

Database Management System Project - UE19CS301

Movie ticket booking system

Assignment 3

Team Details:

Name	SRN
Prakruti P	PES1UG19CS338
Pruthvi P	PES1UG19CS355
Raghav Pandit	PES1UG19CS364

QUEIRES:

1)Movies with rating greater than 3

```
select m_name  
FROM movie  
where Rating>=3;
```

```
movie=# select m_name  
FROM movie  
where Rating>=3;  
 m_name  
-----  
Hella  
Ava  
Batman  
Run  
Time  
Dune  
Vertigo  
Big  
Up  
(9 rows)  
movie=#
```

2)movies playing in screen 1

```
select ti_id as ID,m_id  
from TICKET  
where SCREEN_no=1;
```

```
movie=# select ti_id as ID,m_id  
movie-# from TICKET  
movie-# where SCREEN_no=1;  
   id   | m_id  
-----+-----  
 12521  | 001  
 13323  | 004  
 13333  | 007  
 13346  | 001  
 12545  | 001  
 12423  | 001  
 12523  | 001  
(7 rows)
```

3)Discounts with values greater than 30

```
select d_name, d_id as code from  
discount  
where d_value>=30;
```

```
movie=# Select d_name, d_id AS code  
FROM Discount  
Where d_value>=30;  
   d_name   | code  
-----+-----  
 BIGSHOW   | PAJURB  
 Welcome   | BYSTEH  
(2 rows)
```

4) customers with no mail id listed

```
select c_id as id, c_name as name
from customer
where mail is null;
```

```
movie=# Select c_id As id, c_name AS name
movie-# From Customer
movie-# Where mail is NULL
movie-# ;
   id  | name
-----+-----
   111 | Bell
(1 row)
```

5) Movies released in year 2013

```
select m_name
FROM movie
where Release_date>='2013-01-01'
AND Release_date<'2013-12-30';
```

```
movie=# select m_name
FROM movie
where Release_date>='2013-01-01'
AND Release_date<'2014-12-30';
   m_name
-----
   Batman
(1 row)
```

6) theatre names that start with 'S'

```
Select t_id,t_name
from Theatre
where t_name~'^S.*$';
```

```
movie=# Select t_id,t_name
movie=# from Theatre
movie=# where t_name~'^S.*$';
 t_id |  t_name
-----+-----
  022 | September
  066 | Saturday
  077 | Star
(3 rows)
```

7) customers who purchased more than 2 tickets.

```
select customer.c_id, c_name, Count(*)
from customer, soldTO
where Customer.c_id=Soldto.c_id
group by Customer.c_id
having count(*)>=2;
```

```
movie=# Select Customer.c_id, c_name, Count(*)
from Customer, SoldTo
Where Customer.c_id=SoldTo.c_id
group By Customer.c_id
Having Count(*)>=2;
 c_id | c_name | count
-----+-----+-----
  666 | Glenn  |      2
(1 row)
```

8) ticket master who are currently free

```
select tm_name,tm_id
from ticketmaster
where tm_id not in (Select tm_id from ticket);
```

```
movie=# SELECT tm_name, tm_id
from TicketMaster
Where tm_id not in (Select tm_id from ticket);
      tm_name      | tm_id
-----+-----
Olivia             | 12564
Elizabeth          | 12632
Amy                | 12835
Glenn              | 12735
(4 rows)
```

9) ticket master who work in more than one theatre

```
select tmm.tm_id,tm_name, Count(*)
from ticketmaster as tmm, worksfor
where tmm.tm_id=worksfor.tm_id
group by tmm.tm_id
having count(*)>=1;
```

```
movie=# select tmm.tm_id,tm_name, Count(*)
movie=# from ticketmaster as tmm, worksfor
movie=# where tmm.tm_id=worksfor.tm_id
movie=# group by tmm.tm_id
movie=# having count(*)>=1;
   tm_id |      tm_name      | count
-----+-----+-----
12742   | Noah              |      1
12521   | Ava               |      1
12835   | Amy               |      2
12123   | Liam              |      1
12632   | Elizabeth         |      1
12564   | Olivia            |      1
12832   | Oliver            |      1
(7 rows)
```

