

DBMS

ASSIGNMENT 3

Name : Kishan R Vaghamashi

Student ID: 202312014

11)

```
select count(ticket_id)ticket_count from movie.tickets where  
show_id=9;
```

The screenshot shows the pgAdmin 4 web interface. On the left is the 'Object Explorer' tree with the 'movie' database expanded, showing tables like 'cinema', 'customer', 'movies', 'screen', 'show', 'slots', and 'tickets'. The 'tickets' table is selected. The main pane shows a SQL query editor with the following query:

```
1 select count(ticket_id)ticket_count from movie.tickets where  
2 show_id=9;
```

Below the query editor is the 'Data Output' pane, which displays the results of the query in a table:

	ticket_count bigint
1	9

At the bottom of the interface, a green status bar indicates: 'Successfully run. Total query runtime: 51 msec. 1 rows affected.' Below this, it shows 'Total rows: 1 of 1' and 'Query complete 00:00:00.051'.

12)

```
select distinct m2.c_name  
from movie.customer as m1, movie.customer as m2  
where m2.c_id <> m1.c_id and m2.c_name = m1.c_name
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Language
- Publications
- Schemas (3)
 - movie
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - slots
 - tickets
 - Trigger Functions
 - Types
 - Views
 - public
 - stock_db
 - Subscriptions
- Login/Group Roles
- Tablespaces

postgres/postgres@kishan*

Query

```
1 select distinct m2.c_name
2 from movie.customer as m1, movie.customer as m2
3 where m2.c_id <> m1.c_id and m2.c_name = m1.c_name
4
```

Data Output

c_name
character varying (30)
1 Rahul
2 Neha
3 Prashant
4 Kena
5 Amisha

Total rows: 11 of 11 Query complete 00:00:00.065 Ln 3, Col 44

13)

```
select max(price) as Max_price
from movie.show;
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Language
- Publications
- Schemas (3)
 - movie
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - slots
 - tickets
 - Trigger Functions
 - Types
 - Views
 - public
 - stock_db
 - Subscriptions
- Login/Group Roles
- Tablespaces

postgres/postgres@kishan*

Query

```
1 select max(price) as Max_price
2 from movie.show;
```

Data Output

max_price
integer
1 500

Successfully run. Total query runtime: 44 msec. 1 rows affected.

Total rows: 1 of 1 Query complete 00:00:00.044 Ln 2, Col 11

14)

```
select distinct m1.cinema_name from movie.cinema m1 ,  
movie.screen m2 where m1.cinema_id=m2.cinema_id AND  
m2.capacity>100
```

The screenshot shows the pgAdmin 4 web interface. On the left is the 'Object Explorer' tree with the 'movie' schema expanded, showing tables like 'cinema', 'customer', 'movies', 'screen', 'show', 'slots', 'tickets', 'trigger_functions', 'types', 'views', 'public', 'stock_db', 'subscriptions', 'login/group_roles', and 'tablespaces'. The 'cinema' table is selected. The main panel displays a SQL query in the 'Query' tab:

```
1 select distinct m1.cinema_name from movie.cinema m1 ,  
2 movie.screen m2 where m1.cinema_id=m2.cinema_id AND  
3 m2.capacity>100
```

Below the query is the 'Data Output' tab, which shows a table with 12 rows and 1 column. The column is 'cinema_name' (character varying (30)). The rows list the cinema names: cinema land, cinepolis, film city, wide angle, juhu plex, electron cinema, amit cinema, cine lights, pvr acropolis, proton cinema, nilkanth, and modi cinema. At the bottom, it says 'Total rows: 12 of 12' and 'Query complete 00:00:00.072'.

cinema_name
cinema land
cinepolis
film city
wide angle
juhu plex
electron cinema
amit cinema
cine lights
pvr acropolis
proton cinema
nilkanth
modi cinema

15)

```
select distinct c.cinema_name from  
movie.cinema c where c.cinema_id in (select distinct  
s.cinema_id  
from movie.screen s where s.screen_id in (select  
distinct sh.screen_id from movie.show sh  
where sh.show_id in (select distinct t.show_id from  
movie.tickets t inner join movie.customer cu on t.c_id =  
cu.c_id where cu.c_name = 'Diya')));
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Language
- Publications
- Schemas (3)
 - movie
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - slots
 - tickets
 - Trigger Functions
 - Types
 - Views
 - public
 - stock_db
 - Subscriptions
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 select distinct c.cinema_name from
2 movie.cinema c where c.cinema_id in (select distinct
3 s.cinema_id
4 from movie.screen s where s.screen_id in (select
5 distinct sh.screen_id from movie.show sh
6 where sh.show_id in (select distinct t.show_id from
7 movie.tickets t inner join movie.customer cu on t.c_id =
8 cu.c_id where cu.c_name = 'Diya')));

```

Data Output

cinema_name
electron cinema
galaxy
nilkanth
rajshree

Successfully run. Total query runtime: 66 msec. 4 rows affected.

