



PES UNIVERSITY

DATABASE MANAGEMENT SYSTEM

PHARMACY MANAGEMENT SYSTEM

BASANAGOUDA S HADIMANI-PES2UG19CS082

BHAGYASHREE SHANKAR-PES2UG19CS085

BHAVANA R-PES2UG19CS089

SIMPLE QUERIES

SQL Shell (psql)

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (14.0)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.
```

```
postgres=# \c pharmacy
You are now connected to database "pharmacy" as user "postgres".
pharmacy=# \l
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
company	postgres	UTF8	English_India.1252	English_India.1252	
pharmacy	postgres	UTF8	English_India.1252	English_India.1252	
postgres	postgres	UTF8	English_India.1252	English_India.1252	
template0	postgres	UTF8	English_India.1252	English_India.1252	=c/postgres +
					postgres=CtC/postgres
template1	postgres	UTF8	English_India.1252	English_India.1252	=c/postgres +
					postgres=CtC/postgres

(5 rows)

```
pharmacy=# \l pharmacy
```

List of databases					
Name	Owner	Encoding	Collate	Ctype	Access privileges
pharmacy	postgres	UTF8	English_India.1252	English_India.1252	


(1 row)

```
pharmacy=# \d
```

List of relations			
Schema	Name	Type	Owner
public	company	table	postgres
public	drugs	table	postgres
public	expiry	table	postgres
public	history_sales	table	postgres
public	inbox	table	postgres
public	login	table	postgres
public	message_history	table	postgres
public	purchase	table	postgres
public	sale	table	postgres
public	users	table	postgres

1) SELECT *FROM COMPANY;

SELECT *FROM COMPANY WHERE PHONE LIKE '8%';

 SQL Shell (psql)

```
pharmacy=# select *from company;
```

name	address	phone
Cipla	Mumbai	8523452596
Sun Pharma	Mysore	8971199805
Med_City	Nellore	6669923589
Sanofi	mumbai	6894512397
Johnson & Johnson	banglore	8645321798
Pfizer	hyderabad	9456312878
Abbott	kolkata	8631597234
Glenmark	delhi	7539518264

(8 rows)

```
pharmacy=# SELECT * FROM COMPANY WHERE PHONE LIKE '8%';
```

name	address	phone
Cipla	Mumbai	8523452596
Sun Pharma	Mysore	8971199805
Johnson & Johnson	banglore	8645321798
Abbott	kolkata	8631597234

(4 rows)

2)SELECT *FROM USERS;

UPDATE USERS SET SALARY=15000 WHERE ID=5;

SQL Shell (psql)

```
pharmacy=# select *from users;
 id | name | dob | address | phone | salary | password
-----+-----+-----+-----+-----+-----+-----
 1 | admin | 23-12-1995 | Bangalore | 9800000000 | 50000 | admin
 2 | mark | 3-2-1972 | Bangalore | 9632587418 | 2000 | mark
 3 | clark | 3-2-1971 | Richmond circle | 8521479633 | 4000 | rootaccess
 4 | arya | 7-8-1977 | Bangalore | 6478932143 | 3000 | rootaccess
 5 | bhavana | 3-4-2002 | Bangalore | 9980589711 | 10000 | rootaccess
 6 | basana | 12-07-2001 | Belgaum | 6547893218 | 7000 | rootaccess
 7 | bhagya | 06-06-2001 | Bangalore | 8265479313 | 9000 | rootaccess
 8 | Tom | 09-06-2001 | mumbai | 8971199805 | 10000 | rootaccess
(8 rows)
```

```
pharmacy=# UPDATE USERS SET SALARY = 15000 WHERE ID = 5;
UPDATE 1
pharmacy=# select *from users;
 id | name | dob | address | phone | salary | password
-----+-----+-----+-----+-----+-----+-----
 1 | admin | 23-12-1995 | Bangalore | 9800000000 | 50000 | admin
 2 | mark | 3-2-1972 | Bangalore | 9632587418 | 2000 | mark
 3 | clark | 3-2-1971 | Richmond circle | 8521479633 | 4000 | rootaccess
 4 | arya | 7-8-1977 | Bangalore | 6478932143 | 3000 | rootaccess
 6 | basana | 12-07-2001 | Belgaum | 6547893218 | 7000 | rootaccess
 7 | bhagya | 06-06-2001 | Bangalore | 8265479313 | 9000 | rootaccess
 8 | Tom | 09-06-2001 | mumbai | 8971199805 | 10000 | rootaccess
 5 | bhavana | 3-4-2002 | Bangalore | 9980589711 | 15000 | rootaccess
(8 rows)
```

pharmacy=#

3)SELECT *FROM DRUGS;

SELECT *FROM DRUGS ORDER BY COST_PRICE ASC;

