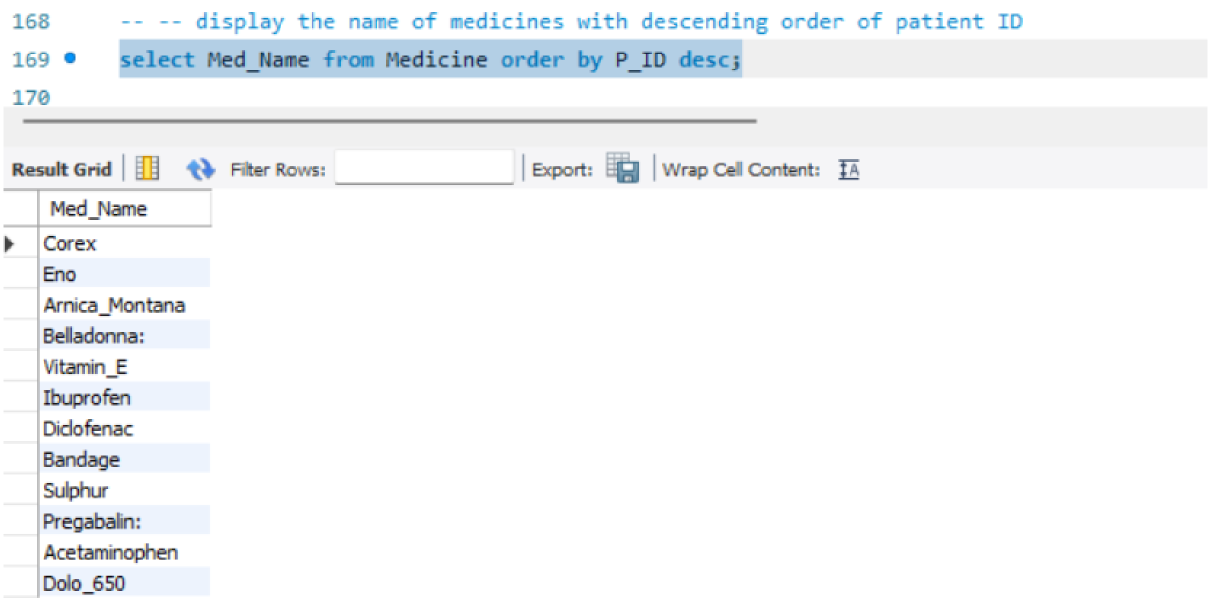
Exp_Date	Exp_Month
2025-03-15	March
2026-01-12	January
2024-04-10	April
2025-04-12	April
2026-01-12	January
2025-10-18	October
2026-03-12	March
2024-12-10	December
2027-05-12	May
2025-09-14	September
2025-10-17	October
2027-01-18	January

Que.11. display the name of medicines with descending order of patient ID

Query:

select Med_Name from Medicine order by P_ID desc;

- Order by clause is used to sort the result set of select statement based on one or more columns in ascending or descending order.



```
-- -- display the name of medicines with descending order of patient ID
select Med_Name from Medicine order by P_ID desc;
```

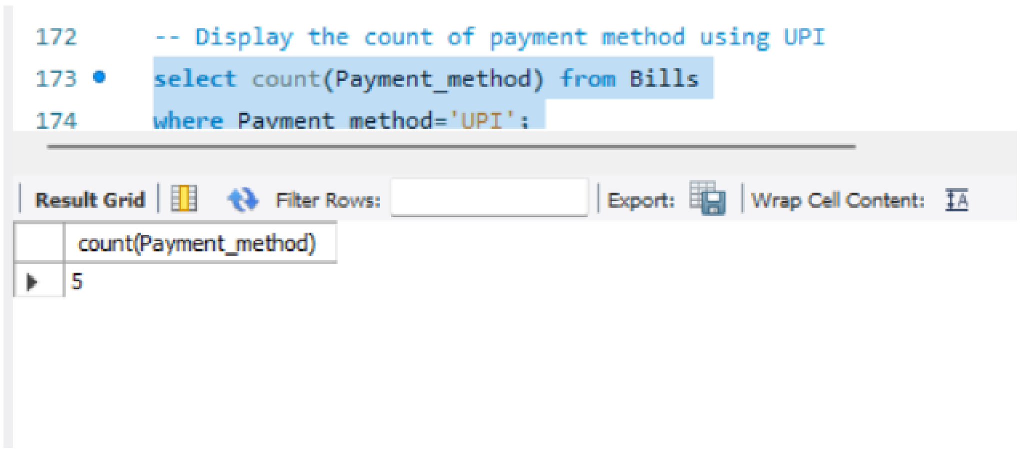
```
Med_Name
Corex
Eno
Arnica_Montana
Belladonna:
Vitamin_E
Ibuprofen
Diclofenac
Bandage
Sulphur
Pregabalin:
Acetaminophen
Dolo_650
```

Que.12. Display the count of payment method using UPI

Query:

select count(Payment_method) from Bills

where Payment_method='UPI';



```
-- Display the count of payment method using UPI
select count(Payment_method) from Bills
where Payment_method='UPI';
```