Total rows: 4 of 4 Query complete 00:00:00.066 Ln 8, Col 37

16)

```

select distinct m1.c_name from movie.customer m1,
movie.show m2, movie.tickets t where m1.c_id=t.c_id and
m2.show_id=t.show_id and t.num_of_tickets * m2.price >2500

```

pgAdmin 4

File Object Tools Help

Object Explorer

- Language
- Publications
- Schemas (3)
 - movie
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - slots
 - tickets
 - Trigger Functions
 - Types
 - Views
 - public
 - stock_db
 - Subscriptions
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 select distinct m1.c_name from movie.customer m1,
2 movie.show m2, movie.tickets t where m1.c_id=t.c_id and
3 m2.show_id=t.show_id and t.num_of_tickets * m2.price >2500

```

Data Output

c_name
Noor
Mayank
Kashvi
Rahul
Drashti
Neha
Monica
Trisha
Shubham
Rachel
Kashish
Sidharth
Avi

Successfully run. Total query runtime: 99 msec. 76 rows affected.

Total rows: 76 of 76 Query complete 00:00:00.099 Ln 3, Col 59

17)

```
select c_name,c_age from movie.customer where  
c_gender='F' And c_age IN(select min(c_age)Age from  
movie.customer WHERE c_gender = 'F')
```

The screenshot shows the pgAdmin 4 web interface. On the left is the 'Object Explorer' tree with 'movie' expanded. The main pane shows a SQL query in the 'Query' tab. Below the query is the 'Data Output' tab, which displays a table with 4 rows. At the bottom, a status bar indicates 'Total rows: 4 of 4' and 'Query complete 00:00:00.063'. A green message box at the bottom right says 'Successfully run. Total query runtime: 63 msec. 4 rows affected.'

	c_name character varying (30)	c_age integer
1	Aanandi	20
2	Shreya	20
3	Chaitali	20
4	Dimple	20

18)

```
select c_name from movie.customer where c_id IN  
(select c_id from movie.tickets where show_id IN(select show_id  
from movie.show where m_id IN (select m_id from  
movie.movies where m_genre='horror')) GROUP BY c_id HAVING  
COUNT(*) >= 3);
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Languages
- Publications
- Schemas (3)
 - movie
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - slots
 - tickets
 - Trigger Functions
 - Types
 - Views
 - public
 - stock_db
 - Subscriptions
 - Login/Group Roles
 - Tablespaces

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 select c_name from movie.customer where c_id IN
2 (select c_id from movie.tickets where show_id IN(select show_id
3 from movie.show where m_id IN (select m_id from
4 movie.movies where m_genre='horror')) GROUP BY c_id HAVING
5 COUNT(*) >= 3);

```

Data Output

c_name
character varying (30)

Total rows: 0 of 0 Query complete 00:00:00.061 Ln 1, Col 1

19)

```

select count(t1.c_id) as ticket_booked, t1.c_gender
from movie.customer t1 inner join movie.tickets t2
on t1.c_id = t2.c_id
group by t1.c_gender

```

pgAdmin 4

File Object Tools Help

Object Explorer

- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - slots
 - tickets
 - Columns (5)
 - ticket_id
 - show_id
 - c_id
 - show_date
 - num_of_tickets
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - Trigger Functions

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 select count(t1.c_id) as ticket_booked, t1.c_gender
2 from movie.customer t1 inner join movie.tickets t2
3 on t1.c_id = t2.c_id
4 group by t1.c_gender
5

```

Data Output

	ticket_booked bigint	c_gender character varying (1)
1	221	M
2	279	F

Total rows: 2 of 2 Query complete 00:00:00.074 Ln 5, Col 1

20)

```
select slot_id from movie.slots where slot_id NOT IN  
(select distinct slot_id from movie.show)
```

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane displays a tree view of the database schema, including tables like 'cinema', 'customer', 'movies', 'screen', 'show', 'slots', and 'tickets'. The 'Columns (5)' section for the 'tickets' table is expanded, showing columns: 'ticket_id', 'show_id', 'c_id', 'show_date', and 'num_of_tickets'. The main pane displays a SQL query in the 'Query' tab:

```
1 select slot_id from movie.slots where slot_id NOT IN  
2 (select distinct slot_id from movie.show)
```

Below the query, the 'Data Output' pane shows the results of the query in a table format:

	slot_id [PK] integer
1	9
2	12
3	14
4	16
5	17
6	25

At the bottom of the interface, a status bar indicates: 'Total rows: 6 of 6', 'Query complete 00:00:00.160', and a green message box stating 'Successfully run. Total query runtime: 160 msec. 6 rows affected.' The right side of the status bar shows 'Ln 2, Col 42'.

