

# Tema 8 - Laborator

Aelenei Alex

# Cuprins

1	Introducere	2
2	Eroarea "mutating table"	2

# 1 Introducere

Diagrama conceptuală și diagrama entitate-relație utilizate în această temă sunt mai jos. Datele de test folosite sunt cele utilizate în Tema 1 și Tema 2.

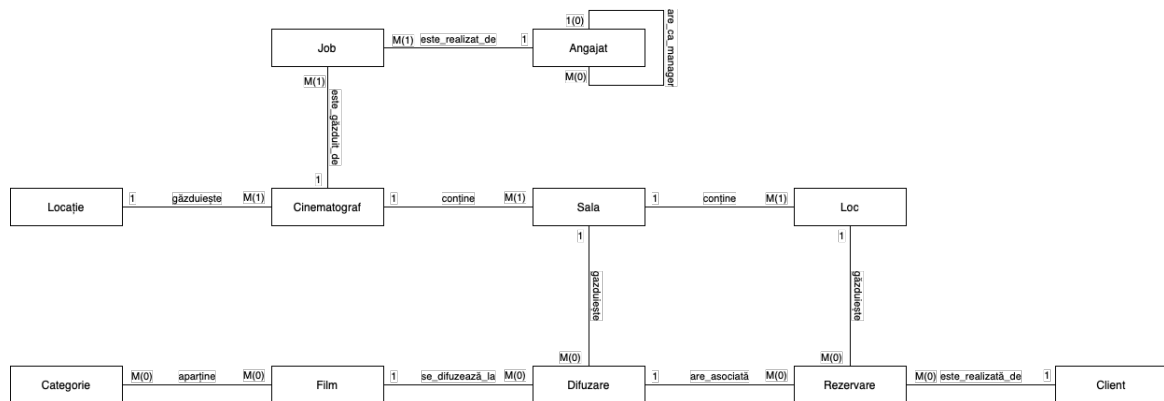


Figura 1: Diagrama conceptuală

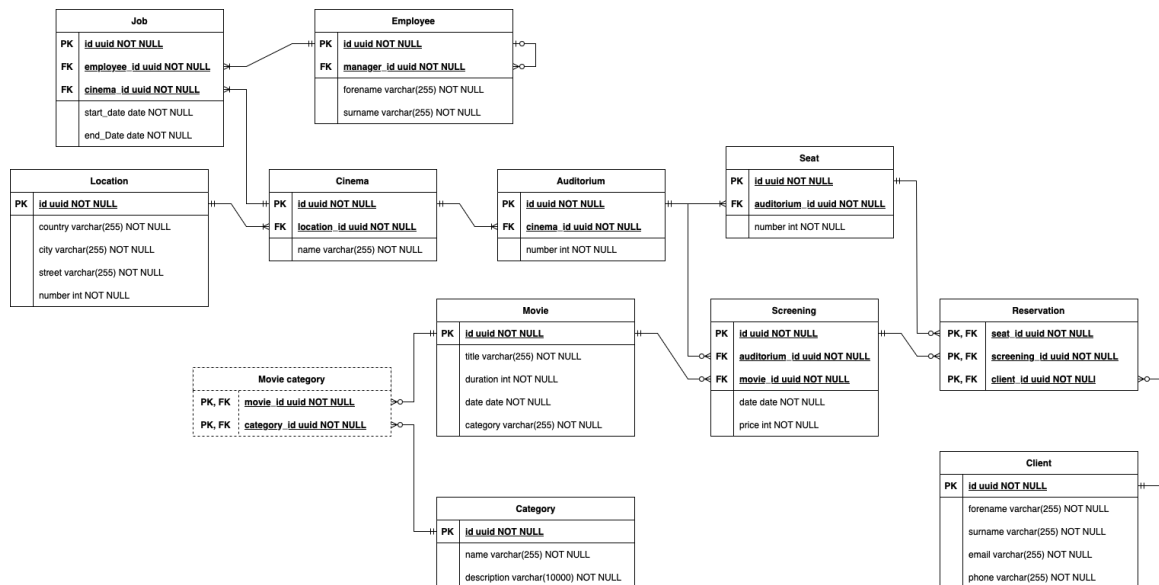


Figura 2: Diagrama entitate-relație

Aceste query-uri au fost rulate pe o instanță de Oracle Database 19, bazată pe imaginea oficială de la Oracle, dar cu un build pentru MacOS ARM local.