10) what are different types of seats and how many were booked.

```
SELECT count(*) FILTER (WHERE seat_type = 'Luxury') AS luxury,  
       count(*) FILTER (WHERE seat_type = 'Premium') AS Premium,  
       count(*) FILTER (WHERE seat_type = 'Regular') AS Regular  
FROM SeatDetails;
```

```
movie=# SELECT count(*) FILTER (WHERE seat_type = 'Luxury') AS luxury,  
movie-#         count(*) FILTER (WHERE seat_type = 'Premium') AS Premium,  
movie-#         count(*) FILTER (WHERE seat_type = 'Regular') AS Regular  
movie-# FROM SeatDetails;  
 luxury | premium | regular  
-----+-----+-----  
        3 |         2 |         3  
(1 row)
```

Performance Analysis

1) Explain analyze
select tmm.tm_id,tm_name, Count(*)
from ticketmaster as tmm, worksfor
where tmm.tm_id=worksfor.tm_id
group by tmm.tm_id
having count(*)>=1;

```
QUERY PLAN  
-----  
HashAggregate (cost=42.08..48.08 rows=160 width=150) (actual time=0.065..0.071 rows=7 loops=1)  
  Group Key: tmm.tm_id  
  Filter: (count(*) >= 1)  
    -> Hash Join (cost=20.80..37.88 rows=560 width=142) (actual time=0.050..0.056 rows=8 loops=1)  
        Hash Cond: ((worksfor.tm_id)::text = (tmm.tm_id)::text)  
        -> Seq Scan on worksfor (cost=0.00..15.60 rows=560 width=58) (actual time=0.008..0.009 rows=8 loops=1)  
        -> Hash (cost=14.80..14.80 rows=480 width=142) (actual time=0.018..0.018 rows=9 loops=1)  
            Buckets: 1024 Batches: 1 Memory Usage: 9kB  
            -> Seq Scan on ticketmaster tmm (cost=0.00..14.80 rows=480 width=142) (actual time=0.004..0.005 rows=9 loops=1)  
s=1)  
Planning Time: 0.496 ms  
Execution Time: 0.198 ms  
(11 rows)
```

2) Explain analyze
Select t_id,t_name
from Theatre
where t_name~'^S.*\$';

```
movie=# explain analyze  
movie-# Select t_id,t_name  
movie-# from Theatre  
movie-# where t_name~'^S.*$';  
QUERY PLAN  
-----  
Seq Scan on theatre (cost=0.00..14.12 rows=1 width=116) (actual time=0.029..0.033 rows=3 loops=1)  
  Filter: ((t_name)::text ~ '^S.*$'::text)  
  Rows Removed by Filter: 5  
Planning Time: 0.426 ms  
Execution Time: 0.063 ms  
(5 rows)
```

USERS:

Creating multiple users:

```
Command Prompt - psql -U postgres

C:\Program Files\PostgreSQL\13\bin>psql -U postgres
Password for user postgres:
psql (13.4)
WARNING: Console code page (850) 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 movie
You are now connected to database "movie" as user "postgres".
movie=# create user buyer1 password 'buyer1';
CREATE ROLE
movie=# create user buyer2 password 'buyer2';
CREATE ROLE
movie=# create user ticketmaster password 'tm1';
ERROR:  role "ticketmaster" already exists
movie=#
```