21)

```
select m.m_name  
from movie.movies m inner join movie.show s  
on m.m_id = s.m_id  
order by s.price desc  
limit 3
```


pgAdmin 4

File Object Tools Help

Object Explorer

- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - Columns (5)
 - show_id
 - m_id
 - screen_id
 - slot_id
 - price
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - slots
 - tickets
 - Columns (5)

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 select m.m_name
2 from movie.movies m inner join movie.show s
3 on m.m_id = s.m_id
4 order by s.price desc
5 limit 3
6

```

Data Output

m_name
Raaz
Bhul Bhulaiya
Ek tha Tiger

Successfully run. Total query runtime: 65 msec. 3 rows affected.

Total rows: 3 of 3 Query complete 00:00:00.065 Ln 3, Col 16

22)

```

select m.m_name, count(t.c_id)popularity from
movie.movies m, movie.show sh, movie.tickets t,
movie.screen s, movie.customer c where
t.show_id=sh.show_id and sh.screen_id=s.screen_id and
sh.m_id=m.m_id and t.c_id=c.c_id and c.c_age>50
GROUP BY m.m_name ORDER BY popularity DESC;

```

pgAdmin 4

File Object Tools Help

Object Explorer

- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - Columns (5)
 - show_id
 - m_id
 - screen_id
 - slot_id
 - price
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - slots
 - tickets
 - Columns (5)

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 select m.m_name, count(t.c_id)popularity from
2 movie.movies m, movie.show sh, movie.tickets t,
3 movie.screen s, movie.customer c where
4 t.show_id=sh.show_id and sh.screen_id=s.screen_id and
5 sh.m_id=m.m_id and t.c_id=c.c_id and c.c_age>50
6 GROUP BY m.m_name ORDER BY popularity DESC;

```

Data Output

m_name	popularity
Welcome	10
Bhul Bhulaiya	7
Pari	6
Don	5
Khichdi	5
Talaash	4
Rahasya	4
Jane tu ya janana	4
Kanchna	3
Chachi 420	3
Prestige	3
Tirangaa	3
3 Idiots	3

Successfully run. Total query runtime: 88 msec. 27 rows affected.

Total rows: 27 of 27 Query complete 00:00:00.088 Ln 6, Col 44

23)

```
SELECT c.c_id, c.c_name
FROM movie.customer c
WHERE c.c_id IN ( SELECT t.c_id
                  FROM movie.tickets t
                  INNER JOIN ( SELECT DISTINCT a.show_id, a.slot_id
                              FROM movie.show a) sb
                  ON t.show_id = sb.show_id
                  GROUP BY t.c_id
HAVING COUNT(DISTINCT sb.slot_id) >= 5);
```

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane displays a tree structure of database objects, with 'show' selected under 'Tables (7)'. The main pane shows a SQL query editor with the following query:

```
1 SELECT c.c_id, c.c_name
2 FROM movie.customer c
3 WHERE c.c_id IN ( SELECT t.c_id
4                 FROM movie.tickets t
5                 INNER JOIN ( SELECT DISTINCT a.show_id, a.slot_id
6                             FROM movie.show a) sb
7                 ON t.show_id = sb.show_id
8                 GROUP BY t.c_id
9                 HAVING COUNT(DISTINCT sb.slot_id) >= 5);
10
```

Below the query editor, the 'Data Output' pane displays the results of the query in a table format:

c_id [PK] integer	c_name character varying (30)
1	Devanshi
2	Riya
3	Aash
4	Diya
5	Priya
6	Harshil
7	Foram
8	Kashish
9	Shivani
10	Neha
11	Devyani
12	Riya
13	Ayush

At the bottom of the interface, a status bar indicates: 'Total rows: 50 of 50', 'Query complete 00:00:00.066', and a green message box stating 'Successfully run. Total query runtime: 66 msec. 50 rows affected.'