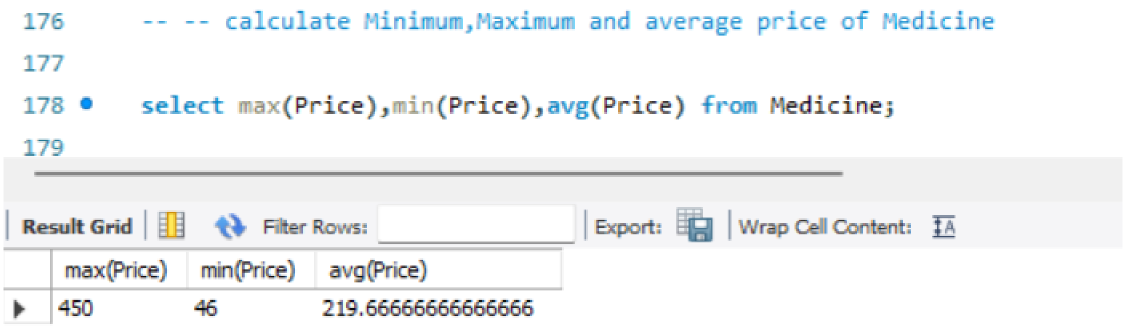
```
count(Payment_method)
5
```

Que.13. calculate Minimum,Maximum and average price of Medicine

Query:

select max(Price),min(Price),avg(Price) from Medicine;

- MIN,MAX AND AVG functions are used to perform calculations on numeric columns in a database table.



```
-- -- calculate Minimum,Maximum and average price of Medicine

select max(Price),min(Price),avg(Price) from Medicine;

```

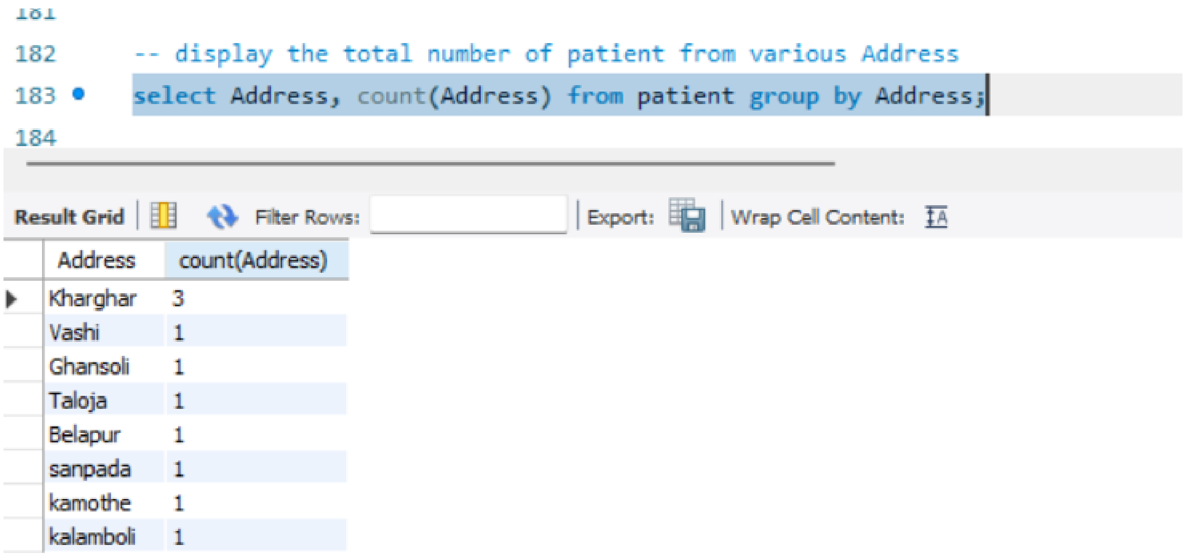
max(Price)	min(Price)	avg(Price)
450	46	219.66666666666666

Que.14. display the total number or count of patient from various Address

Query:

select Address, count(Address) from patient group by Address;

- Group by clausebis used to group rows with similar values in one or more columns into summary rows.



```

-- display the total number of patient from various Address
select Address, count(Address) from patient group by Address;

```

Address	count(Address)
Kharghar	3
Vashi	1
Ghansoli	1
Taloja	1
Belapur	1
sanpada	1
kamothe	1
kalamboli	1