SQL Shell (psql)

```
pharmacy=# select *from drugs;
 name | type | barcode | dose | code | cost_price | selling_price | expiry | company_name | production_date | expiration_date | place | quantity
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
Novalo | Bills | fsdgdjfhjorodsf | normal | 3d00 | 2 | 3 | Available for use | Cipla | 2021-03-03 | 2023-03-03 | N-Right | 40
novafol | Bills | ftrkl432432md | normal | 2xaa | 33 | 40 | Available for use | Sun Pharma | 2020-01-01 | 2022-01-01 | N-Left | 27
Abacavir | infection | fbankjiicfnl56 | normal | bhab | 44 | 50 | Available for use | Med_City | 2019-06-09 | 2022-04-03 | 262 DannyThomasPlace | 100
Amifostine | Anxiety | fnjldhicfjaipsk | normal | 2cf5 | 39 | 55 | Available for use | Sanofi | 2021-03-05 | 2023-05-07 | 262 DannyThomasPlace | 90
Calcium | mineral | fois54hiusdh | normal | sd3d | 69 | 80 | Available for use | Johnson & Johnson | 2020-08-08 | 2025-11-18 | N-Left | 150
Clindamycin | skin | fljhsfc35bsyc | normal | sbcr | 50 | 60 | Available for use | Pfizer | 2021-12-12 | 2022-11-15 | N-Left | 80
Dopamine | Lbp | fkjoifjp5sjlhvu | normal | 52kh | 100 | 150 | Available for use | Abbott | 2020-09-22 | 2025-09-30 | N-right | 200
Morphine | Painkiller | fkjhsido54zsc | normal | fs25 | 250 | 270 | Available for use | Glenmark | 2021-05-25 | 2024-12-09 | 262 DannyThomasPlace | 300
(8 rows)
```

```
pharmacy=# SELECT * FROM DRUGS ORDER BY COST_PRICE ASC;
 name | type | barcode | dose | code | cost_price | selling_price | expiry | company_name | production_date | expiration_date | place | quantity
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
Novalo | Bills | fsdgdjfhjorodsf | normal | 3d00 | 2 | 3 | Available for use | Cipla | 2021-03-03 | 2023-03-03 | N-Right | 40
novafol | Bills | ftrkl432432md | normal | 2xaa | 33 | 40 | Available for use | Sun Pharma | 2020-01-01 | 2022-01-01 | N-Left | 27
Amifostine | Anxiety | fnjldhicfjaipsk | normal | 2cf5 | 39 | 55 | Available for use | Sanofi | 2021-03-05 | 2023-05-07 | 262 DannyThomasPlace | 90
Abacavir | infection | fbankjiicfnl56 | normal | bhab | 44 | 50 | Available for use | Med_City | 2019-06-09 | 2022-04-03 | 262 DannyThomasPlace | 100
Clindamycin | skin | fljhsfc35bsyc | normal | sbcr | 50 | 60 | Available for use | Pfizer | 2021-12-12 | 2022-11-15 | N-Left | 80
Calcium | mineral | fois54hiusdh | normal | sd3d | 69 | 80 | Available for use | Johnson & Johnson | 2020-08-08 | 2025-11-18 | N-Left | 150
Dopamine | Lbp | fkjoifjp5sjlhvu | normal | 52kh | 100 | 150 | Available for use | Abbott | 2020-09-22 | 2025-09-30 | N-right | 200
Morphine | Painkiller | fkjhsido54zsc | normal | fs25 | 250 | 270 | Available for use | Glenmark | 2021-05-25 | 2024-12-09 | 262 DannyThomasPlace | 300
(8 rows)
```

4) SELECT *FROM LOGIN;

SELECT *FROM LOGIN WHERE ID=1;

```
pharmacy=# SELECT *FROM LOGIN;
 name | type | date | time | id
-----+-----+-----+-----+---
 admin | Admin | 2017-02-17 | 10:30:24 | 1
 admin | Admin | 2017-02-17 | 10:32:48 | 2
 mark | Employee | 2017-02-17 | 10:32:56 | 1
 admin | Admin | 2017-02-17 | 10:33:10 | 1
 bhavana | Employee | 2017-02-17 | 10:33:37 | 5
 admin | Admin | 2017-02-17 | 10:36:21 | 1
 basana | Admin | 2017-02-17 | 10:36:53 | 6
 admin | Admin | 2017-02-17 | 10:49:27 | 1
 bhagya | Admin | 2017-02-17 | 11:02:23 | 7
 admin | Admin | 2017-02-17 | 01:40:08 | 1
(10 rows)

pharmacy=# SELECT *FROM LOGIN WHERE ID=1;
 name | type | date | time | id
-----+-----+-----+-----+---
 admin | Admin | 2017-02-17 | 10:30:24 | 1
 mark | Employee | 2017-02-17 | 10:32:56 | 1
 admin | Admin | 2017-02-17 | 10:33:10 | 1
 admin | Admin | 2017-02-17 | 10:36:21 | 1
 admin | Admin | 2017-02-17 | 10:49:27 | 1
 admin | Admin | 2017-02-17 | 01:40:08 | 1
(6 rows)

pharmacy=# _
```

5) SELECT *FROM LOGIN;

SELECT DISTINCT NAME FROM LOGIN;

SQL Shell (psql)

```
pharmacy=# SELECT *FROM LOGIN;
 name | type | date | time | id
-----+-----+-----+-----+---
 admin | Admin | 2017-02-17 | 10:30:24 | 1
 admin | Admin | 2017-02-17 | 10:32:48 | 2
 mark | Employee | 2017-02-17 | 10:32:56 | 1
 admin | Admin | 2017-02-17 | 10:33:10 | 1
 bhavana | Employee | 2017-02-17 | 10:33:37 | 5
 admin | Admin | 2017-02-17 | 10:36:21 | 1
 basana | Admin | 2017-02-17 | 10:36:53 | 6
 admin | Admin | 2017-02-17 | 10:49:27 | 1
 bhagya | Admin | 2017-02-17 | 11:02:23 | 7
 admin | Admin | 2017-02-17 | 01:40:08 | 1
(10 rows)

pharmacy=# SELECT DISTINCT NAME FROM LOGIN;
 name
-----
 bhavana
 admin
 bhagya
 mark
 basana
(5 rows)
```