24)

```
SELECT a.slot_id FROM movie.show a WHERE a.m_id IN
(SELECT m.m_id FROM movie.movies m WHERE m.m_genre = 'Horror')
GROUP BY a.slot_id
HAVING COUNT(*) =
( SELECT MAX(show_count) FROM ( SELECT COUNT(*) AS show_count
FROM movie.show a WHERE a.m_id IN
(SELECT m.m_id FROM movie.movies m WHERE m.m_genre = 'Horror')
GROUP BY a.slot_id)sb);
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - Columns (5)
 - show_id
 - m_id
 - screen_id
 - slot_id
 - price
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - slots
 - tickets
 - Columns (5)

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 SELECT a.slot_id FROM movie.show a WHERE a.m_id IN
2 (SELECT m.m_id FROM movie.movies m WHERE m.m_genre = 'Horror')
3 GROUP BY a.slot_id
4 HAVING COUNT(*) =
5 ( SELECT MAX(show_count) FROM ( SELECT COUNT(*) AS show_count
6 FROM movie.show a WHERE a.m_id IN
7 (SELECT m.m_id FROM movie.movies m WHERE m.m_genre = 'Horror')
8 GROUP BY a.slot_id)sb);
9

```

Data Output

slot_id	integer
1	27
2	30

Successfully run. Total query runtime: 62 msec. 2 rows affected.

Total rows: 2 of 2 Query complete 00:00:00.062 Ln 6, Col 28

25)

```

SELECT c.cinema_name FROM movie.cinema c GROUP
BY c.cinema_name HAVING COUNT(*) = 1;

```

pgAdmin 4

File Object Tools Help

Object Explorer

- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - Columns (5)
 - show_id
 - m_id
 - screen_id
 - slot_id
 - price
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - slots
 - tickets
 - Columns (5)

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

Query Query History Messages Notifications

```

1 SELECT c.cinema_name FROM movie.cinema c GROUP
2 BY c.cinema_name HAVING COUNT(*) = 1;

```

Data Output

cinema_name	character varying (30)
1	raahul cinema
2	sonia cinema
3	proton cinema
4	modi cinema
5	murlidhar
6	cinema land
7	cinepollis
8	galaxy
9	film city
10	neutron cinema
11	juhu plex
12	electron cinema
13	carnival

Successfully run. Total query runtime: 69 msec. 24 rows affected.

Total rows: 24 of 24 Query complete 00:00:00.069 Ln 1, Col 32

26)

```
SELECT DISTINCT m.m_name
FROM movie.movies m, movie.tickets t, movie.show sh,
movie.customer c WHERE t.show_id = sh.show_id
AND sh.m_id = m.m_id AND t.c_id IN (10, 11) AND t.c_id = c.c_id;
```

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the 'Object Explorer' with a tree view of database objects. The 'show' table is selected, showing its columns: show_id, m_id, screen_id, slot_id, and price. The main pane displays the SQL query entered in the 'Query' tab. The 'Data Output' pane shows the results of the query, which are 5 rows of movie names. A status bar at the bottom indicates the query was successfully run with a runtime of 69 msec, affecting 5 rows.

pgAdmin 4

File Object Tools Help

Object Explorer

- > Collations
- > Domains
- > FTS Configurations
- > FTS Dictionaries
- > FTS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > Sequences
- > Tables (7)
 - > cinema
 - > customer
 - > movies
 - > screen
 - > show
 - > Columns (5)
 - show_id
 - m_id
 - screen_id
 - slot_id
 - price
 - > Constraints
 - > Indexes
 - > RLS Policies
 - > Rules
 - > Triggers
 - > slots
 - > tickets
 - > Columns (5)

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

No limit

Query Query History Messages Notifications

```
1 SELECT DISTINCT m.m_name
2 FROM movie.movies m, movie.tickets t, movie.show sh,
3 movie.customer c WHERE t.show_id = sh.show_id
4 AND sh.m_id = m.m_id AND t.c_id IN (10, 11) AND t.c_id = c.c_id;
```

Data Output

	m_name character varying (30)
1	Chashme Badoor
2	Grudge
3	KGF
4	Raaz
5	Titanic

Successfully run. Total query runtime: 69 msec. 5 rows affected.