Que.15.Display name medicine which are prescibe by doctor

Query: select M.Med_Name,M.P_ID,D.D_Name

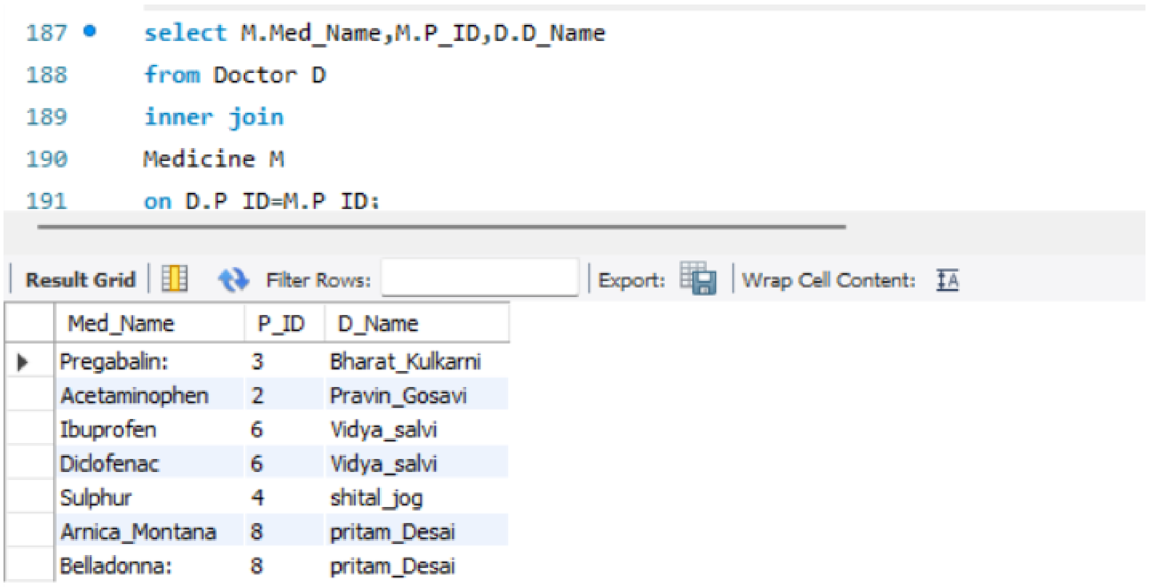
from Doctor D

inner join

Medicine M

on D.P_ID=M.P_ID;

- The INNER JOIN keyword selects records that have matching values in both tables.



```
select M.Med_Name,M.P_ID,D.D_Name
from Doctor D
inner join
Medicine M
on D.P_ID=M.P_ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Med_Name	P_ID	D_Name
Pregabalin:	3	Bharat_Kulkarni
Acetaminophen	2	Pravin_Gosavi
Ibuprofen	6	Vidya_salvi
Diclofenac	6	Vidya_salvi
Sulphur	4	shital_jog
Arnica_Montana	8	pritam_Desai
Belladonna:	8	pritam_Desai

Que.16.display name of patient who not prescibe by doctor

Query:

select P.P_ID,P.P_Name

from Patient P

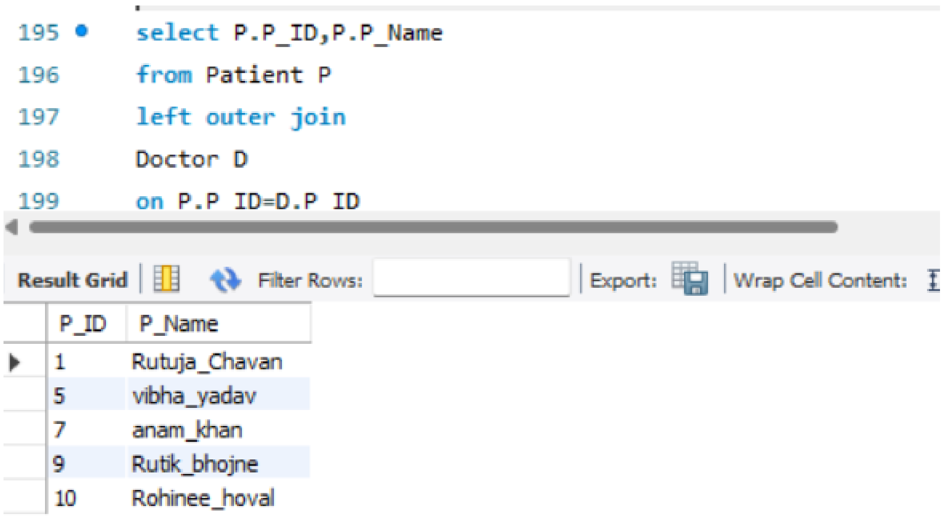
left outer join

Doctor D

on P.P_ID=D.P_ID

where D.P_ID is null;

- The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.



```
select P.P_ID,P.P_Name
from Patient P
left outer join
Doctor D
on P.P ID=D.P ID
```

P_ID	P_Name
1	Rutuja_Chavan
5	vibha_yadav
7	anam_khan
9	Rutik_bhojne
10	Rohinee_hoval

Que.17.display the name of medicines which are supplied by vendors

Query: select M.Med_ID,M.Med_Name

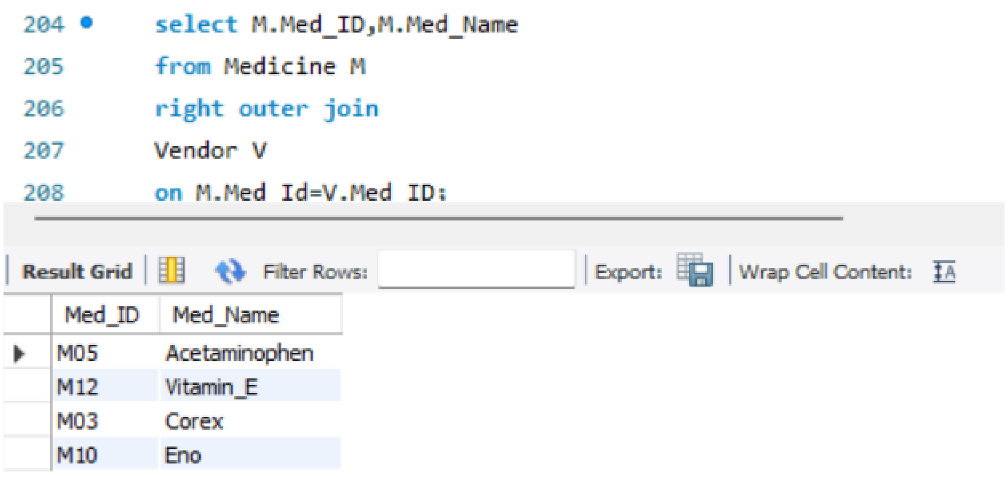
from Medicine M

right outer join

Vendor V

on M.Med_Id=V.Med_ID;

- The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.