## 2 Eroarea "mutating table"

În general, eroarea "mutating table" (ORA-04091) apare când un trigger la nivel de linie încearcă să acceseze tot tabelul, ceea ce declanșează eroarea, pentru ca baza de date nu poate asigura un view consecvent.

Pentru a demonstra eroarea vom crea un tabel numit MOVIE\_STATISTICS in care vom memora pentru fiecare film numărul mediu de rezervări pentru difuzările filmului.

```

01 | CREATE TABLE MOVIE_STATISTICS AS (
02 |     SELECT
03 |         MOVIE_ID,
04 |         AVG(SEAT_COUNT) AS RESERVATIONS_PER_SCREENING_AVERAGE
05 |     FROM
06 |         (
07 |             SELECT
08 |                 S.SCREENING_ID,
09 |                 S.MOVIE_ID,
10 |                 COUNT(R.SEAT_ID) AS SEAT_COUNT
11 |             FROM
12 |                 RESERVATION R
13 |                 INNER JOIN SCREENING S
14 |                 ON R.SCREENING_ID = S.SCREENING_ID
15 |             GROUP BY
16 |                 S.SCREENING_ID,
17 |                 S.MOVIE_ID
18 |         )
19 |     GROUP BY
20 |         MOVIE_ID
21 | );

