

Question 1

```
DECLARE
full_name    VARCHAR2(100);
movie_count  NUMBER(3);
avg_score    movie.score%TYPE;
BEGIN

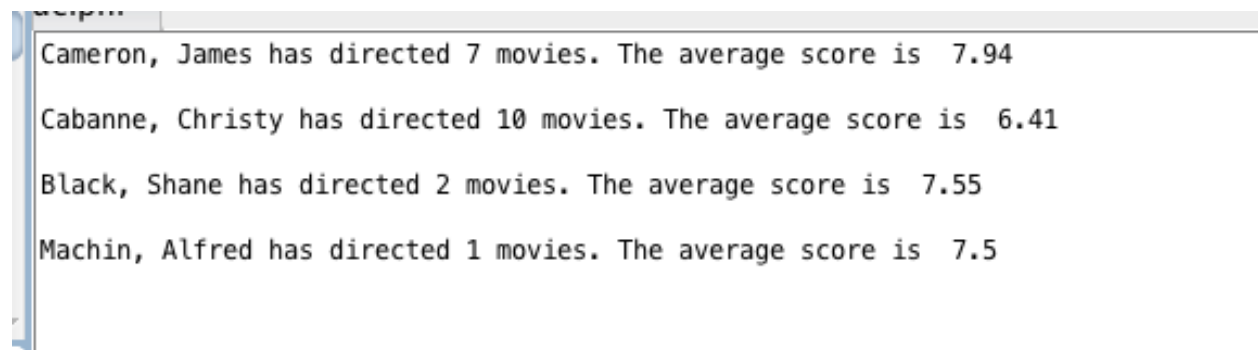
full_name:= &last_name || ', ' || &first_name;

SELECT ROUND(AVG(score),2) , COUNT(*)
INTO avg_score, movie_count
FROM movie
WHERE full_name = movie.director;

DBMS_OUTPUT.PUT_LINE(full_name || ' has directed ' || movie_count || ' movies. The average
score is ' || avg_score);

END;
```

The actors used were 1. Cameron, James 2.Cabanne, Christy. 3. Black, Shane 4. Machin, Alfred



```
Script
Cameron, James has directed 7 movies. The average score is 7.94
Cabanne, Christy has directed 10 movies. The average score is 6.41
Black, Shane has directed 2 movies. The average score is 7.55
Machin, Alfred has directed 1 movies. The average score is 7.5
```

Question 2

```
DECLARE
    m_year    NUMBER(4,0) := &m_year;
    i          NUMBER :=1;
CURSOR c_movie IS
    SELECT *
    FROM
        (
            SELECT movie.title, movie.votes, movie.score, actor.name
            FROM movie, casting, actor
            WHERE movie.ID = casting.movie_id AND casting.actor_ID = actor.id AND movie.yr =
m_year and casting.ord =1
            ORDER BY movie.votes DESC
        )
        WHERE rownum <=5;

BEGIN
    IF m_year >= 1990 AND m_year < 2013 THEN

        FOR m_movie IN c_movie
        LOOP

            DBMS_OUTPUT.PUT_LINE('Rank ' || i || ' : ' || m_movie.title || ' Votes: ' || m_movie.votes || '
Score: ' || m_movie.score || ' Lead Actor: ' || m_movie.name);
            i := i+1;
        END LOOP;

    ELSE
        DBMS_OUTPUT.PUT_LINE('Invalid Year');
    END IF;
END;
```