Granting access privileges to the users:

```
Command Prompt - psql -U postgres

Type "help" for help.

postgres=# \c movie
You are now connected to database "movie" as user "postgres".
movie=# create user buyer1 password 'buyer1';
CREATE ROLE
movie=# create user buyer2 password 'buyer2';
CREATE ROLE
movie=# create user ticketmaster password 'tm1';
ERROR:  role "ticketmaster" already exists
movie=# grant select,insert on Movie,Theatre,SeatDetails to buyer1,buyer2;
GRANT
movie=# grant select,insert,update on Customer,Ticket to ticketmaster;
GRANT
movie=#
```

```

C:\Program Files\PostgreSQL\13\bin>psql -U ticketmaster movie

C:\Program Files\PostgreSQL\13\bin>psql -U buyer1 movie
Password for user buyer1:
psql (13.4)
WARNING: Console code page (850) 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.

movie=> select * from Movie;
 m_id | m_name | duration | release_date | rating
-----+-----+-----+-----+-----
  001 | Hella |      120 | 2012-12-10 | 4
  002 | Ava |      122 | 2002-11-15 | 4
  003 | Batman |      100 | 2013-04-23 | 4
  004 | Run |       98 | 1982-03-17 | 4
  005 | Time |      128 | 2007-07-28 | 4
  006 | Dune |      140 | 2021-09-09 | 4
  007 | Vertigo |      210 | 2011-11-11 | 4
  008 | Big |      138 | 2010-08-10 | 4
  009 | Up |      120 | 2015-11-09 | 4
(9 rows)

movie=> select * from Theatre;
 t_id | t_name | t_location
-----+-----+-----
  011 | One | 450 Stone, Houston,TX
  022 | September | 731 Fondren,Houston,TX
  033 | Bloom | 638 voss,Houston,TX
  044 | No Drama | 3321 Castle, Spring,TX
  055 | Magic | 291 Berry, Bellaire,TX
  066 | Saturday | 975 Fire Oak, Humble, TX
  077 | Star | 5631 Rice,Houston,TX
  088 | Linc | 980 Dallas, Houston,TX
(8 rows)

movie=> select * from Customer;
ERROR: permission denied for table customer
movie=> \q

```

```

C:\Program Files\PostgreSQL\13\bin>psql -U ticketmaster movie

C:\Program Files\PostgreSQL\13\bin>psql -U ticketmaster movie
Password for user ticketmaster:
psql (13.4)
WARNING: Console code page (850) 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.

movie=> select * from Customer;
 c_id | c_name | age | mail | phone
-----+-----+-----+-----+-----
  111 | Bell | 21 | | 9019273618
  222 | Chiyenne | 34 | chiyenne876@hotmail.com | 9019273678
  333 | Vayre | 16 | reyAv_87@gmail.com | 9018645792
  444 | Andy | 33 | Andyyyyy@gmail.com | 8547546931
  555 | Jonah | 46 | JJ_roxx2324@hotmail.com | 8479996325
  666 | Glenn | 25 | Glenn_jesus@gmail.com | 9663554771
  777 | Jake | 12 | Jake_bk99@gmail.com | 6544441223
  888 | Amy | 10 | AmySan8989@hotmail.com | 8745445663
(8 rows)

movie=> select * from Ticket;
 ti_id | show_date | show_time | screen_no | price | m_id | tm_id
-----+-----+-----+-----+-----+-----+-----
 12521 | 2010-10-23 | 12:30:00 | 1 | 350 | 001 | 12111
 12442 | 2021-10-10 | 12:30:00 | 4 | 700 | 006 | 12832
 12342 | 2016-09-12 | 13:00:00 | 2 | 350 | 005 | 12742
 14233 | 2017-02-03 | 17:30:00 | 5 | 400 | 006 | 12742
 13323 | 2019-10-21 | 20:30:00 | 1 | 350 | 004 | 12832
 13333 | 2018-07-22 | 09:30:00 | 1 | 300 | 007 | 12521
 13346 | 2010-10-23 | 12:30:00 | 1 | 350 | 001 | 12123
 12545 | 2010-10-23 | 12:30:00 | 1 | 350 | 001 | 12123
 12423 | 2010-10-23 | 12:30:00 | 1 | 350 | 001 | 12123
 12523 | 2010-10-23 | 12:30:00 | 1 | 350 | 001 | 12123
(10 rows)

movie=> select * from SeatDetails;
ERROR: permission denied for table seatdetails
movie=> grant select,insert,update on SeatDetails to ticketmaster;
ERROR: permission denied for table seatdetails
movie=>