```

Listing 1: Query pentru crearea tabelului

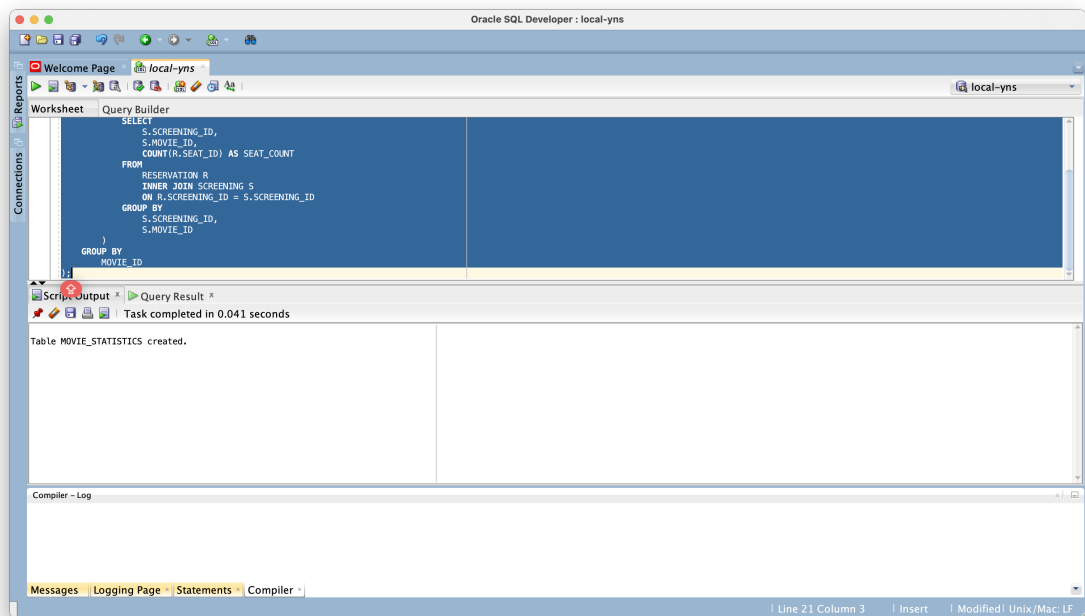


Figura 3: Rezultatul query-ului pentru crearea tabelului

De asemenea, pentru fiecare rezervare nouă creată vom reîmprospăta media pentru filmul asociat difuzării la care se face rezervarea. Un exemplu de astfel de trigger la nivel de linie este următorul:

```
01 | CREATE OR REPLACE TRIGGER UPDATE_MOVIE_STATISTICS AFTER
02 |     INSERT ON RESERVATION FOR EACH ROW
03 | DECLARE
04 |     ID_MOVIE MOVIE.MOVIE_ID%TYPE;
05 |     AVERAGE MOVIE_STATISTICS.RESERVATIONS_PER_SCREENING_AVERAGE%TYPE;
06 | BEGIN
07 |     SELECT
08 |         S.MOVIE_ID INTO ID_MOVIE
09 |     FROM
10 |         SCREENING S
11 |     WHERE
12 |         S.SCREENING_ID = :NEW.SCREENING_ID;
13 |     SELECT
14 |         AVG(SEAT_COUNT) INTO AVERAGE
15 |     FROM
16 |         (
17 |             SELECT
18 |                 S.SCREENING_ID ,
19 |                 COUNT(R.SEAT_ID) AS SEAT_COUNT
20 |             FROM
21 |                 SCREENING S
22 |                 INNER JOIN RESERVATION R
23 |                 ON R.SCREENING_ID = S.SCREENING_ID
24 |             WHERE
25 |                 S.MOVIE_ID = ID_MOVIE
26 |             GROUP BY
27 |                 S.SCREENING_ID
28 |         );
29 |     UPDATE MOVIE_STATISTICS
30 |     SET
31 |         RESERVATIONS_PER_SCREENING_AVERAGE = AVERAGE
32 |     WHERE
33 |         MOVIE_ID = ID_MOVIE;
34 | END;
```

Listing 2: Query pentru crearea trigger-ului la nivel de linie

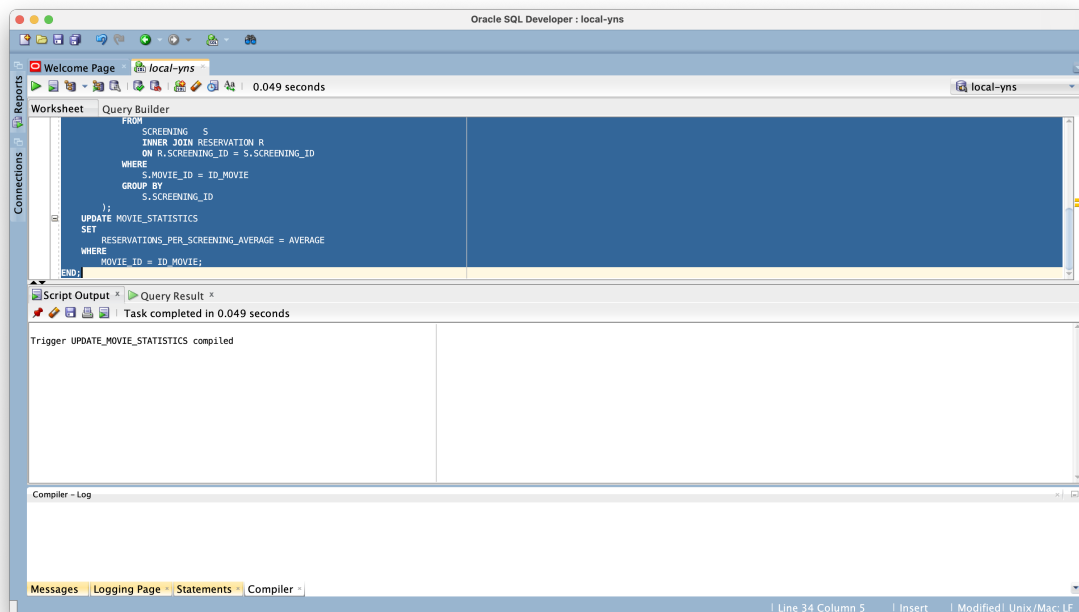


Figura 4: Rezultatul query-ului pentru crearea trigger-ului

Acest trigger generează eroarea "mutating table" deoarece încearcă să acceseze datele din tabel în timp ce îl updatează. Această eroare poate fi generată cu ușurință transformând trigger-ul de la nivel de linie la nivel de comandă.

