PROGRAM 9: MOVIE DATABASE

Consider the schema for Movie Database:

ACTOR(Act\_id, Act\_Name, Act\_Gender)

DIRECTOR(Dir\_id, Dir\_Name, Dir\_Phone)

MOVIES(Mov\_id, Mov\_Title, Mov\_Year, Mov\_Lang, Dir\_id)

MOVIE\_CAST(Act\_id, Mov\_id, Role)

RATING(Mov\_id, Rev\_Stars)

mysql> create database movie;

Query OK, 1 row affected (0.36 sec)

mysql> use movie

Database changed

mysql> create table actor(

-> act\_id int primary key,

-> act\_name varchar(20),

-> act\_gender char(10));

Query OK, 0 rows affected (1.81 sec)

mysql> describe actor;

+------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+------------+-------------+------+-----+---------+-------+

| act\_id | int | NO | PRI | NULL | |

| act\_name | varchar(20) | YES | | NULL | |

| act\_gender | char(10) | YES | | NULL | |

+------------+-------------+------+-----+---------+-------+

3 rows in set (0.08 sec)

mysql> create table director(

-> dir\_id int primary key,

-> dir\_name varchar(20),

-> dir\_phone int);

Query OK, 0 rows affected (0.65 sec)

mysql> describe director;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| dir\_id | int | NO | PRI | NULL | |

| dir\_name | varchar(20) | YES | | NULL | |

| dir\_phone | int | YES | | NULL | |

+-----------+-------------+------+-----+---------+-------+

3 rows in set (0.12 sec)

mysql> create table movies(

-> mov\_id int primary key,

-> mov\_title varchar(30),

-> mov\_year int,

-> mov\_lang varchar(20),

-> dir\_id int,

-> foreign key(dir\_id) references director(dir\_id) on update cascade on delete cascade);

Query OK, 0 rows affected (2.15 sec)

mysql> describe movies;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| mov\_id | int | NO | PRI | NULL | |

| mov\_title | varchar(30) | YES | | NULL | |

| mov\_year | int | YES | | NULL | |

| mov\_lang | varchar(20) | YES | | NULL | |

| dir\_id | int | YES | MUL | NULL | |

+-----------+-------------+------+-----+---------+-------+

5 rows in set (0.04 sec)

mysql> create table movie\_cast(

-> act\_id int,

-> mov\_id int,

-> role varchar(20),

-> foreign key(act\_id) references actor(act\_id) on update cascade on delete cascade,

-> foreign key(mov\_id) references movies(mov\_id) on update cascade on delete cascade);

Query OK, 0 rows affected (2.37 sec)

mysql> describe movies;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| mov\_id | int | NO | PRI | NULL | |

| mov\_title | varchar(30) | YES | | NULL | |

| mov\_year | int | YES | | NULL | |

| mov\_lang | varchar(20) | YES | | NULL | |

| dir\_id | int | YES | MUL | NULL | |

+-----------+-------------+------+-----+---------+-------+

5 rows in set (0.04 sec)

mysql> describe movie\_cast;

+--------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------+-------+

| act\_id | int | YES | MUL | NULL | |

| mov\_id | int | YES | MUL | NULL | |

| role | varchar(20) | YES | | NULL | |

+--------+-------------+------+-----+---------+-------+

3 rows in set (0.01 sec)

mysql> create table rating(

-> mov\_id int,

-> rev\_stars varchar(10),

-> foreign key(mov\_id) references movies(mov\_id));

Query OK, 0 rows affected (0.93 sec)

mysql> describe rating;

+-----------+------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+------+------+-----+---------+-------+

| mov\_id | int | YES | MUL | NULL | |

| rev\_stars | varchar(10) | YES | | NULL | |

+-----------+------+------+-----+---------+-------+

2 rows in set (0.14 sec)

mysql> insert into actor values

-> (301,'ANUSHKA','F'),

-> (302,'PRABHAS','M'),

-> (303,'PUNEETH','M'),

-> (304,'JERMY','M');

Query OK, 4 rows affected (0.25 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM ACTOR;

+--------+----------+------------+

| act\_id | act\_name | act\_gender |

+--------+----------+------------+

| 301 | ANUSHKA | F |

| 302 | PRABHAS | M |

| 303 | PUNEETH | M |

| 304 | JERMY | M |

+--------+----------+------------+

4 rows in set (0.00 sec)

mysql> INSERT INTO DIRECTOR VALUES

-> (60,'RAJAMOULI', 8751611001),

-> (61,'HITCHCOCK', 7766138911),

-> (62,'FARAN', 9986776531),

-> (63,'STEVEN SPIELBERG', 8989776530);

Query OK, 4 rows affected (0.14 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM DIRECTOR;

+--------+------------------+------------+

| dir\_id | dir\_name | DIR\_PHONE |

+--------+------------------+------------+

| 60 | RAJAMOULI | 8751611001 |

| 61 | HITCHCOCK | 7766138911 |

| 62 | FARAN | 9986776531 |

| 63 | STEVEN SPIELBERG | 8989776530 |

+--------+------------------+------------+

4 rows in set (0.00 sec)

mysql> SELECT \* FROM DIRECTOR;