COMPLEX QUERIES

- 1) CREATE VIEW SALE_VIEW AS
SELECT NAME, TYPE, BARCODE
FROM SALE;
SELECT *FROM SALE_VIEW;

```
pharmacy=#  
pharmacy=#  
pharmacy=# select *from sale;  
  barcode | name      | type    | dose  | quantity | price | amount | date      |  
-----+-----+-----+-----+-----+-----+-----+-----  
fkjoifjp5sjlhvu | dopamine | Lbp     | normal | 2         | 150   | 300    | 2021-10-12 |  
fljhsfc35bsyc   | Clindamycin | skin    | normal | 4         | 60    | 200    | 2021-10-12 |  
fnjldhicfjaipsk | Amifostine | Anxiety | normal | 10        | 39    | 400    | 2021-10-13 |  
fljhsfc35bsyc   | Clindamycin | skin    | normal | 4         | 60    | 200    | 2021-10-13 |  
fkjoifjp5sjlhvu | dopamine | Lbp     | normal | 2         | 150   | 300    | 2021-10-14 |  
fkjhsido54zsc   | Morphine   | Painkiller | normal | 10        | 270   | 2700   | 2021-10-14 |  
fois54hiusdh    | Calcium    | mineral  | normal | 50        | 80    | 4000   | 2021-10-14 |  
fbankjiicfnl56  | Abacavir   | infection | normal | 50        | 50    | 1000   | 2021-10-14 |  
(8 rows)
```

```
pharmacy=# CREATE VIEW SALE_VIEW AS  
pharmacy=# SELECT NAME,TYPE,BARCODE  
pharmacy=# FROM SALE;  
CREATE VIEW  
pharmacy=# SELECT *FROM SALE_VIEW;  
  name    | type    | barcode      |  
-----+-----+-----  
dopamine | Lbp     | fkjoifjp5sjlhvu |  
Clindamycin | skin    | fljhsfc35bsyc   |  
Amifostine | Anxiety | fnjldhicfjaipsk |  
Clindamycin | skin    | fljhsfc35bsyc   |  
dopamine | Lbp     | fkjoifjp5sjlhvu |  
Morphine   | Painkiller | fkjhsido54zsc   |  
Calcium    | mineral  | fois54hiusdh    |  
Abacavir   | infection | fbankjiicfnl56  |  
(8 rows)
```

```
pharmacy=#
```

- 2) SELECT *FROM HISTORY_SALES;
SELECT *FROM HISTORY_SALES WHERE QUANTITY>=3 AND
AMOUNT>=13;

```
pharmacy=# SELECT *FROM HISTORY_SALES;  
user_name | barcode | name      | type    | dose      | quantity | price | amount | date      | time      |  
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----  
admin     | fsdgjfhjorodsf | Novalo   | Bills   | Free used | 2         | 6     | 12     | 2017-02-12 | 05:02:06 |  
admin     | ftrkl432432md  | novafol  | Bills   | Free used | 2         | 6     | 12     | 2017-02-12 | 05:02:26 |  
admin     | fbankjiicfnl56 | novafol  | Bills   | Free used | 4         | 6     | 24     | 2017-02-12 | 05:02:40 |  
admin     | fnjldhicfjaipsk | Amifostine | Injection | 1 (Day)   | 2         | 14    | 28     | 2017-02-13 | 01:38:00 |  
admin     | fois54hiusdh    | Calcium   | Injection | 1 (Day)   | 2         | 14    | 28     | 2017-02-13 | 01:38:10 |  
admin     | fljhsfc35bsyc   | Clindamycin | Injection | 1 (Day)   | 7         | 14    | 98     | 2017-02-13 | 01:38:28 |  
admin     | fkjoifjp5sjlhvu | Dopamine  | Injection | 1 (Day)   | 1         | 14    | 14     | 2017-02-13 | 01:38:46 |  
mark      | fkjhsido54zsc   | Morphine  | Bills   | Free used | 2         | 6     | 12     | 2017-02-13 | 01:59:34 |  
(8 rows)
```

```
pharmacy=# SELECT * FROM HISTORY_SALES WHERE QUANTITY >=3 AND AMOUNT>=13;
```

user_name	barcode	name	type	dose	quantity	price	amount	date	time
admin	fbankjijcfnl56	novafol	Bills	Free used	4	6	24	2017-02-12	05:02:40
admin	fljhsfc35bsyc	Clindamycin	Injection	1 (Day)	7	14	98	2017-02-13	01:38:28

(2 rows)

3) SELECT *FROM DRUGS;

