

CSCI 5333 DBMS
Spring 2020
Homework #7

Views and Stored Subroutines

This assignment refers to the Sakila database that comes preloaded with MySQL installation. Refer to the documentation at: <http://dev.mysql.com/doc/sakila/en/>. For local development in your computer, download the database at <http://downloads.mysql.com/docs/sakila-db.zip> to review the SQL statements for declaring and populating the database. (Note that XAMPP does not include Sakila and you will need to install it yourself.)

(1) Write a view s20v1 so it can be used in the following manner.

```
mysql> SELECT *
      -> FROM s20v1
      -> WHERE actor_id = 11;
```

actor_id	actor	category	film
11	ZERO CAGE	Action	DANCES NONE
11	ZERO CAGE	Action	HANDICAP BOONDOCK
11	ZERO CAGE	Action	STORY SIDE
11	ZERO CAGE	Animation	CANYON STOCK
11	ZERO CAGE	Animation	HORN WORKING
11	ZERO CAGE	Children	JERSEY SASSY
11	ZERO CAGE	Children	TOOTSIE PILOT
11	ZERO CAGE	Children	UPTOWN YOUNG
11	ZERO CAGE	Comedy	VELVET TERMINATOR
11	ZERO CAGE	Documentary	MOD SECRETARY
11	ZERO CAGE	Documentary	THIN SAGEBRUSH
11	ZERO CAGE	Drama	RACER EGG
11	ZERO CAGE	Drama	WEST LION
11	ZERO CAGE	Family	GANDHI KWAI
11	ZERO CAGE	Foreign	MEET CHOCOLATE
11	ZERO CAGE	Games	ENCINO ELF
11	ZERO CAGE	Games	MOONWALKER FOOL
11	ZERO CAGE	Horror	STRANGERS GRAFFITI
11	ZERO CAGE	Music	OLEANDER CLUE
11	ZERO CAGE	New	ENDING CROWDS
11	ZERO CAGE	New	GODFATHER DIARY
11	ZERO CAGE	Sports	HONEY TIES
11	ZERO CAGE	Sports	IMAGE PRINCESS
11	ZERO CAGE	Sports	LOSER HUSTLER
11	ZERO CAGE	Travel	WORKER TARZAN

25 rows in set (0.04 sec)

(2) Provide the SQL statement to show all films with 13 or more actors in the following manner. You must use the view s20v1 defined in (1).

film	actors
BOONDOCK BALLROOM	[13 actors with ids]: 56; 64; 3; 88; 4; 98; 13; 119; 16; 184; 21; 190; 30,
CHITTY LOCK	[13 actors with ids]: 22; 178; 31; 184; 37; 188; 101; 149; 165; 5; 166; 20; 171,
CRAZY HOME	[13 actors with ids]: 114; 177; 125; 141; 149; 31; 158; 82; 161; 97; 168; 112; 173,
DRACULA CRYSTAL	[13 actors with ids]: 52; 137; 63; 155; 70; 197; 72; 83; 86; 2; 113; 41; 129,
LAMBS CINCINATTI	[15 actors with ids]: 37; 111; 45; 138; 47; 147; 53; 150; 61; 170; 75; 186; 81; 28; 102,
MUMMY CREATURES	[13 actors with ids]: 76; 190; 87; 102; 105; 17; 107; 25; 112; 32; 116; 39; 188,
RANDOM GO	[13 actors with ids]: 36; 166; 45; 182; 52; 200; 57; 65; 100; 4; 115; 20; 161,

7 rows in set (0.04 sec)

(3)

Write a function to return the *dot product* of the category film count vectors of two actors:

```
CREATE FUNCTION s20f1(  
    actorId_1 INT,  
    actorId_2 INT) RETURNS INT  
READS SQL DATA
```

The category film count vector of an actor is the vector of the numbers of films the actor appeared in the categories. For example, for actor #1, we have:

actor_id	category_id	num_films
1	1	0
1	2	1
1	3	1
1	4	2
1	5	1
1	6	1
1	7	0
1	8	2
1	9	1
1	10	2
1	11	3
1	12	1
1	13	2
1	14	1
1	15	1
1	16	0

This is because actor #1 appears in 0 film in category 1, 1 film in category 2, 1 film in category 3, 2 films in category 4, and so on. The category vector of actor #1 is thus (0,1,1,2,1,1,0,2,1,2,3,1,2,1,1,0). Note that there are 16 dimensions in the vector as there are 16 categories. Furthermore, the entries for categories 1, 7 and 16 are 0.

Similarly, for actor #2:

actor_id	category_id	num_films
2	1	1
2	2	1
2	3	1
2	4	2
2	5	2
2	6	1
2	7	1
2	8	4
2	9	2
2	10	1
2	11	0
2	12	1
2	13	4

2	14	2
2	15	0
2	16	2

The category vector of actor #2 is thus (1,1,1,2,2,1,1,4,2,1,0,1,4,2,0,2).

An example run:

```
mysql> SELECT s20f1(1,2);
+-----+
| s20f1(1,2) |
+-----+
|          32 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT s20f1(1,1);
+-----+
| s20f1(1,1) |
+-----+
|          33 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT s20f1(2,2);
+-----+
| s20f1(2,2) |
+-----+
|          59 |
+-----+
1 row in set (0.00 sec)
```

You may or may not use cursors in your function.

(4) Write a function s20f2 to return the *cosine similarity* of the category vectors of two actors. It can be used in a recommendation system. From Wikipedia:

Definition [\[edit\]](#)

The cosine of two non-zero vectors can be derived by using the [Euclidean dot product](#) formula:

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \|\mathbf{B}\| \cos \theta$$

Given two [vectors](#) of attributes, A and B , the cosine similarity, $\cos(\theta)$, is represented using a [dot product](#) and [magnitude](#) as

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

where A_i and B_i are [components](#) of vector A and B respectively.

For example:

```
CREATE FUNCTION s20f2(  
    actorId_1 INT,  
    actorId_2 INT  
) RETURNS FLOAT  
READS SQL DATA
```

For example:

```
mysql> SELECT s20f2(1,2);  
+-----+  
| s20f2(1,2) |  
+-----+  
| 0.7252153754234314 |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql>  
mysql> SELECT s20f2(1,3);  
+-----+  
| s20f2(1,3) |  
+-----+  
| 0.4222003221511841 |  
+-----+  
1 row in set (0.00 sec)
```

Submission:

Submit your homework through Blackboard with a sql file <<Yourname>><<YourStudentID>>_h7.sql containing all SQL statements. Your sql file should be executable directly under MySQL.