

# Netflix

1. Резултати от създадените функции, приложени върху данните от база данни на NetflixDB.

- Изчислява средната оценка за даден филм по неговия идентификатор (movie\_id).

```
CREATE OR REPLACE FUNCTION AVG_MOVIE_RATING(p_movie_id NUMBER) RETURN
VARCHAR2 IS v_title VARCHAR2(200); v_avg_rating NUMBER(3,2); BEGIN SELECT
m.title, NVL(AVG(r.stars), 0) INTO v_title, v_avg_rating FROM Movie m LEFT JOIN Rating r
ON m.movie_id = r.movie_id WHERE m.movie_id = p_movie_id GROUP BY m.title;

RETURN v_title || '(ID = ' || p_movie_id || ')', средна оценка: ' || v_avg_rating;
```

```
EXCEPTION WHEN NO_DATA_FOUND THEN RETURN 'Няма данни за филм с ID = ' ||
p_movie_id; END; /
```

```
SELECT AVG_MOVIE_RATING(1) AS RESULT FROM DUAL;
```

RESULT	
1	Inception (ID = 1), средна оценка: 5

- Връща броя на изгледаните филми от даден потребител (user\_id), както и броя напълно завършени гледания.

```
CREATE OR REPLACE FUNCTION USER_WATCH_COUNT(p_user_id NUMBER) RETURN
VARCHAR2 IS v_name VARCHAR2(100); v_total NUMBER; v_completed NUMBER;
BEGIN SELECT u.name, COUNT(w.history_id), SUM(CASE WHEN w.completed = 'Y'
THEN 1 ELSE 0 END) INTO v_name, v_total, v_completed FROM Users u LEFT JOIN
WatchHistory w ON u.user_id = w.user_id WHERE u.user_id = p_user_id GROUP BY
u.name;
```

```
RETURN v_name || ' е гледал ' || v_total || ' филма, от които ' ||
v_completed || ' са изгледани докрай.';
```

```
EXCEPTION WHEN NO_DATA_FOUND THEN RETURN 'Няма намерен потребител с ID  
= ' || p_user_id; END; /  
  
SELECT USER_WATCH_COUNT(3) AS RESULT FROM DUAL;
```

 RESULT  
1 Georgi Kolev е гледал 1 филма, от които 1 са изгледани докрай.

- Извежда името на абонаментния план и броя потребители, които го използват.

```
CREATE OR REPLACE FUNCTION PLAN_TOTAL_USERS(p_plan_id NUMBER) RETURN  
VARCHAR2 IS v_plan_name VARCHAR2(100); v_count NUMBER; BEGIN SELECT  
s.name, COUNT(u.user_id) INTO v_plan_name, v_count FROM SubscriptionPlan s LEFT  
JOIN Users u ON s.plan_id = u.plan_id WHERE s.plan_id = p_plan_id GROUP BY s.name;  
  
RETURN 'План "' || v_plan_name || '" (ID = ' || p_plan_id ||  
) има ' || v_count || ' потребителя.';
```

```
EXCEPTION WHEN NO_DATA_FOUND THEN RETURN 'Няма намерен план с ID = ' ||  
p_plan_id; END; /  
  
SELECT PLAN_TOTAL_USERS(2) AS RESULT FROM DUAL;
```

 RESULT  
1 План "Standard" (ID = 2) има 1 потребителя.

**2. Резултати от създадените процедури, приложени върху данните от база данни на NetflixDB.**

- Извежда филмите с най-висока средна оценка (по данни от таблица Rating).

```
CREATE OR REPLACE PROCEDURE TOP_RATED_MOVIES IS BEGIN  
DBMS_OUTPUT.PUT_LINE('--- ТОП ФИЛМИ ПО СРЕДНА ОЦЕНКА ---'); FOR rec IN (  
SELECT m.title, ROUND(AVG(r.stars), 2) AS avg_rating, COUNT(r.rating_id) AS  
rating_count FROM Movie m JOIN Rating r ON m.movie_id = r.movie_id GROUP BY
```

```
m.title ORDER BY avg_rating DESC ) LOOP DBMS_OUTPUT.PUT_LINE( rec.title || '|  
Средна оценка: ' || rec.avg_rating || ' | Брой оценки: ' || rec.rating_count ); END LOOP;  
END; /
```

```
SET SERVEROUTPUT ON; EXEC TOP_RATED_MOVIES;
```

Procedure TOP\_RATED\_MOVIES compiled

```
--- ТОП ФИЛМИ ПО СРЕДНА ОЦЕНКА ---  
The Matrix | Средна оценка: 5 | Брой оценки: 1  
Parasite | Средна оценка: 5 | Брой оценки: 1  
Joker | Средна оценка: 5 | Брой оценки: 1  
Inception | Средна оценка: 5 | Брой оценки: 1  
The Batman | Средна оценка: 5 | Брой оценки: 1  
Dune | Средна оценка: 5 | Брой оценки: 1  
Interstellar | Средна оценка: 4 | Брой оценки: 1  
Tenet | Средна оценка: 4 | Брой оценки: 1  
Avatar | Средна оценка: 4 | Брой оценки: 1  
Oppenheimer | Средна оценка: 4 | Брой оценки: 1
```

- Извежда всички потребители, които са абонирани за даден план (по plan\_id).

```
CREATE OR REPLACE PROCEDURE USERS_BY_PLAN(p_plan_id NUMBER) IS BEGIN  
DBMS_OUTPUT.PUT_LINE('--- ПОТРЕБИТЕЛИ С АБОНАМЕНТЕН ПЛАН ---'); FOR rec IN  
( SELECT u.name, u.email, s.name AS plan_name FROM Users u JOIN SubscriptionPlan  
s ON u.plan_id = s.plan_id WHERE s.plan_id = p_plan_id ) LOOP  
DBMS_OUTPUT.PUT_LINE(rec.name || ' | ' || rec.email || ' | План: ' || rec.plan_name);  
END LOOP; END; /
```

```
SET SERVEROUTPUT ON; EXEC USERS_BY_PLAN(1);
```

Procedure USERS\_BY\_PLAN compiled

```
--- ПОТРЕБИТЕЛИ С АБОНАМЕНТЕН ПЛАН ---  
Ivan Ivanov | ivan@abv.bg | План: Basic
```

PL/SQL procedure successfully completed.