**SELECT *FROM DRUGS WHERE COST_PRICE>=50 OR
QUANTITY>=50;**

```
pharmacy=# SELECT *FROM DRUGS;
```

name	type	barcode	dose	code	cost_price	selling_price	expiry	company_name	production_date	expiration_date	place	quantity
Novalo	Bills	fsdgjfiijorodsf	normal	3d00	2	3	Available for use	Cipla	2021-03-03	2023-03-03	N-Right	40
novafol	Bills	ftrkl432432md	normal	2xaa	33	40	Available for use	Sun Pharma	2020-01-01	2022-01-01	N-Left	27
Abacavir	infection	fbankjijcfnl56	normal	bhab	44	50	Available for use	Med_City	2019-06-09	2022-04-03	262 DannyThomasPlace	100
Amifostine	Anxiety	fnjldhicfjaipsk	normal	2cf5	39	55	Available for use	Sanofi	2021-03-05	2023-05-07	262 DannyThomasPlace	90
Calcium	mineral	fois54hiusdh	normal	sd3d	69	80	Available for use	Johnson & Johnson	2020-08-08	2025-11-18	N-Left	150
Clindamycin	skin	fljhsfc35bsyc	normal	sbcrr	50	60	Available for use	Pfizer	2021-12-12	2022-11-15	N-Left	80
Dopamine	Lbp	fkjoifjp5sjlhvu	normal	52kh	100	150	Available for use	Abbott	2020-09-22	2025-09-30	N-right	200
Morphine	Painkiller	fkjhsido54zsc	normal	fs25	250	270	Available for use	Glenmark	2021-05-25	2024-12-09	262 DannyThomasPlace	300

(8 rows)

```
pharmacy=# SELECT *FROM DRUGS WHERE COST_PRICE>=50 OR QUANTITY >=50;
```

name	type	barcode	dose	code	cost_price	selling_price	expiry	company_name	production_date	expiration_date	place	quantity
Abacavir	infection	fbankjijcfnl56	normal	bhab	44	50	Available for use	Med_City	2019-06-09	2022-04-03	262 DannyThomasPlace	100
Amifostine	Anxiety	fnjldhicfjaipsk	normal	2cf5	39	55	Available for use	Sanofi	2021-03-05	2023-05-07	262 DannyThomasPlace	90
Calcium	mineral	fois54hiusdh	normal	sd3d	69	80	Available for use	Johnson & Johnson	2020-08-08	2025-11-18	N-Left	150
Clindamycin	skin	fljhsfc35bsyc	normal	sbcrr	50	60	Available for use	Pfizer	2021-12-12	2022-11-15	N-Left	80
Dopamine	Lbp	fkjoifjp5sjlhvu	normal	52kh	100	150	Available for use	Abbott	2020-09-22	2025-09-30	N-right	200
Morphine	Painkiller	fkjhsido54zsc	normal	fs25	250	270	Available for use	Glenmark	2021-05-25	2024-12-09	262 DannyThomasPlace	300

(6 rows)

4) SELECT *FROM SALE;

SELECT BARCODE FROM SALE GROUP BY BARCODE HAVING
COUNT(BARCODE)<15;

```
pharmacy=#
pharmacy=#
pharmacy=#
pharmacy=# SELECT *FROM sale;
  barcode | name      | type   | dose  | quantity | price | amount | date
-----+-----+-----+-----+-----+-----+-----+-----
fkjoifjp5sjlhvu | dopamine | Lbp    | normal | 2         | 150   | 300    | 2021-10-12
fljhsfc35bsyc   | Clindamycin | skin   | normal | 4         | 60    | 200    | 2021-10-12
fnjldhicfjaipsk | Amifostine | Anxiety | normal | 10        | 39    | 400    | 2021-10-13
fljhsfc35bsyc   | Clindamycin | skin   | normal | 4         | 60    | 200    | 2021-10-13
fkjoifjp5sjlhvu | dopamine | Lbp    | normal | 2         | 150   | 300    | 2021-10-14
fkjhsido54zsc   | Morphine   | Painkiller | normal | 10        | 270   | 2700   | 2021-10-14
fois54hiusdh    | Calcium    | mineral | normal | 50        | 80    | 4000   | 2021-10-14
fbankjiijcfnl56 | Abacavir   | infection | normal | 50        | 50    | 1000   | 2021-10-14
(8 rows)

pharmacy=# SELECT BARCODE FROM sale GROUP BY barcode HAVING count(barcode)<15;
  barcode
-----
fkjoifjp5sjlhvu
fnjldhicfjaipsk
fois54hiusdh
fljhsfc35bsyc
fbankjiijcfnl56
fkjhsido54zsc
(6 rows)

pharmacy=#
```

5) INNER JOIN

SELECT *FROM DRUGS;

CREATE VIEW DRUGS_VIEW AS SELECT COMPANY_NAME,
BARCODE FROM DRUGS;

SELECT NAME, ADDRESS, BARCODE

FROM COMPANY INNER JOIN DRUGS_VIEW


ON COMPANY.NAME=DRUGS_VIEW.COMPANY_NAME;


```
pharmacy=# CREATE VIEW DRUS_VIEW AS SELECT COMPANY_NAME, BARCODE FROM DRUGS;  
CREATE VIEW
```

```
pharmacy=# SELECT *FROM DRUGS_VIEW;  
ERROR: relation "drugs_view" does not exist  
LINE 1: SELECT *FROM DRUGS_VIEW;  
          ^
```

```
pharmacy=# SELECT *FROM DRUS_VIEW;  
   company_name | barcode  
-----+-----  
Cipla           | fsdgjfhjorodsf  
Sun Pharma      | ftrkl432432md  
Med_City        | fbankjijcfnl56  
Sanofi          | fnjldhiczfaipsk  
Johnson & Johnson | fois54hiusdh  
Pfizer          | fljhsfc35bsyc  
Abbott          | fkjoifjp5sjlhvu  
Glenmark        | fkjhsido54zsc  
(8 rows)
```


```
pharmacy=# SELECT *FROM COMPANY;  
   name | address | phone  
-----+-----+-----  
Cipla   | Mumbai  | 8523452596  
Sun Pharma | Mysore  | 8971199805  
Med_City | Nellore | 6669923589  
Sanofi   | mumbai  | 6894512397  
Johnson & Johnson | banglore | 8645321798  
Pfizer   | hyderabad | 9456312878  
Abbott   | kolkata  | 8631597234  
Glenmark | delhi    | 7539518264  
(8 rows)
```