Total rows: 5 of 5 Query complete 00:00:00.069 Ln 1, Col 1

27)

```
SELECT DISTINCT c.c_name FROM movie.customer c
WHERE c.c_id IN ( SELECT t.c_id
FROM movie.tickets t GROUP BY t.c_id HAVING COUNT(*) > 1 );
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - Columns (5)
 - show_id
 - m_id
 - screen_id
 - slot_id
 - price
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - slots
 - tickets
 - Columns (5)

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

No limit

Query Query History Messages Notifications

```

1 SELECT DISTINCT c.c_name FROM movie.customer c
2 WHERE c.c_id IN ( SELECT t.c_id
3 FROM movie.tickets t GROUP BY t.c_id HAVING COUNT(*) > 1 );

```

Data Output

c_name
Noor
Kashvi
Mayank
Rahul
Drashti
Neha
Monica
Yagnik
Trisha
Shubham
Prerak
Rachel
Kashish

Total rows: 87 of 87 Query complete 00:00:00.061 Ln 3, Col 11

28)

```

SELECT DISTINCT c.c_name FROM movie.customer c
WHERE c.c_id IN ( SELECT t.c_id FROM movie.tickets t
WHERE t.show_id IN ( SELECT a.show_id
FROM movie.tickets a,
movie.show g,
movie.movies m
WHERE a.show_id = g.show_id
AND g.m_id = m.m_id
AND m.m_name = 'Welcome')
GROUP BY t.c_id HAVING COUNT(*) > 1);

```

pgAdmin 4

File Object Tools Help

Object Explorer

- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - Columns (5)
 - show_id
 - m_id
 - screen_id
 - slot_id
 - price
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - slots
 - tickets
 - Columns (5)

Dashboard Properties SQL Statistics Dependencies Dependents Processes postgres/postgres@kishan*

postgres/postgres@kishan

No limit

Query Query History Messages Notifications

```

1 SELECT DISTINCT c.c_name FROM movie.customer c
2 WHERE c.c_id IN ( SELECT t.c_id FROM movie.tickets t
3 WHERE t.show_id IN ( SELECT a.show_id
4 FROM movie.tickets a,
5 movie.show g,
6 movie.movies m
7 WHERE a.show_id = g.show_id
8 AND g.m_id = m.m_id
9 AND m.m_name = 'Welcome')
10 GROUP BY t.c_id HAVING COUNT(*) > 1 );
11

```

Data Output

c_name
Foram
Janki
Kashvi
Neelu
Phoebe
Prahar
Rahul
Ross

Total rows: 8 of 8 Query complete 00:00:00.071

Successfully run. Total query runtime: 71 msec. 8 rows affected. Ln 6, Col 18

29)

```
select count(cinema_id)total from movie.cinema where  
cinema_city='ahmedabad'
```

The screenshot shows the pgAdmin 4 web interface. On the left, the 'Object Explorer' pane is open, showing a tree view of the database schema. The 'show' table is selected, and its columns (show_id, m_id, screen_id, slot_id, price) are visible. The main pane displays a SQL query in the 'Query' tab:

```
1 select count(cinema_id)total from movie.cinema where  
2 cinema_city='ahmedabad'
```

Below the query, the 'Data Output' pane shows the results of the query. The results are displayed in a table with two columns: 'total' and 'bigint'. The first row shows the value '5'.

total	bigint
5	

At the bottom of the interface, a status bar indicates: 'Total rows: 1 of 1', 'Query complete 00:00:00.058', and a green message: 'Successfully run. Total query runtime: 58 msec. 1 rows affected.'

30)

```
SELECT m.m_name  
FROM movie.movies m  
WHERE m.m_id = ( SELECT a.m_id  
                  FROM movie.show a,  
                  movie.tickets t,  
                  movie.screen k,  
                  movie.cinema c  
                WHERE a.show_id = t.show_id AND a.screen_id = k.screen_id  
                AND k.cinema_id = c.cinema_id AND c.cinema_city = 'ahmedabad'  
                GROUP BY a.m_id  
                ORDER BY SUM(t.num_of_tickets * a.price) DESC LIMIT 1);
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Languages
- Publications
- Schemas (3)
 - movie
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (7)
 - cinema
 - customer
 - movies
 - screen
 - show
 - slots
 - tickets
 - Trigger Functions
 - Types
 - Views
 - public
 - stock_db
 - Subscriptions
- Login/Group Roles
- Tablespaces

postgres/postgres@kishan*

Query

```
2 FROM movie.movies m
3 WHERE m.m_id = ( SELECT a.m_id
4                 FROM movie.show a,
5                      movie.tickets t,
6                      movie.screen k,
7                      movie.cinema c
8                 WHERE a.show_id = t.show_id AND a.screen_id = k.screen_id
9                      AND k.cinema_id = c.cinema_id AND c.cinema_city = 'ahmedabad'
10                GROUP BY a.m_id
11                ORDER BY SUM(t.num_of_tickets * a.price) DESC LIMIT 1);
12
```

Data Output

m_name
character varying (30)
1 Conjuring

Total rows: 1 of 1 Query complete 00:00:00.065 Ln 3, Col 28