

Question 1
Stored Function

```
CREATE OR REPLACE FUNCTION search_actor(input_str IN VARCHAR2)
RETURN INTEGER
IS
Total_count NUMBER;
V_actor_ID actor.id%TYPE;
BEGIN
SELECT count(name) INTO Total_count
FROM actor
WHERE name=input_str;
IF Total_count =0 THEN
RETURN 0;
ELSIF Total_count >0 THEN
select min(id) into V_actor_ID
FROM actor
where name=input_str;
RETURN v_actor_id;
END IF;
END;
```

SELECT search_actor('Cage, Nicolas') as output FROM dual;



The screenshot shows a 'Query Result' window with a toolbar containing icons for saving, printing, and refreshing. The status bar indicates 'All Rows Fetched: 1 in 0.029 seconds'. The 'OUTPUT' tab is active, displaying a single row with the value 242633.

OUTPUT	
1	242633

Stored Procedure

```
CREATE OR REPLACE PROCEDURE show_stats(v_actor_id IN INTEGER)
IS
v_Total_Count NUMBER;
v_Min_Score NUMBER;
v_Max_Score NUMBER;
v_Avg_Score NUMBER;
BEGIN
select COUNT(movie.id) AS Total_Movies,MIN(score) AS MIN_Score,MAX(score)AS
MAX_Score,ROUND(AVG(score),2) AS Average_Score
```

```

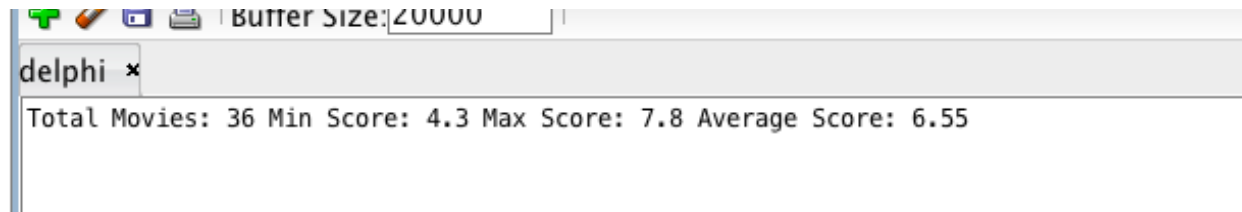
INTO v_Total_Count,v_Min_Score,v_Max_Score,v_Avg_Score
from actor,movie,casting
where actor.id=casting.actor_id AND movie.id=casting.movie_id AND actor_ID=v_actor_id;
IF v_Total_Count > 0 THEN
dbms_output.put_line('Total Movies: ' || v_Total_Count || ' Min Score: ' || v_Min_Score || '
Max Score: ' || v_Max_Score || ' Average Score: ' || v_Avg_Score);
ELSE
dbms_output.put_line('No Movie Record');
END IF;
END;

```

```

exec show_stats('242633');

```



Anonymous PL/SQL Block

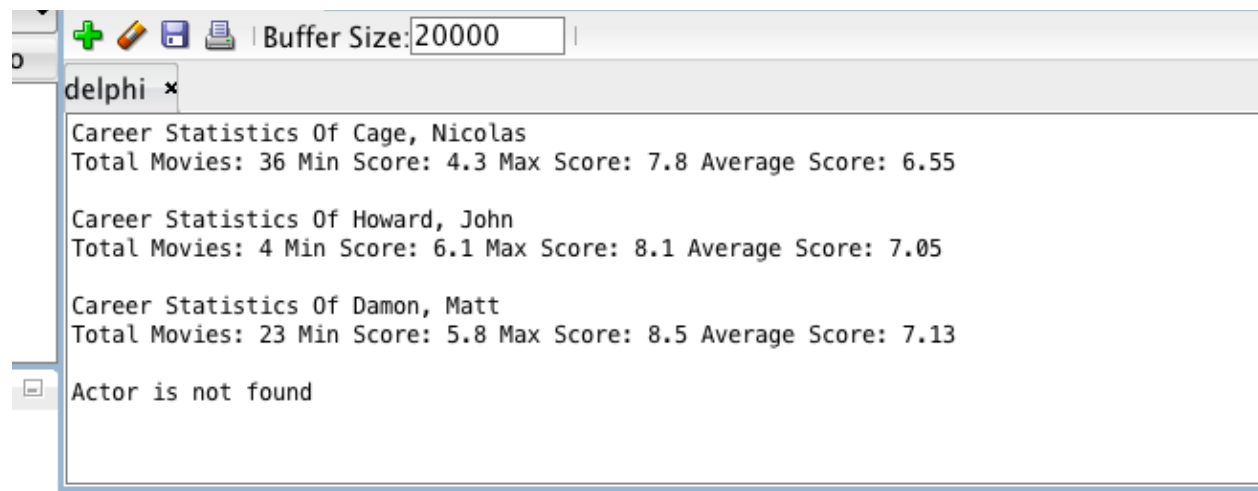
```

DECLARE
ID_type actor.id%Type;
New_ID INTEGER;
New_name actor.name%TYPE:=&user_input;
BEGIN
new_id:=search_actor(new_name);
IF new_id > 0 THEN
select id into ID_type
from actor
where id = new_id;
dbms_output.put_line('Career Statistics Of ' || New_name);
show_stats(ID_type);
ELSE
dbms_output.put_line('Actor is not found');