 Select SQL Shell (psql)

```
pharmacy=# SELECT NAME, ADDRESS, BARCODE FROM COMPANY INNER JOIN DRUS_VIEW ON COMPANY.NAME = DRUS_VIEW.COMPANY_NAME;  
   name | address | barcode  
-----+-----+-----  
Cipla   | Mumbai  | fsdgjfhjorodsf  
Sun Pharma | Mysore  | ftrkl432432md  
Med_City | Nellore | fbankjijcfnl56  
Sanofi   | mumbai  | fnjldhiczfaipsk  
Johnson & Johnson | banglore | fois54hiusdh  
Pfizer   | hyderabad | fljhsfc35bsyc  
Abbott   | kolkata  | fkjoifjp5sjlhvu  
Glenmark | delhi    | fkjhsido54zsc  
(8 rows)
```

NESTED QUERIES

```
SELECT NAME, COMPANY_NAME, BARCODE
FROM DRUGS
WHERE BARCODE IN
(SELECT BARCODE FROM SALE
WHERE QUANTITY>=10);
```

 SQL Shell (psql)

```
pharmacy=# SELECT name, company_name,barcode
pharmacy=#
pharmacy=# FROM drugs
pharmacy=# WHERE barcode IN
pharmacy=# (SELECT barcode FROM sale
pharmacy=# WHERE quantity>=10)
pharmacy=# ;
   name   | company_name | barcode
-----+-----+-----
 Abacavir | Med_City     | fbankjijcfnl56
 Amifostine | Sanofi       | fnjldhiczfjaipsk
 Calcium   | Johnson & Johnson | fois54hiusdh
 Morphine   | Glenmark     | fkjhsido54zsc
(4 rows)

pharmacy=#
```

SELECT *FROM USERS;

SELECT NAME, ADDRESS, SALARY FROM USERS

WHERE SALARY>

(SALARY MAX(SALARY) FROM USERS

WHERE NAME= "CLARK");

```
pharmacy=# select *from users;
```

id	name	dob	address	phone	salary	password
1	admin	23-12-1995	Banglore	9800000000	50000	admin
2	mark	3-2-1972	Bangalore	9632587418	2000	mark
3	clark	3-2-1971	Richmond circle	8521479633	4000	rootaccess
4	arya	7-8-1977	Banglore	6478932143	3000	rootaccess
6	basana	12-07-2001	Belgaum	6547893218	7000	rootaccess
7	bhagya	06-06-2001	Banglore	8265479313	9000	rootaccess
8	Tom	09-06-2001	mumbai	8971199805	10000	rootaccess
5	bhavana	3-4-2002	Banglore	9980589711	15000	rootaccess

(8 rows)

```
pharmacy=# SELECT name,address,salary FROM users
pharmacy=# WHERE salary>
pharmacy=# (SELECT max(salary) FROM users
pharmacy=# WHERE name='clark');
```

name	address	salary
admin	Banglore	50000
basana	Belgaum	7000
bhagya	Banglore	9000
Tom	mumbai	10000
bhavana	Banglore	15000

(5 rows)

SELECT COUNT (*) AS "RECORDS" FROM COMPANY;

```
pharmacy=# SELECT *FROM COMPANY;
```

name	address	phone
Cipla	Mumbai	8523452596
Sun Pharma	Mysore	8971199805
Med_City	Nellore	6669923589
Sanofi	mumbai	6894512397
Johnson & Johnson	banglore	8645321798
Pfizer	hyderabad	9456312878
Abbott	kolkata	8631597234
Glenmark	delhi	7539518264

(8 rows)

```
pharmacy=# SELECT COUNT(*) AS "RECORDS" FROM COMPANY;
```

RECORDS
8

(1 row)

EXECUTION PLANS

SELECT *FROM COMPANY;

SELECT *FROM USERS;

