UE19CS301 – DBMS Assignment 4

B R Pratheek – PES1UG19CS101
Arvind Krishna – PES1UG19CS090
Anurag Khanra – PES1UG19CS072

Ticket Booking System

Problem statement

Our DBMS application is designed to tackle the ticket booking system in theatres and cinemas.

Dependencies installed for the connectivity of our project:

- We used the PERN stack to handle all connectivity between the front-end and backend.
- pgPool was used to connect nodeJS to our PostgreSQL database.
- Our frontend interface was minimilastically designed using reactJS
- The backend framework was developed on ExpressJS and Node.JS was used as the JavaScript runtime environment.

Statement executed on the Front-end and features deployed:

- Customer Sign up/Log in Dashboard
- Customer can book a ticket from available running choices of movies and shows
- Theatre admin can add or remove movies and shows from the currently running list
- Different login/signup dashboard for theatre admin

```
tbs_assignment=# --user ticket booking
tbs_assignment=# SELECT cust_id FROM customer WHERE cust_name='Alex';
 cust_id
---------
 HNFOSA
(1 row)


tbs_assignment=# SELECT show_ID FROM shows WHERE movie_id = (SELECT movie_id FROM movie WHERE movie_name='Silentium');
 show_id
---------
 JSAIDH
(1 row)


tbs_assignment=# SELECT theatre_id FROM shows WHERE movie_id = (SELECT movie_id FROM movie WHERE movie_name='Silentium');
 theatre_id
------------
 A7VGBD
(1 row)


tbs_assignment=# SELECT discount FROM offer WHERE offer_id = '9HSDAU';
 discount
----------
       10
(1 row)


tbs_assignment=# INSERT INTO ticket (ticket_no, seat_no, price, offer_id, final_price, show_id, cust_id, theatre_id) VALUES ('ABC123', 24, 200 , '9HSDAU', 160 , 'JSAIDH' , 'HNFOSA', 'A7VGBD' ) returning *;
 ticket_no | seat_no | price | offer_id | final_price | show_id | cust_id | theatre_id
-----------+---------+-------+----------+-------------+---------+---------+------------
 ABC123    |      24 |   200 | 9HSDAU   |         160 | JSAIDH  | HNFOSA  | A7VGBD
(1 row)


INSERT 0 1
tbs_assignment=# UPDATE theatre set seats_available=seats_available-1 where theatre_id=('A7VGBD');
UPDATE 1
```

```
tbs_assignment=# --all movies running in a particular theatre
tbs_assignment=# SELECT movie_name,Q.release_date,language,start_time,end_time,screen_no FROM theatre NATURAL JOIN
tbs_assignment-# (SELECT * FROM movie NATURAL JOIN shows) AS Q WHERE theatre_name='Fun Zone';
            movie_name             | release_date | language | start_time | end_time | screen_no
-----------------------------------+--------------+----------+------------+----------+-----------
 Flicker                           | 7/24/2021    |        1 | 05:30      | 08:00    |         3
 Do You Wanna Know a Secret?       | 9/16/2021    |        1 | 13:45      | 16:40    |         6
 How to Meet Girls from a Distance | 10/14/2021   |        2 | 10:50      | 13:00    |         3
 spiderman                         | 10/10/2021   |        3 | 13:00      | 15:00    |         3
(4 rows)
```

```
tbs_assignment=# --all tickets booked by user
tbs_assignment=# SELECT movie_name,screen_no,ticket_no,seat_no,final_price FROM movie JOIN
tbs_assignment-# (SELECT * FROM shows INNER JOIN (SELECT * FROM customer NATURAL JOIN ticket) AS Q ON Q.show_id=shows.show_id) AS E
tbs_assignment-# ON E.movie_id=movie.movie_id where cust_name='Alex';
         movie_name          | screen_no | ticket_no | seat_no | final_price
-----------------------------+-----------+-----------+---------+-------------
 Flicker                     |         3 | YHSA8H    |      12 |         190
 Do You Wanna Know a Secret? |         6 | ASIDH1    |      26 |         190
 Fame                        |         1 | 2G7UF1    |       4 |         180
 Silentium                   |         4 | ABC123    |      24 |         160
(4 rows)
```

```
tbs_assignment=# --to display to user all movies currently running in all theatres
tbs_assignment=# SELECT theatre_name, movie_name,Q.release_date,language FROM theatre NATURAL JOIN
tbs_assignment-# (SELECT * FROM movie NATURAL JOIN shows) AS Q ORDER BY(theatre_name);
   theatre_name   |            movie_name             | release_date | language
------------------+-----------------------------------+--------------+----------
 Banglore cinemas | Fame                              | 8/23/2021    |        3
 Banglore cinemas | Silentium                         | 8/8/2021     |        1
 Banglore cinemas | How to Meet Girls from a Distance | 10/14/2021   |        5
 Banglore cinemas | Flicker                           | 7/24/2021    |        4
 Central Movies   | Horrible Dr. Hichcock, The        | 12/19/2020   |        2
 Central Movies   | Operation Mad Ball                | 11/17/2020   |        3
 Central Movies   | Operation Mad Ball                | 11/17/2020   |        5
 Fun Zone         | spiderman                         | 10/10/2021   |        3
 Fun Zone         | Do You Wanna Know a Secret?       | 9/16/2021    |        1
 Fun Zone         | How to Meet Girls from a Distance | 10/14/2021   |        2
 Fun Zone         | Flicker                           | 7/24/2021    |        1
(11 rows)
```

```
tbs_assignment=# --adding shows
tbs_assignment=# SELECT movie_id FROM movie WHERE movie_name='RRR';
 movie_id
----------
 F1NBAC
(1 row)


tbs_assignment=# SELECT theatre_id FROM theatre WHERE theatre_name='Fun Zone';
 theatre_id
------------
 8UHD7A
(1 row)


tbs_assignment=# INSERT INTO shows (start_time, end_time, show_id, language,screen_no, show_date,movie_id,theatre_id)
tbs_assignment-# VALUES ('13:00','16:00','UEFA21',3,5,'07/01/2021','F1NBAC','8UHD7A');
INSERT 0 1
```

```
tbs_assignment=# --adding movies
tbs_assignment=# INSERT INTO movie (movie_id , movie_name , director ,release_date) VALUES ('F1NBAC','RRR','Rajamouli','07/01/2021') returning *;
 movie_id | movie_name | director  | release_date
----------+------------+-----------+--------------
 F1NBAC   | RRR        | Rajamouli | 07/01/2021
(1 row)


INSERT 0 1
tbs_assignment=# INSERT INTO actors (Actor_name,Age,Sex,movie_id) VALUES ('NTR',38,'M','F1NBAC') returning *;
 actor_name | age | sex | movie_id
------------+-----+-----+----------
 NTR        |  38 | M   | F1NBAC
(1 row)
```