The years used were: 1997, 2008, 2010, and 2017

```

Rank 1 : Titanic Votes: 548078 Score: 7.7 Lead Actor: DiCaprio, Leonardo
Rank 2 : Good Will Hunting Votes: 399765 Score: 8.2 Lead Actor: Damon, Matt
Rank 3 : L.A. Confidential Votes: 312518 Score: 8.3 Lead Actor: Spacey, Kevin
Rank 4 : Men in Black Votes: 288409 Score: 7.2 Lead Actor: Jones, Tommy Lee
Rank 5 : Life is Beautiful Votes: 256412 Score: 8.6 Lead Actor: Benigni, Roberto

Rank 1 : The Dark Knight Votes: 1115076 Score: 9 Lead Actor: Bale, Christian
Rank 2 : WALL·E Votes: 462166 Score: 8.5 Lead Actor: Burt, Ben
Rank 3 : Iron Man Votes: 459539 Score: 7.9 Lead Actor: Downey Jr., Robert
Rank 4 : Slumdog Millionaire Votes: 456348 Score: 8.1 Lead Actor: Patel, Dev
Rank 5 : Gran Torino Votes: 394342 Score: 8.2 Lead Actor: Eastwood, Clint

Rank 1 : Inception Votes: 910334 Score: 8.8 Lead Actor: DiCaprio, Leonardo
Rank 2 : Shutter Island Votes: 476953 Score: 8.1 Lead Actor: DiCaprio, Leonardo
Rank 3 : Black Swan Votes: 400391 Score: 8.1 Lead Actor: Portman, Natalie
Rank 4 : Toy Story 3 Votes: 345082 Score: 8.5 Lead Actor: Hanks, Tom
Rank 5 : The Social Network Votes: 327021 Score: 7.8 Lead Actor: Eisenberg, Jesse

Invalid Year

```

Question 3

```

CREATE TABLE New_Movie(
id INTEGER NOT NULL,
title VARCHAR2 (100),
yr NUMBER(4,0),
score NUMBER,
votes INTEGER,
director VARCHAR2 (100),
CONSTRAINT New_Movie_PK PRIMARY KEY (id));

```

```

DECLARE
Row_amount NUMBER:=0;
CURSOR c_movie IS
    SELECT id,title,yr,score,votes,director
    FROM movie
    WHERE movie.votes >=50000;

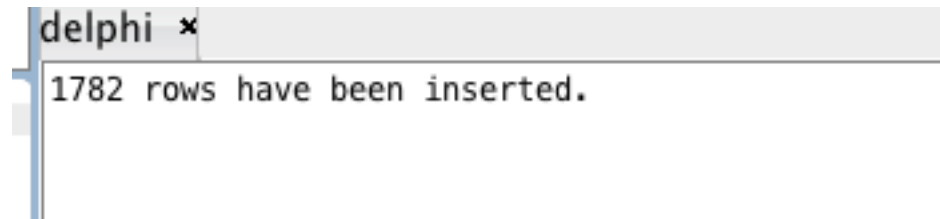
```

```

BEGIN
    FOR m_movie IN c_movie
    LOOP
        INSERT INTO New_movie VALUES(m_movie.ID, m_movie.title,
m_movie.yr,m_movie.score, m_movie.votes, m_movie.director);
        Row_amount:= Row_amount+1;
    END LOOP;
END

```

```
END LOOP;  
DBMS_OUTPUT.PUT_LINE(Row_amount || ' rows have been inserted.');
```



OR This way

```
DECLARE  
CURSOR c_movie IS  
  SELECT id,title,yr,score,votes,director  
  FROM movie  
  WHERE movie.votes >=50000;  
  number_total_rows NUMBER;  
  
BEGIN  
  FOR m_movie IN c_movie  
  LOOP  
    INSERT INTO New_movie VALUES(m_movie.ID, m_movie.title,  
m_movie.yr,m_movie.score, m_movie.votes, m_movie.director);  
    number_total_rows:=c_movie%ROWCOUNT;  
  END LOOP;  
  DBMS_OUTPUT.PUT_LINE( number_total_rows || ' rows have been inserted.');
```

question 4

```
Declare
cursor c_cursor IS SELECT * FROM (SELECT
actor.id,actor.name,COUNT(DISTINCT(casting.movie_id))AS Movie_Total FROM
Actor,Movie,Casting
WHERE actor.id=casting.actor_id AND movie.id=casting.movie_id AND YR >= 2000
GROUP BY actor.id,actor.name
ORDER BY Movie_Total DESC)
WHERE rownum <=10;
Begin
For v_record in c_cursor
Loop
For t in 2000..2009
Loop
SELECT COUNT(Title) AS Movie_count INTO v_record
FROM Movie,Casting
WHERE movie.id=casting.movie_id AND yr >= t and yr <=t+4 and actor_id=v_record.actor_id;

If v_record.Movie_count >=8 THEN
DBMS_OUTPUT.PUT_LINE('||v_record.name||' played in '||v_record.Movie_count||' movies
between'||yr||' and '||yr);
End Loop;
End Loop;
End;
```