EXPLAIN SELECT *FROM COMPANY T1,USERS T2 WHERE
T1.ADDRESS=T2.ADDRESS;

```
postgres=# select * from company;
   name   | address | phone
-----+-----+-----
 Cipla    | Mumbai  | 8523452596
 Sun_Pharma | Mysore  | 8971199805
 Med_City  | Nellore | 6669923589
 Sanofi    | mumbai  | 6894512397
 Johnson & Johnson | banglore | 8645321798
 Pfizer    | hyderabad | 9456312878
 Abbott    | kolkata | 8631597234
 Glenmark  | delhi    | 7539518264
 Pfizer    | hyderabad | 9456312878
 Abbott    | kolkata | 8631597234
 Glenmark  | delhi    | 7539518264
(11 rows)
```

```
postgres=# select * from users;
 id | name | dob   | address          | phone      | salary | password
----+-----+-----+-----+-----+-----+-----
  1 | admin | 23-12-1995 | Bangalore        | 9800000000 | 50000 | admin
  2 | mark  | 3-2-1972  | Bangalore        | 9632587418 | 2000  | mark
  3 | clark | 3-2-1971  | Richmond circle  | 8521479633 | 4000  | rootaccess
  4 | arya  | 7-8-1977  | Bangalore        | 6478932143 | 3000  | rootaccess
  5 | bhavana | 3-4-2002 | Bangalore        | 9980589711 | 10000 | rootaccess
  6 | basana | 12-07-2001 | Belgaum          | 6547893218 | 7000  | rootaccess
  7 | bhagya | 06-06-2001 | Bangalore        | 8265479313 | 9000  | rootaccess
  8 | Tom   | 09-06-2001 | mumbai           | 8971199805 | 10000 | rootaccess
(8 rows)
```

```
postgres=# explain select * from company t1, users t2 where t1.address=t2.address;
               QUERY PLAN
-----
Hash Join  (cost=13.15..28.34 rows=175 width=816)
  Hash Cond: ((t1.address)::text = (t2.address)::text)
    -> Seq Scan on company t1  (cost=0.00..12.50 rows=250 width=294)
    -> Hash  (cost=11.40..11.40 rows=140 width=522)
          -> Seq Scan on users t2  (cost=0.00..11.40 rows=140 width=522)
(5 rows)
```

EXPLAIN SELECT DISTINCT MESSAGE_FROM INBOX;

EXPLAIN SELECT *FROM USERS WHERE PASSWORD= 'ROOTACCESS';

EXPLAIN UPDATE COMPANY SET PHONE=6985324175 WHERE
NAME= 'PFIZER';

```

postgres=# explain select distinct message_from from inbox;
               ^
               QUERY PLAN
-----
HashAggregate (cost=11.75..13.15 rows=140 width=58)
  Group Key: message_from
    -> Seq Scan on inbox (cost=0.00..11.40 rows=140 width=58)
(3 rows)

```

```

postgres=# explain select * from users where password='rootaccess';
               ^
               QUERY PLAN
-----
Seq Scan on users (cost=0.00..11.75 rows=1 width=522)
  Filter: ((password)::text = 'rootaccess'::text)
(2 rows)

```

```

postgres=# explain update company set phone= 6985324175 where name = 'Pfizer';
               ^
               QUERY PLAN
-----
Update on company (cost=0.14..8.16 rows=0 width=0)
  -> Index Scan using company_pkey on company (cost=0.14..8.16 rows=1 width=64)
        Index Cond: ((name)::text = 'Pfizer'::text)
(3 rows)

```

TRIGGER FUNCTIONS

1) CREATE OR REPLACE FUNCTION DRUGSLOGFUNC() RETURNS TRIGGER AS \$EXAMPLE_TABLE\$

BEGIN

INSERT INTO drugs(NAME, TYPE, BARCODE, DOSE, CODE, COST_PRICE, SELLING_PRICE, EXPIRY, COMPANY_NAME, PRODUCTION_DATE, EXPIRATION_DATE, PLACE, QUANTITY)

VALUES('Novalo', 'Bills', 'fsdgjfhjorodsf', 'normal', '3d00', 2, 3, 'Available for use', 'Cipla', '2021-03-03', '2023-03-03', 'N-Right', 40),

RETURN NEW;

END;

\$EXAMPLE_TABLE\$ LANGUAGE PLPGSQL;

```
pharmacy=# CREATE OR REPLACE FUNCTION drugslogfunc() RETURNS TRIGGER AS $example_table$
pharmacy$# BEGIN
pharmacy$# INSERT INTO drugs(NAME, TYPE, BARCODE, DOSE, CODE, COST_PRICE, SELLING_PRICE, EXPIRY, COMPANY_NAME, PRODUCTION_DATE, EXPIRATION_DATE, PLACE, QUANTITY) VALUES ('Novalo', 'Bills', 'fsdgjfhjorodsf', 'normal', '3d00', 2, 3, 'Available for use', 'Cipla', '2021-03-03', '2023-03-03', 'N-Right', 40);
pharmacy$# RETURN NEW;
pharmacy$# END;
pharmacy$# $example_table$ LANGUAGE plpgsql;
CREATE FUNCTION
pharmacy=# CREATE TRIGGER example_trigger AFTER INSERT ON COMPANY
pharmacy=# FOR EACH ROW EXECUTE PROCEDURE drugslogfunc();
CREATE TRIGGER
pharmacy=#
```

CREATE TRIGGER EXAMPLE_TRIGGER AFTER INSERT ON COMPANY
FOR EACH ROW EXECUTE PROCEDURE DRUGSLOGFUNC();

```

2) CREATE OR REPLACE FUNCTION DRUGS_INSERT ()
RETURNS TRIGGER AS
$$
BEGIN
INSERT INTO DRUGS_LOG (NAME, COST_PRICE, EDITTIME)
VALUES (NEW.NAME,NEW.COST_PRICE, CURRENT_DATE);
RETURN NEW;
END;
$$
LANGUAGE 'PLPGSQL';