```
select M.Med_ID,M.Med_Name
from Medicine M
right outer join
Vendor V
on M.Med_Id=V.Med_ID;
```

Med_ID	Med_Name
M05	Acetaminophen
M12	Vitamin_E
M03	Corex
M10	Eno

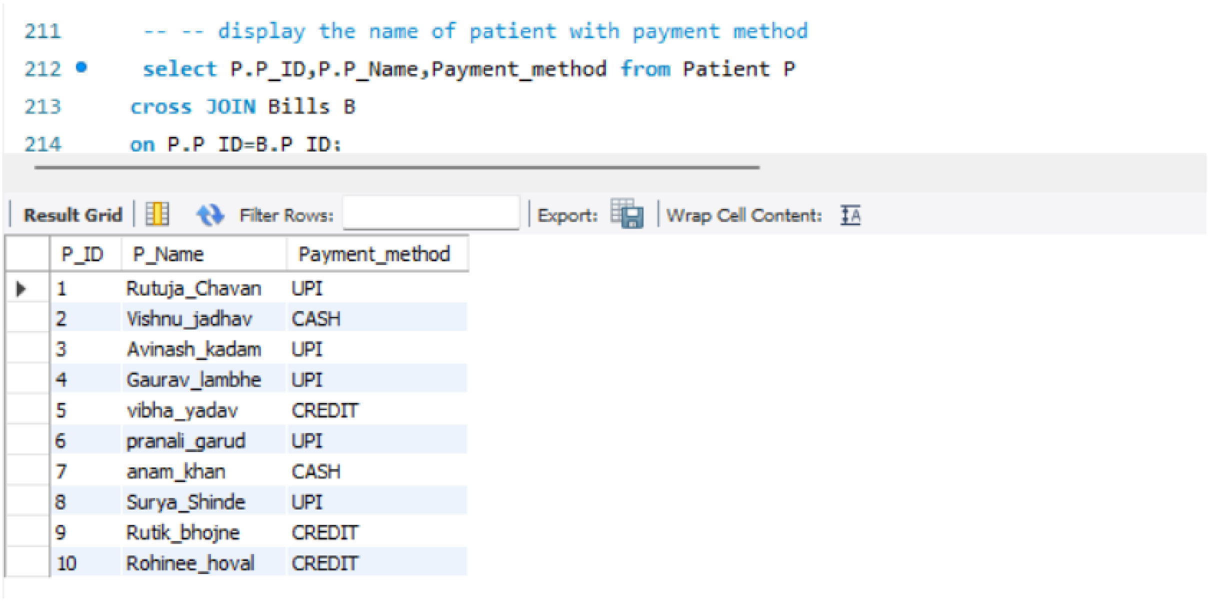
Que.18.display the name of patient with payment method

Query: select P.P_ID,P.P_Name,Payment_method from Patient P

cross JOIN Bills B

on P.P_ID=B.P_ID;

- The CROSS JOIN keyword returns all records from both tables (table1 and table2).



```
-- -- display the name of patient with payment method
select P.P_ID,P.P_Name,Payment_method from Patient P
cross JOIN Bills B
on P.P_ID=B.P_ID;
```

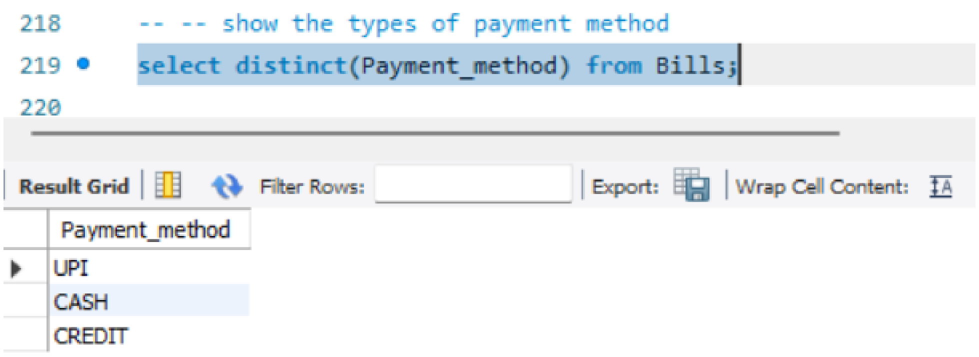
P_ID	P_Name	Payment_method
1	Rutuja_Chavan	UPI
2	Vishnu_jadhav	CASH
3	Avinash_kadam	UPI
4	Gaurav_lambhe	UPI
5	vibha_yadav	CREDIT
6	pranali_garud	UPI
7	anam_khan	CASH
8	Surya_Shinde	UPI
9	Rutik_bhojne	CREDIT
10	Rohinee_hoval	CREDIT

Que.19. show the types of payment method

Query:

select distinct(Payment_method) from Bills;

- The SELECT DISTINCT statement is used to return only distinct (different) values.