```


TRANSACTIONS:

```
You are now connected to database "movie" as user "postgres".
movie=# BEGIN;
BEGIN
movie=# INSERT INTO Customer values('999','Elio',17,'elio23@hotmail.com',8
763445663);
INSERT 0 1
movie=# commit;
COMMIT
```

Before Commit:

```
movie=# SELECT * FROM Customer;
 c_id | c_name | age | mail | phone
-----+-----+----+-----+-----
 111 | Bell | 21 | | 9019273618
 222 | Chiyenne | 34 | chiyenne876@hotmail.com | 9019273678
 333 | Vayre | 16 | reyAv_87@gmail.com | 9018645792
 444 | Andy | 33 | Andyyyyy@gmail.com | 8547546931
 555 | Jonah | 46 | JJ_roxx2324@hotmail.com | 8479996325
 666 | Glenn | 25 | Glenn__jesus@gmail.com | 9663554771
 777 | Jake | 12 | Jake_bk99@gmail.com | 6544441223
 888 | Amy | 10 | AmySan8989@hotmail.com | 8745445663
(8 rows)
```

After Commit:

```
movie=# SELECT * FROM Customer;
 c_id | c_name | age | mail | phone
-----+-----+----+-----+-----
 111 | Bell | 21 | | 9019273618
 222 | Chiyenne | 34 | chiyenne876@hotmail.com | 9019273678
 333 | Vayre | 16 | reyAv_87@gmail.com | 9018645792
 444 | Andy | 33 | Andyyyyy@gmail.com | 8547546931
 555 | Jonah | 46 | JJ_roxx2324@hotmail.com | 8479996325
 666 | Glenn | 25 | Glenn__jesus@gmail.com | 9663554771
 777 | Jake | 12 | Jake_bk99@gmail.com | 6544441223
 888 | Amy | 10 | AmySan8989@hotmail.com | 8745445663
 999 | Elio | 17 | elio23@hotmail.com | 8763445663
(9 rows)
```

Concurrent transactions:

Starting a new transaction

```
movie=# BEGIN;  
BEGIN  
movie=# Update movie set duration=122 where m_id='001';  
UPDATE 1  
movie=#
```

WE NOTICE THAT THE VALUE HAS BEEN UPDATED

Running a query on the same id and table in another terminal

```
postgres=# \c movie  
You are now connected to database "movie" as user "postgres".  
movie=# Update movie set duration=142 where m_id='001';  
UPDATE 1  
movie=#
```

THIS IS WAITING FOR THE TRANSACTION TO BE COMPLETED

After rollback of the transaction

```
movie=# BEGIN;  
BEGIN  
movie=# Update movie set duration=122 where m_id='001';  
UPDATE 1  
movie=# rollback;  
ROLLBACK  
movie=#
```

After rollback of transaction the query finishes execution

```
movie=# Update movie set duration=142 where m_id='001';  
UPDATE 1  
movie=#
```

We notice that latest update in the table.

```
movie=# select * from movie;
 m_id | m_name  | duration | release_date | rating
-----+-----+-----+-----+-----
 002  | Ava     |    122  | 2002-11-15   |      4
 003  | Batman  |    100  | 2013-04-23   |      4
 004  | Run     |     98  | 1982-03-17   |      4
 005  | Time    |    128  | 2007-07-28   |      4
 006  | Dune    |    140  | 2021-09-09   |      4
 007  | Vertigo |    210  | 2011-11-11   |      4
 008  | Big     |    138  | 2010-08-10   |      4
 009  | Up      |    120  | 2015-11-09   |      4
 001  | Hella   |    142  | 2012-12-10   |      4
(9 rows)
```

Contributions:

Name	Work	Time spent(Hours)
Prakruti P	Users, Report	2
Pruthvi P	Queries, transaction, Report	3
Raghav Pandit	Queries	2