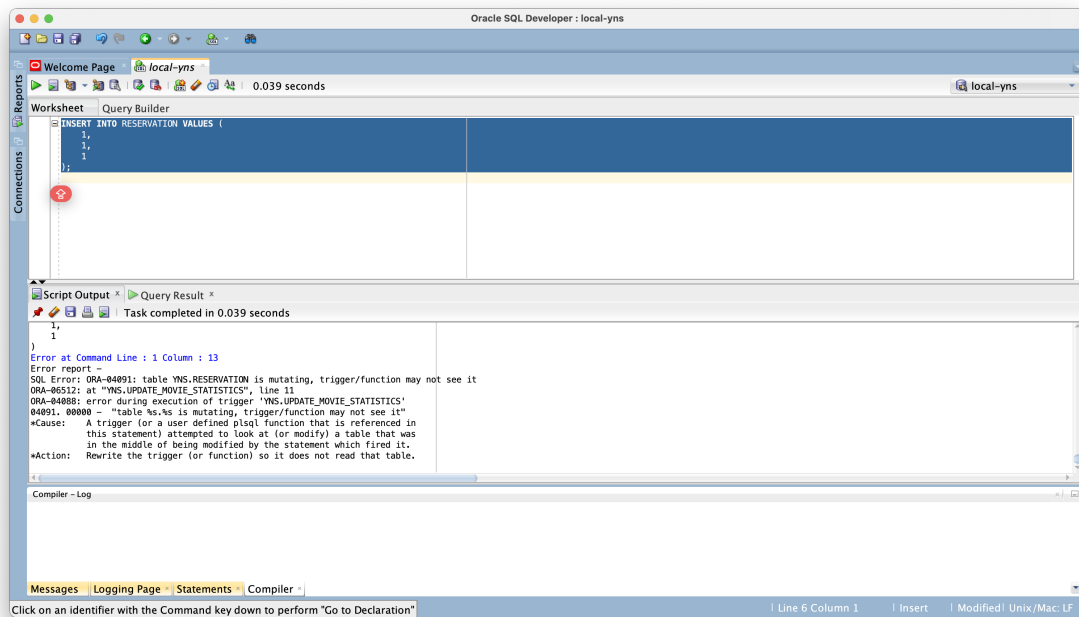


Figura 5: Rezultatul query-ului pentru crearea unei rezervări noi

```

01 | CREATE OR REPLACE TRIGGER UPDATE_MOVIE_STATISTICS AFTER
02 |     INSERT ON RESERVATION
03 | DECLARE
04 |     ID_MOVIE MOVIE.MOVIE_ID%TYPE;
05 |     AVERAGE MOVIE_STATISTICS.RESERVATIONS_PER_SCREENING_AVERAGE%TYPE;
06 | BEGIN
07 |     FOR ENTRY IN (
08 |         SELECT
09 |             MOVIE_ID,
10 |             AVG(SEAT_COUNT) AS AVERAGE
11 |         FROM
12 |             (
13 |                 SELECT
14 |                     S.SCREENING_ID,
15 |                     S.MOVIE_ID,
16 |                     COUNT(R.SEAT_ID) AS SEAT_COUNT
17 |                 FROM
18 |                     RESERVATION R
19 |                     INNER JOIN SCREENING S
20 |                     ON R.SCREENING_ID = S.SCREENING_ID
21 |                 GROUP BY
22 |                     S.SCREENING_ID,
23 |                     S.MOVIE_ID
24 |             )
25 |         GROUP BY
26 |             MOVIE_ID
27 |     ) LOOP
28 |         UPDATE MOVIE_STATISTICS
29 |         SET
30 |             RESERVATIONS_PER_SCREENING_AVERAGE=AVERAGE
31 |         WHERE
32 |             MOVIE_ID=ENTRY.MOVIE_ID;
33 |     END LOOP;
34 | END;

```

Listing 3: Query pentru crearea trigger-ului la nivel de comandă

Problema este astfel rezolvată și eroarea este evitată. Totuși, în trigger-ul la nivel de comandă nu avem acces la valorile :OLD și :NEW și deci nu putem realiza un update în baza acestor valori. Două soluții pentru această problemă sunt înlanțuirea mai multor trigger-uri, atât la nivel de linie cât și la nivel de comandă, sau utilizarea unui trigger compus, specific Oracle SQL.