```
-- -- show the types of payment method
select distinct(Payment_method) from Bills;
```

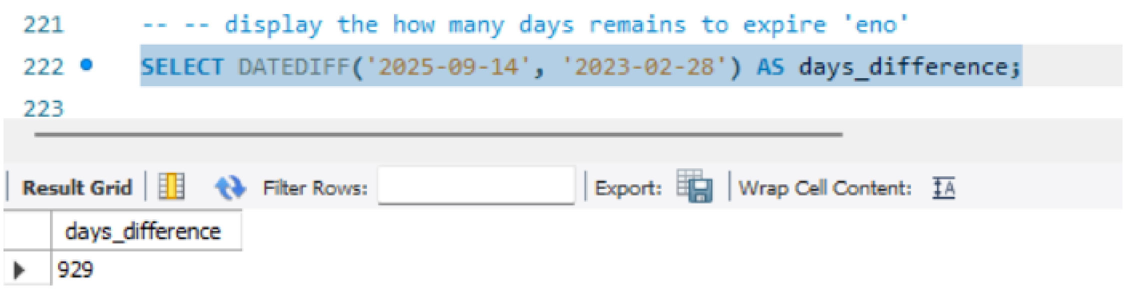
Payment_method
UPI
CASH
CREDIT

Que.20. display the how many days remains to expire 'eno'

Query:

SELECT DATEDIFF('2025-09-14', '2023-02-28') AS days_difference;

- The DATEDIFF() function returns the difference between two dates.



```
-- -- display the how many days remains to expire 'eno'
SELECT DATEDIFF('2025-09-14', '2023-02-28') AS days_difference;
```

days_difference
929

Que.21 display average price of medicine with medicine name whose price greater than 100

Query: SELECT Med_Name, AVG(Price) AS avg_Price

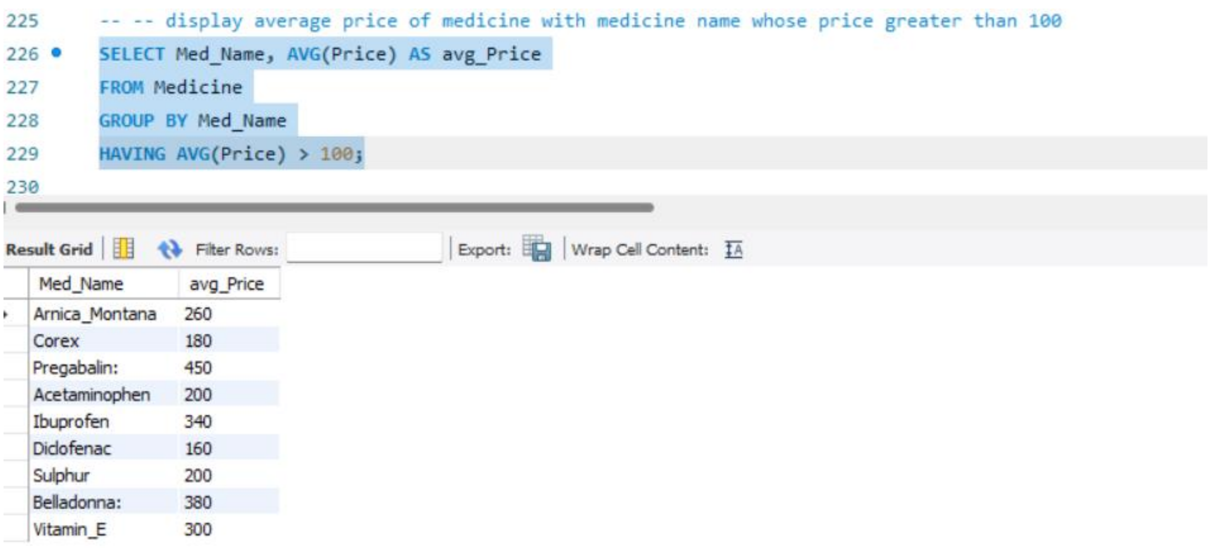
FROM Medicine

GROUP BY Med_Name

HAVING AVG(Price) > 100;

- The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".

- The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.



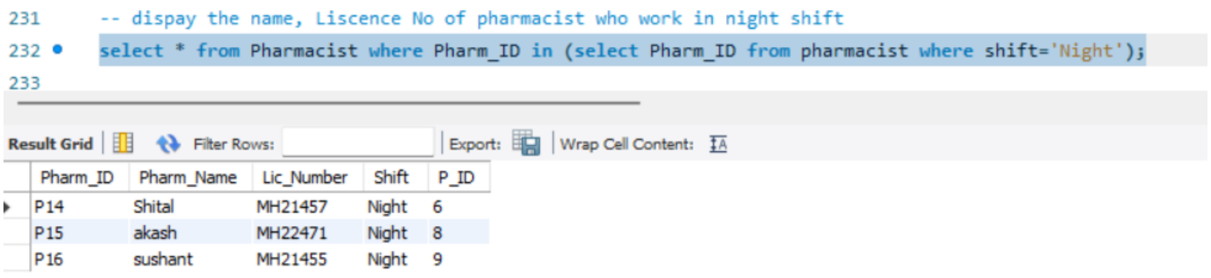
```
-- -- display average price of medicine with medicine name whose price greater than 100
SELECT Med_Name, AVG(Price) AS avg_Price
FROM Medicine
GROUP BY Med_Name
HAVING AVG(Price) > 100;
```

Med_Name	avg_Price
Arnica_Montana	260
Corex	180
Pregabalin:	450
Acetaminophen	200
Ibuprofen	340
Diclofenac	160
Sulphur	200
Belladonna:	380
Vitamin_E	300

Que.22 display the name, Liscence No of pharmacist who work in night shift

Query:

select * from Pharmacist where Pharm_ID in (select Pharm_ID from pharmacist where shift='Night');