+--------+------------------+------------+

| dir\_id | dir\_name | DIR\_PHONE |

+--------+------------------+------------+

| 60 | RAJAMOULI | 8751611001 |

| 61 | HITCHCOCK | 7766138911 |

| 62 | FARAN | 9986776531 |

| 63 | STEVEN SPIELBERG | 8989776530 |

+--------+------------------+------------+

4 rows in set (0.00 sec)

mysql> INSERT INTO MOVIES VALUES

-> (1001,'BAHUBALI-2',2017,'TELAGU', 60),

-> (1002,'BAHUBALI-1', 2015,'TELAGU',60),

-> (1003,'AKASH', 2008,'KANNADA',61),

-> (1004,'WAR HORSE',2011,'ENGLISH',63);

Query OK, 4 rows affected (0.19 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM MOVIES;

+--------+------------+----------+----------+--------+

| mov\_id | mov\_title | mov\_year | mov\_lang | dir\_id |

+--------+------------+----------+----------+--------+

| 1001 | BAHUBALI-2 | 2017 | TELAGU | 60 |

| 1002 | BAHUBALI-1 | 2015 | TELAGU | 60 |

| 1003 | AKASH | 2008 | KANNADA | 61 |

| 1004 | WAR HORSE | 2011 | ENGLISH | 63 |

+--------+------------+----------+----------+--------+

4 rows in set (0.00 sec)

mysql> INSERT INTO MOVIE\_CAST VALUES

-> (301, 1002,'HEROINE'),

-> (301, 1001,'HEROINE'),

-> (303, 1003,'HERO'),

-> (303, 1002,'GUEST'),

-> (304, 1004,'HERO');

Query OK, 5 rows affected (0.30 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM MOVIE\_CAST;

+--------+--------+---------+

| act\_id | mov\_id | role |

+--------+--------+---------+

| 301 | 1002 | HEROINE |

| 301 | 1001 | HEROINE |

| 303 | 1003 | HERO |

| 303 | 1002 | GUEST |

| 304 | 1004 | HERO |

+--------+--------+---------+

5 rows in set (0.00 sec)

mysql> INSERT INTO RATING VALUES

-> (1001,4),

-> (1002,2),

-> (1003, 5),

-> (1004, 4);

Query OK, 4 rows affected (0.20 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM RATING;

+--------+-----------+

| mov\_id | rev\_stars |

+--------+-----------+

| 1001 | 4 |

| 1002 | 2 |

| 1003 | 5 |

| 1004 | 4 |

+--------+-----------+

4 rows in set (0.00 sec)

i. List the titles of all movies directed by ‘Hitchcock’.

mysql> SELECT MOV\_TITLE FROM MOVIES WHERE DIR\_ID=(SELECT DIR\_ID FROM DIRECTOR WHERE DIR\_NAME='HITCHCOCK');

+-----------+

| MOV\_TITLE |

+-----------+

| AKASH |

+-----------+

1 row in set (0.07 sec)

ii. Find the movie names where one or more actors acted in two or more movies.

mysql> select mov\_title from movies m,movie\_cast mv where m.mov\_id=mv.mov\_id and act\_id in(select act\_id from movie\_cast group by act\_id having count(act\_id)>1)group by mov\_title having count(\*)>1;

+------------+

| mov\_title |

+------------+

| BAHUBALI-1 |

+------------+

1 row in set (0.08 sec)

iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

mysql> select act\_name,mov\_title,mov\_year from actor a join movie\_cast b on a.act\_id=b.act\_id join movies c on b.mov\_id=c.mov\_id where mov\_year not between 2000 and 2015;

+----------+------------+----------+

| act\_name | mov\_title | mov\_year |

+----------+------------+----------+

| ANUSHKA | BAHUBALI-2 | 2017 |

+----------+------------+----------+

1 row in set (0.07 sec)

iv. Find the title of movies and number of stars for each movie that has at least one rating and find the highest

mysql> SELECT MOV\_TITLE, MAX(REV\_STARS)

-> FROM MOVIES

-> INNER JOIN RATING USING (MOV\_ID) GROUP

-> BY MOV\_TITLE

-> HAVING MAX(REV\_STARS)>0

-> ORDER BY MOV\_TITLE;

+------------+----------------+

| MOV\_TITLE | MAX(REV\_STARS) |

+------------+----------------+

| AKASH | 5 |

| BAHUBALI-1 | 2 |

| BAHUBALI-2 | 4 |

| WAR HORSE | 4 |

+------------+----------------+

4 rows in set (0.00 sec)

number of stars that movie received. Sort the result by movie title.

v. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

mysql> UPDATE RATING SET REV\_STARS=5

-> WHERE MOV\_ID IN(SELECT MOV\_ID FROM MOVIES WHERE DIR\_ID IN(

-> SELECT DIR\_ID FROM DIRECTOR WHERE DIR\_NAME='STEVEN SPEILBERG'));