```

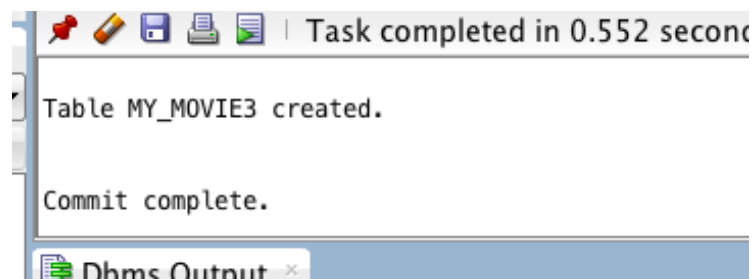
```
END IF;  
END;
```

User inputs: 'Cage, Nicolas' , 'Howard, John', 'Damon, Matt', and 'Pikachu'



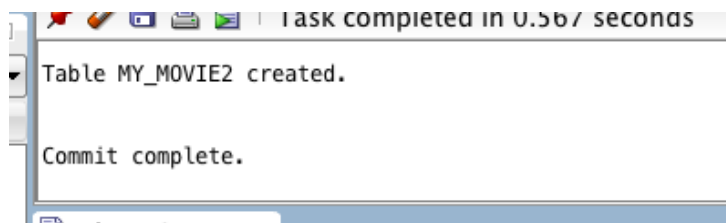
Question 2

```
create table my_movie3 as select * from movie where yr between 1985 and 2010;  
COMMIT;
```

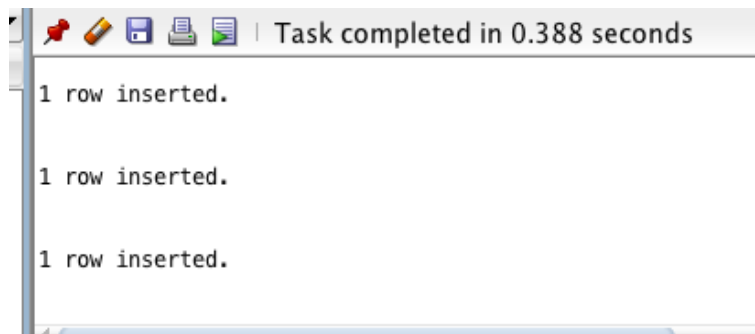


```
##
CREATE OR REPLACE TRIGGER movie_trig2
BEFORE INSERT ON my_movie3
FOR EACH ROW
BEGIN
  IF :NEW.score IS NULL THEN
    IF :NEW.yr < 1995 THEN
      :NEW.score := 8.0;
    ELSIF :NEW.yr >= 1995 THEN
      :NEW.score := 8.5;
    ELSE
      :NEW.score := :NEW.score;
    END IF;
  END IF;
END;

COMMIT;
```



```
INSERT INTO my_movie3 VALUES(99999,'Test Movie 1',1990,NULL, NULL, NULL);
INSERT INTO my_movie3 VALUES(99998,'Test Movie 2',2000,NULL, NULL, NULL);
INSERT INTO my_movie3 VALUES(99997,'Test Movie 3',1993,8.4, NULL, NULL);
```



```
select *
from my_movie3
where title='Test Movie 1' or title='Test Movie 2' OR title='Test Movie 3';
```

Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.033 seconds

	ID	TITLE	YR	SCORE	VOTES	DIRECTOR
1	99999	Test Movie 1	1990	8	(null)	(null)
2	99998	Test Movie 2	2000	8.5	(null)	(null)
3	99997	Test Movie 3	1993	8.4	(null)	(null)

```
BEGIN
  COMMIT;
  INSERT INTO my_movie3 VALUES(99999,'Test Movie 1',1990,NULL, NULL, NULL);
  INSERT INTO my_movie3 VALUES(99998,'Test Movie 2',2000,NULL, NULL, NULL);
  INSERT INTO my_movie3 VALUES(99997,'Test Movie 3',1993,8.4, NULL, NULL);
  COMMIT;
EXCEPTION
WHEN OTHERS THEN
ROLLBACK;
END;
```