```
-- dispay the name, Liscence No of pharmacist who work in night shift
select * from Pharmacist where Pharm_ID in (select Pharm_ID from pharmacist where shift='Night');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

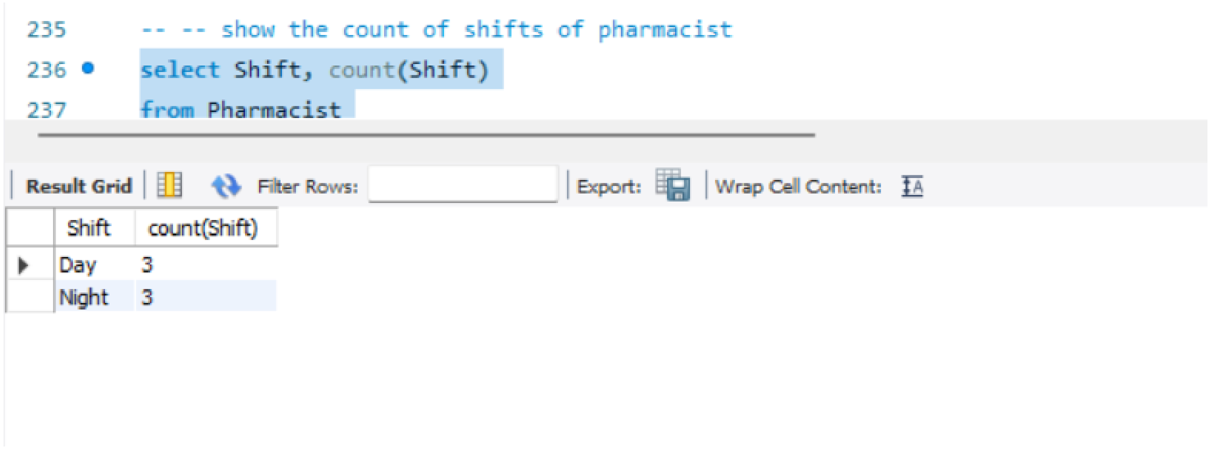
Pharm_ID	Pharm_Name	Lic_Number	Shift	P_ID
P14	Shital	MH21457	Night	6
P15	akash	MH22471	Night	8
P16	sushant	MH21455	Night	9

Que.23.show the count of shifts of pharmacist

Query: select Shift, count(Shift)

from Pharmacist

group by Shift having count(Shift);



```
-- -- show the count of shifts of pharmacist
select Shift, count(Shift)
from Pharmacist
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Shift	count(Shift)
Day	3
Night	3

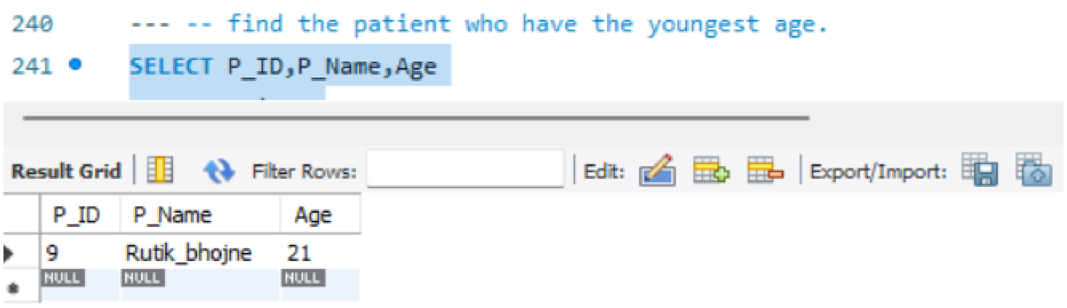
Que.24. find the patient who have the youngest age.

Query:

SELECT P_ID,P_Name,Age

FROM Patient

WHERE Age = (SELECT MIN(Age) FROM Patient);