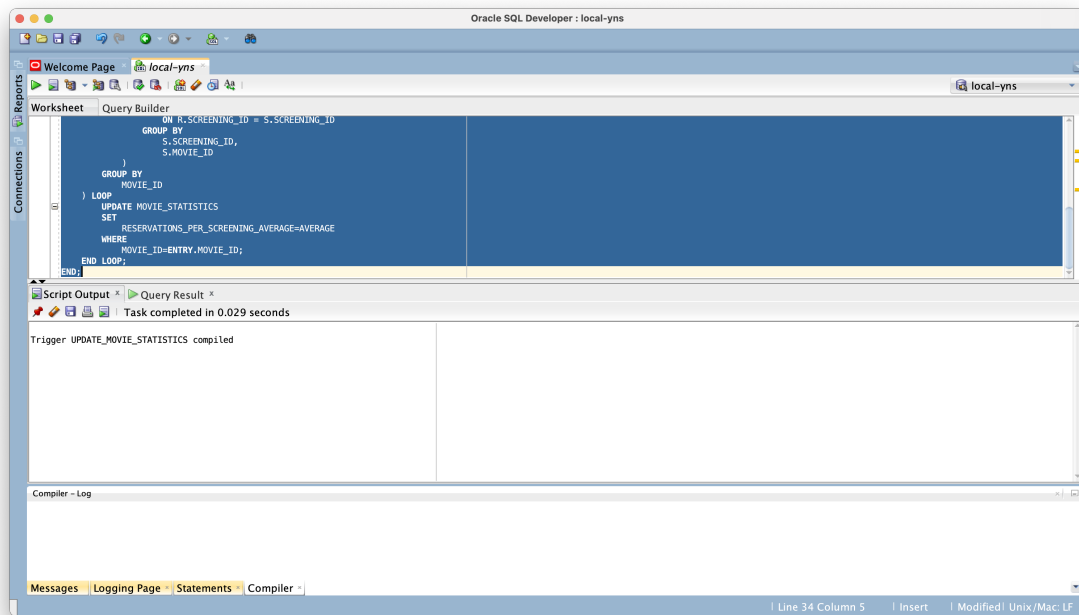


Figura 6: Rezultatul query-ului pentru crearea trigger-ului

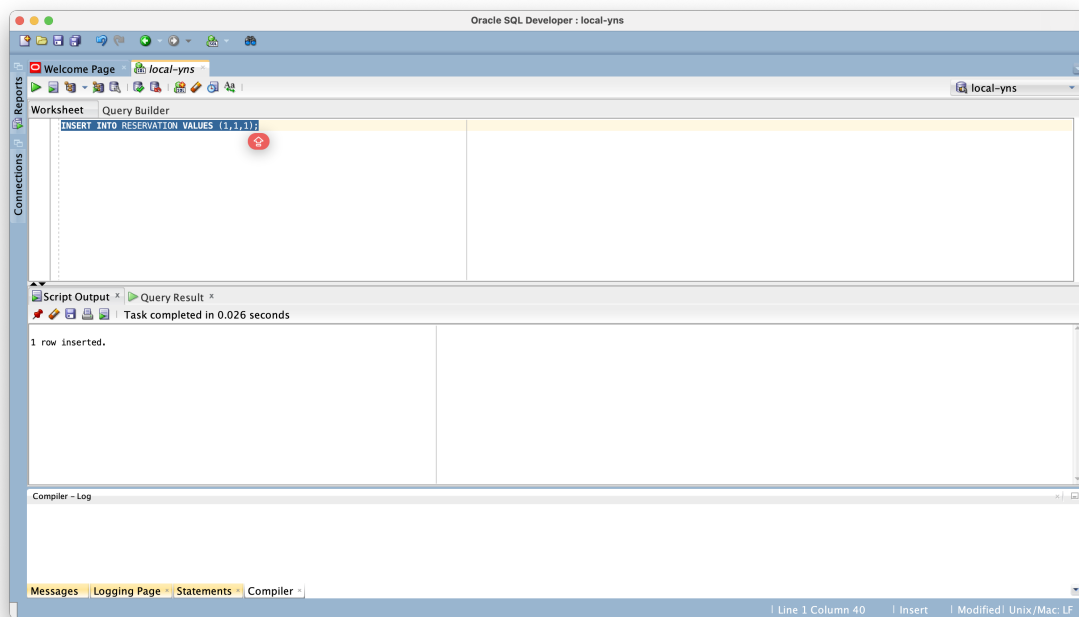


Figura 7: Rezultatul query-ului pentru crearea unei rezervări noi