- Показва всички оценки и коментари за даден филм.

```

CREATE OR REPLACE PROCEDURE MOVIE_RATINGS_DETAILS(p_movie_id NUMBER) IS
v_title VARCHAR2(200); BEGIN SELECT title INTO v_title FROM Movie WHERE movie_id
= p_movie_id;

DBMS_OUTPUT.PUT_LINE('--- ОЦЕНКИ ЗА ФИЛМ: ' || v_title || ' ---');

FOR rec IN (
    SELECT u.name, r.stars, r.comment_text, TO_CHAR(r.rated_at, 'YYYY-MM-DD') AS
rated_date
    FROM Rating r
    JOIN Users u ON r.user_id = u.user_id
    WHERE r.movie_id = p_movie_id
    ORDER BY r.rated_at DESC
) LOOP
    DBMS_OUTPUT.PUT_LINE(
        rec.name || ' | Оценка: ' || rec.stars ||
        ' | "' || rec.comment_text || '" | ' || rec.rated_date
    );
END LOOP;

```

```

EXCEPTION WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Няма данни
за филм с ID = ' || p_movie_id); END; /

```

```
SET SERVEROUTPUT ON; EXEC MOVIE_RATINGS_DETAILS(1);
```

```

Procedure MOVIE_RATINGS_DETAILS compiled

--- ОЦЕНКИ ЗА ФИЛМ: Inception ---
Ivan Ivanov | Оценка: 5 | "Amazing movie!" | 2025-10-27

```

```
PL/SQL procedure successfully completed.
```

### 3. Резултати от създадените тригери, приложени върху данните от база данни на NetflixDB.

- При всяко добавяне на нова оценка в Rating, се извежда съобщение кой потребител е оценил кой филм и с колко звезди.

```

CREATE OR REPLACE TRIGGER TRG_RATING_ALERT AFTER INSERT ON Rating FOR
EACH ROW DECLARE v_user_name VARCHAR2(100); v_movie_title VARCHAR2(200);
BEGIN SELECT name INTO v_user_name FROM Users WHERE user_id = :NEW.user_id;
SELECT title INTO v_movie_title FROM Movie WHERE movie_id = :NEW.movie_id;

DBMS_OUTPUT.PUT_LINE('➡ Потребител "' || v_user_name ||
'" оцени филма "' || v_movie_title ||
'" с ' || :NEW.stars || ' звезди.');

END; /

```

```

SET SERVEROUTPUT ON; INSERT INTO Rating VALUES (11, 2, 1, 4, 'Very good!', SYSDATE);

```

Trigger TRG\_RATING\_ALERT compiled

→ Потребител "Maria Petrova" оцени филма "Inception" с 4 звезди.

1 row inserted.

- При нов запис в WatchHistory, показва съобщение, че потребител е гледал филм и дали го е изгледал докрай.

```

CREATE OR REPLACE TRIGGER TRG_WATCH_HISTORY_ALERT AFTER INSERT ON
WatchHistory FOR EACH ROW DECLARE v_user_name VARCHAR2(100); v_movie_title
VARCHAR2(200); BEGIN SELECT name INTO v_user_name FROM Users WHERE user_id =
:NEW.user_id; SELECT title INTO v_movie_title FROM Movie WHERE movie_id =
:NEW.movie_id;

```

```

IF :NEW.completed = 'Y' THEN
  DBMS_OUTPUT.PUT_LINE('🎥 ' || v_user_name || ' изгледа докрай "' || v_movie_title
|| '".');
ELSE
  DBMS_OUTPUT.PUT_LINE('⏸ ' || v_user_name || ' започна да гледа "' || v_movie_title
|| '".');

```

```
|| "", но не го завърши.');
```

```
END IF;
```

```
END; /
```

```
SET SERVEROUTPUT ON; INSERT INTO WatchHistory VALUES (11, 3, 2, SYSDATE, 80, 'N');
```

```
Trigger TRG_WATCH_HISTORY_ALERT compiled
```

```
■□ Georgi Kolev започна да гледа "Avatar", но не го завърши.
```

```
1 row inserted.
```

- Когато се изтриве потребител, извежда съобщение кой е бил изтрит и от коя държава.

```
CREATE OR REPLACE TRIGGER TRG_USER_DELETE_ALERT AFTER DELETE ON Users
FOR EACH ROW BEGIN DBMS_OUTPUT.PUT_LINE('⚠ Изтрит е потребител: ' ||
:OLD.name || ' (' || :OLD.email || '), държава: ' || :OLD.country); END; /
```

```
SET SERVEROUTPUT ON;
```

```
DELETE FROM Users WHERE user_id = 10;
```

```
Trigger TRG_USER_DELETE_ALERT compiled
```

```
⚠ Изтрит е потребител: Mihail Borisov (mihail@borisov.bg), държава: Bulgaria
```