```
--- -- find the patient who have the youngest age.
SELECT P_ID,P_Name,Age
```

	P_ID	P_Name	Age
▶	9	Rutik_bhojne	21
*	NULL	NULL	NULL

Que.25. retrieve a list of medicines and the doctors who prescribe them

Query:

SELECT M.Med_Name,D.D_Name

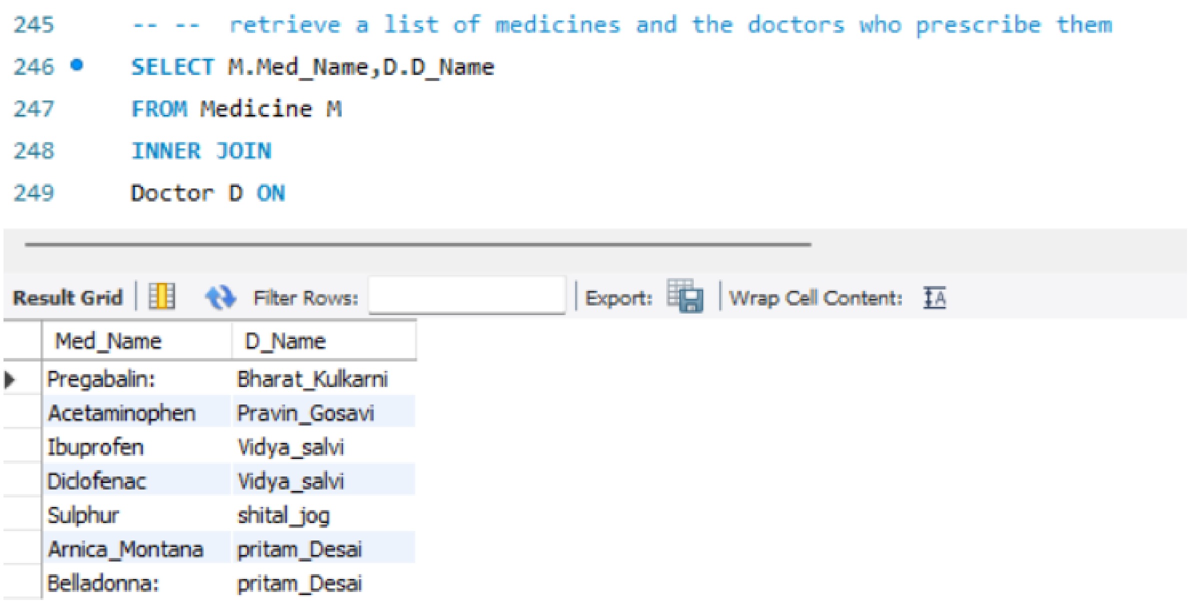
FROM Medicine M

INNER JOIN

Doctor D ON

M.P_ID = D.P_ID;

- The INNER JOIN keyword selects records that have matching values in both tables.


```
-- --  retrieve a list of medicines and the doctors who prescribe them
SELECT M.Med_Name,D.D_Name
FROM Medicine M
INNER JOIN
Doctor D ON
```

Med_Name	D_Name
Pregabalin:	Bharat_Kulkarni
Acetaminophen	Pravin_Gosavi
Ibuprofen	Vidya_salvi
Diclofenac	Vidya_salvi
Sulphur	shital_jog
Arnica_Montana	pritam_Desai
Belladonna:	pritam_Desai

Que.26. write a query to retrieves all medicines and the pharmacists who dispense them.

Query:

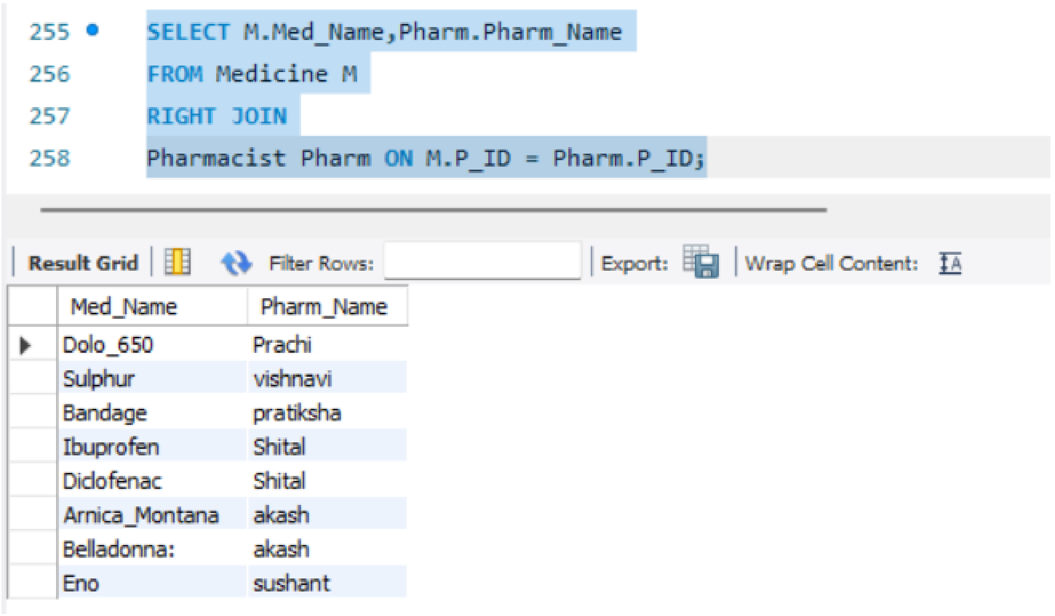
SELECT M.Med_Name,Pharm.Pharm_Name

FROM Medicine M

RIGHT JOIN

Pharmacist Pharm ON M.P_ID = Pharm.P_ID;

- The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match.