```

```

pharmacy=#
pharmacy=#
pharmacy=# CREATE OR REPLACE FUNCTION drugs_insert()
pharmacy=# RETURNS trigger AS
pharmacy=# $$
pharmacy$# BEGIN
pharmacy$# INSERT INTO drugs_log(name,cost_price,edittime)
pharmacy$# VALUES(NEW.name,NEW.cost_price,current_date);
pharmacy$# RETURN NEW;
pharmacy$# END;
pharmacy$# $$
pharmacy=# LANGUAGE 'plpgsql';
CREATE FUNCTION
pharmacy=#
pharmacy=# CREATE TRIGGER drugs_trigger
pharmacy=# AFTER INSERT
pharmacy=# ON drugs
pharmacy=# FRO EACH ROW
pharmacy=# EXECUTE PROCEDURE drugs_insert();
ERROR:  syntax error at or near "FRO"
LINE 4: FRO EACH ROW
        ^
pharmacy=# CREATE TRIGGER drugs_trigger
pharmacy=# AFTER INSERT
pharmacy=# ON drugs
pharmacy=# FOR EACH ROW
pharmacy=# EXECUTE PROCEDURE drugs_insert();
CREATE TRIGGER
pharmacy=#

```

```

CREATE TRIGGER DRUGS_TRIGGER
AFTER INSERT
ON DRUGS

```

FOR EACH ROW

EXECUTE PROCEDURE DRUGS_INSERT ()

```
SQL Shell (psql)
pharmacy=# INSERT INTO drugs(NAME, TYPE, BARCODE, DOSE, CODE, COST_PRICE, SELLING_PRICE, EXPIRY, COMPANY_NAME, PRODUCTION_DATE, EXPIRATION_DATE, PLACE, QUANTITY)
pharmacy-# VALUES('paracetamol', 'Pain', 'fkjhsido54abi', 'normal', 'fs03', 100, 110, 'Available for use', 'Pfizer', '2021-11-04', '2023-12-09', 'west ', 100);
INSERT 0 1
pharmacy=# SELECT *FROM drugs_log;
   name   | cost_price | edittime
-----+-----+-----
paracetamol |      100 | 2021-11-07
(1 row)

pharmacy=#
```

3) CREATE OR REPLACE FUNCTION BEFO_INSERT()

RETURNS TRIGGER AS

\$\$

BEGIN

NEW.NAME=UPPER(NEW.NAME);

RETURN NEW;

END;

\$\$

LANGUAGE 'PLPGSQL';

CREATE TRIGGER BEFO_INS

BEFORE INSERT

ON DRUGS

FOR EACH ROW

EXCEUTE PROCEDURE BEFO_INSERT ();

SQL Shell (psql)

```

pharmacy=#
pharmacy=# CREATE OR REPLACE FUNCTION befo_insert()
pharmacy=# RETURNS trigger AS
pharmacy=# $$
pharmacy$# BEGIN
pharmacy$# NEW.name=UPPER(NEW.name);
pharmacy$# RETURN NEW;
pharmacy$# END;
pharmacy$# $$
pharmacy=# LANGUAGE 'plpgsql';
CREATE FUNCTION
pharmacy=# CREATE TRIGGER che_val_befo_ins
pharmacy=# BEFORE INSERT
pharmacy=# ON drugs
pharmacy=# FOR EACH ROW
pharmacy=# EXECUTE PROCEDURE befo_insert();
ERROR: trigger "che_val_befo_ins" for relation "drugs" already exists
pharmacy=#
pharmacy=#
pharmacy=# CREATE TRIGGER befo_ins
pharmacy=# BEFORE INSERT
pharmacy=# ON drugs
pharmacy=# FOR EACH ROW
pharmacy=# EXECUTE PROCEDURE befo_insert();
CREATE TRIGGER
pharmacy=#

```

```

pharmacy=#
pharmacy=# INSERT INTO drugs(NAME, TYPE, BARCODE, DOSE, CODE, COST_PRICE, SELLING_PRICE, EXPIRY, COMPANY_NAME, PRODUCTION_DATE, EXPIRATION_DATE, PLACE, QUANTITY)
pharmacy=# VALUES('paracetamol', 'Pain', 'fkjhsido54abi', 'normal', 'fs03', 100, 110, 'Available for use', 'Pfizer', '2021-11-04', '2023-12-09', 'west ', 100);
ERROR: duplicate key value violates unique constraint "drugs_pkey"
DETAIL:  Key (barcode)=(fkjhsido54abi) already exists.
pharmacy=# INSERT INTO drugs(NAME, TYPE, BARCODE, DOSE, CODE, COST_PRICE, SELLING_PRICE, EXPIRY, COMPANY_NAME, PRODUCTION_DATE, EXPIRATION_DATE, PLACE, QUANTITY)
pharmacy=# VALUES('asprin', 'Pain', 'fkjhs03Ho54BAB', 'normal', 'fs33', 100, 110, 'Available for use', 'Pfizer', '2021-11-04', '2023-12-09', 'west ', 100);
INSERT 0 1
pharmacy=# SELECT *FROM drugs;