Migrating to a NoSQL DBMS:

Yes, it would be favourable to move the database system to a NoSQL format which uses DBMS software such as Neo4j. The steps undertaken in order for such a transition are:

a. Each row of each table in the RDBMS format is taken as an individual node in the graph database with a table representing a type of node.

b. Each customer node will be attached via edges/ links to all the relevant nodes such as the customer's tickets, bookings, shows, etc. and hence defining the view drawn for a customer user would become much simpler as one would only have to allow each node of type customer to access/read from all nodes directly adjacent to it (as opposed to the all table join used in the RDBMS case where we obtained all the tuples belonging to one customer by joining all the tables based on the customer key. The time required to perform such a join

operation is very large in large instances of the database setup which can be reduced by using the NoSQL approach.

In addition to these benefits, certain general-purpose benefits are obtained by using a graph based NoSQL DBMS which include:

a. Security – Since all the information related to one table is stored as multiple nodes of that particular type, sensitive information is less likely to get leaked as the information is distributed across multiple node objects rather than centralized in one physical base table as in the RDBMS case. Example – Login Passwords for customer.

b. Easier Concurrency control – Since each transaction done in the system creates a node and affects only the account node that did the transaction, Faster transactions and lesser wait times are involved as there is no shared resource that is used by all transactions in this case as opposed to the RDBMS case where the transaction and account tables are shared among all transactions and hence inserting into the transaction table followed by updating the account table would lead to significant wait times.