```
SELECT M.Med_Name,Pharm.Pharm_Name
FROM Medicine M
RIGHT JOIN
Pharmacist Pharm ON M.P_ID = Pharm.P_ID;
```

Med_Name	Pharm_Name
Dolo_650	Prachi
Sulphur	vishnavi
Bandage	pratiksha
Ibuprofen	Shital
Diclofenac	Shital
Arnica_Montana	akash
Belladonna:	akash
Eno	sushant

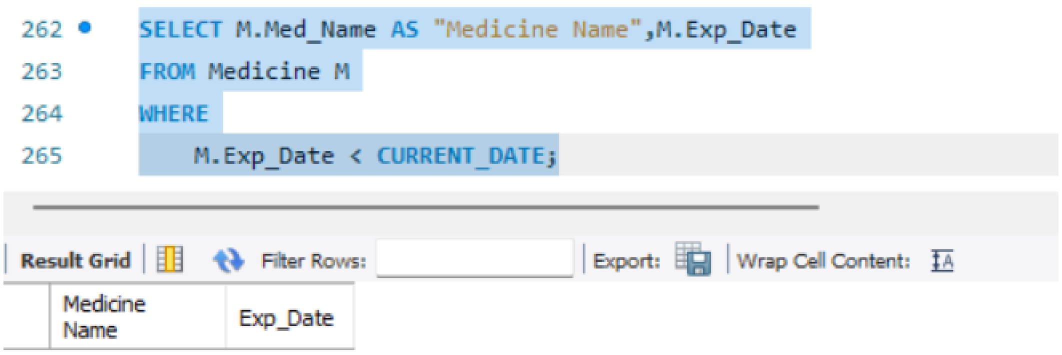
Que.27. write a query will retrieve medicines with expired expiry dates:

Query:

SELECT M.Med_Name AS "Medicine Name",M.Exp_Date

FROM Medicine M WHERE

M.Exp_Date < CURRENT_DATE;



```
SELECT M.Med_Name AS "Medicine Name",M.Exp_Date
FROM Medicine M
WHERE
    M.Exp_Date < CURRENT_DATE;
```

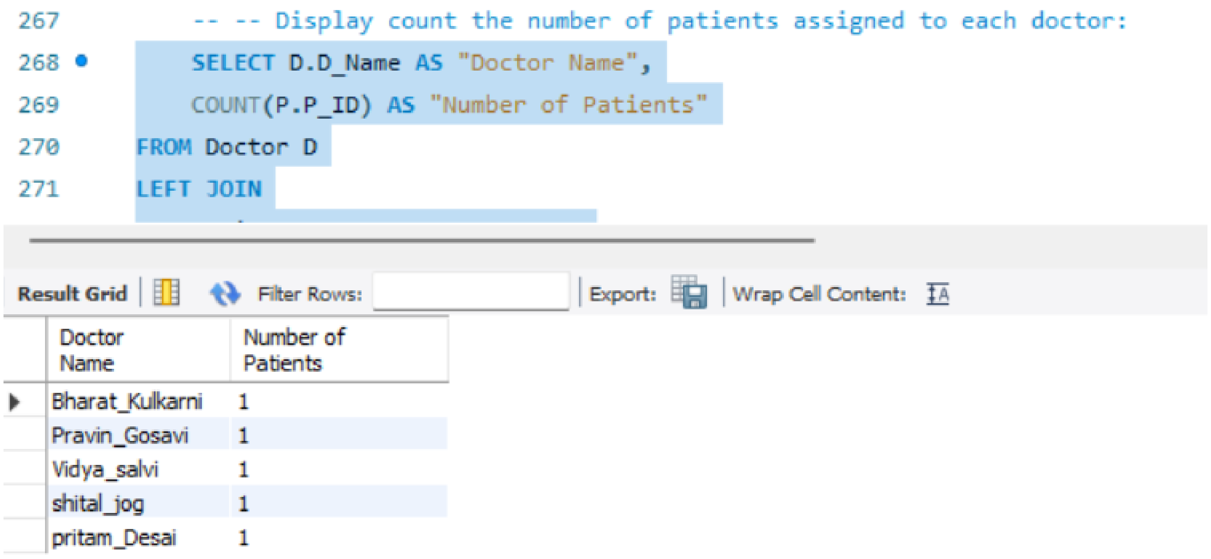
Medicine Name	Exp_Date

Que.28. Display count the number of patients assigned to each doctor:

Query:

```
  SELECT D.D_Name AS "Doctor Name",
   COUNT(P.P_ID) AS "Number of Patients"
FROM Doctor D
LEFT JOIN
   Patient P ON D.P_ID = P.P_ID
GROUP BY D.D_Name;
```

- The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.



```
        -- -- Display count the number of patients assigned to each doctor:
        SELECT D.D_Name AS "Doctor Name",
        COUNT(P.P_ID) AS "Number of Patients"
    FROM Doctor D
    LEFT JOIN
```

Doctor Name	Number of Patients
Bharat_Kulkarni	1
Pravin_Gosavi	1
Vidya_salvi	1
shital_jog	1
pritam_Desai	1