```

name	type	barcode	dose	code	cost_price	selling_price	expiry	company_name	production_date	expiration_date	place	quantity
Novalo	Bills	fsdgdjfhjorodsf	normal	3d00	2	3	Available for use	Cipla	2021-03-03	2023-03-03	N-Right	40
novafol	Bills	fttrl432432md	normal	2xaa	33	40	Available for use	Sun Pharma	2020-01-01	2022-01-01	N-Left	27
Abacavir	infection	fbankjijcfnl56	normal	bhab	44	50	Available for use	Med_City	2019-06-09	2022-04-03	262 DannyThomasPlace	100
Amifostine	Anxiety	fnjldhfcfjaipsk	normal	2cf5	39	55	Available for use	Sanofi	2021-03-05	2023-05-07	262 DannyThomasPlace	90
Calcium	mineral	fois54hiusdh	normal	sd3d	69	80	Available for use	Johnson & Johnson	2020-08-08	2025-11-18	N-Left	150
Clindamycin	skin	fljhsfc3sbsyc	normal	sbcx	50	60	Available for use	Pfizer	2021-12-12	2022-11-15	N-Left	80
Dopamine	Lbp	fkjoifj05sjlhvu	normal	52kh	100	150	Available for use	Abbott	2020-09-22	2025-09-30	N-right	200
Morphine	Painkiller	fkjhsido54zsc	normal	fs25	250	270	Available for use	Glenmark	2021-05-25	2024-12-09	262 DannyThomasPlace	300
paracetamol	Pain	fkjhsido54abi	normal	fs03	100	110	Available for use	Pfizer	2021-11-04	2023-12-09	west	100
ASPRIN	Pain	fkjhs03Ho54BAB	normal	fs33	100	110	Available for use	Pfizer	2021-11-04	2023-12-09	west	100

(10 rows)

4) CREATE OR REPLACE FUNCTION aft_update()

RETURNS trigger AS

\$\$

BEGIN

INSERT into purchase_log VALUES (name, CONCAT('Update
MEDICINE ',

```

OLD.NAME,' PREVIOUS QUANTITY :',OLD.QUANTITY,' PRESENT
QUANTITY ',
NEW.QUANTITY));
RETURN NEW;
END;

$$
LANGUAGE 'plpgsql';

```

```


pharmacy=#
pharmacy=# CREATE OR REPLACE FUNCTION aft_update()
pharmacy=# RETURNS trigger AS
pharmacy=# $$
pharmacy=# BEGIN
pharmacy=# INSERT into purchase_log VALUES (name, CONCAT('Update MEDICINE',OLD.NAME,'PREVIOUS QUANTITY :',OLD.QUANTITY,'PRESENT QUANTITY :',NEW.QUANTITY));
pharmacy=# RETURN NEW;
pharmacy=# END;
pharmacy=# $$
pharmacy=# LANGUAGE 'plpgsql';
CREATE FUNCTION
pharmacy=#

```

```

CREATE TRIGGER updt_log
AFTER UPDATE
ON PURCHASE
FOR EACH ROW
EXECUTE PROCEDURE aft_update();

```

 SQL Shell (psql)

```

pharmacy=#
pharmacy=#
pharmacy=# CREATE TRIGGER updt_log
pharmacy=# AFTER UPDATE
pharmacy=# ON purchase
pharmacy=# FOR EACH ROW
pharmacy=# EXECUTE PROCEDURE aft_update();
CREATE TRIGGER
pharmacy=#

```

ACCESS PRIVILIGES

```

CREATE ROLE BHAGYA;
GRANT UPDATE ON MESSAGE_HISTORY TO BHAGYA

```

```
postgres=# create role bhagya;  
CREATE ROLE  
postgres=# grant update on message_history to bhagya;  
GRANT
```

CREATE ROLE BHAVANA;

GRANT TRUNCATE ON USERS TO BHAVANA;

```
postgres=# create role bhavana;  
CREATE ROLE  
postgres=# grant alter on users to bhavana;  
ERROR: unrecognized privilege type "alter"  
postgres=# grant truncate on users to bhavana;  
GRANT
```

CREATE ROLE BASANA;

GRANT DELETE ON INBOX TO BASANA;

```
postgres=# create role basana;  
CREATE ROLE  
postgres=# grant delete on inbox to basana;  
GRANT
```

GRANT ALL ON PURCHASE TO ADMIN;

```
postgres=# grant all on purchase to admin;  
GRANT
```

CURSOR

CREATE FUNCTION reffunc(refcursor) RETURNS refcursor AS '

BEGIN

OPEN \$1 FOR SELECT name FROM drugs;

```
RETURN $1;

END;

' LANGUAGE plpgsql;

BEGIN;

SELECT reffunc('funcursor');

FETCH ALL IN funcursor;

COMMIT;
```

```
SQL Shell (psql)
pharmacy=#
pharmacy=#
pharmacy=#
pharmacy=# CREATE FUNCTION reffunc(refcursor) RETURNS refcursor AS '
pharmacy'# BEGIN
pharmacy'# OPEN $1 FOR SELECT name FROM drugs;
pharmacy'# RETURN $1;
pharmacy'# END;
pharmacy'# ' LANGUAGE plpgsql;
CREATE FUNCTION
pharmacy=#
pharmacy=# BEGIN;
BEGIN
pharmacy=*# SELECT reffunc('funcursor');
    reffunc
-----
    funcursor
(1 row)

pharmacy=*# FETCH ALL IN funcursor;
    name
-----
Novalo
novafol
Abacavir
Amifostine
Calcium
Clindamycin
Dopamine
Morphine
paracetamol
ASPRIN
(10 rows)

pharmacy=*# COMMIT;
COMMIT
pharmacy=#
```

Contribution

BASANAGOUDA S HADIMANI-PES2UG19CS089 – NESTED QUERIES, REPOR SPENT 2-3 HRS

BHAGYASHREE-PES2UG19CS085- EXECUTION PLAN, ACCESS PRIVILIGES, CURSOR SPENT 3-4 HRS

BHAVANA-PES2UG19CS089- SIMPLE, COMPLEX QUERIES, TRIGGER FUNCTION SPENT 3-4 HRS