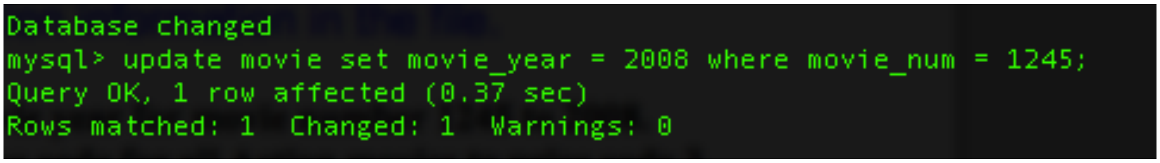
Jacob Reza

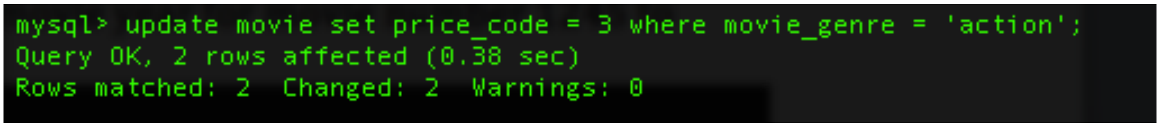
- 002408586
- CS325-11455-01
- CS325 Database Management Systems
- 11-20-14

1. Write the SQL command to change the movie year for movie number 1245 to 2008.



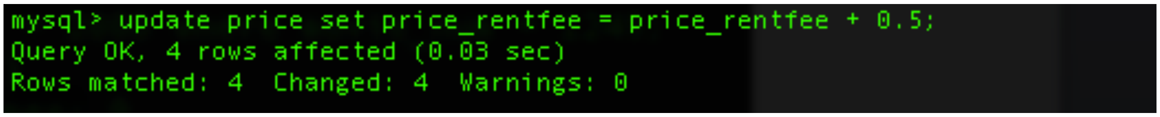
```
Database changed
mysql> update movie set movie_year = 2008 where movie_num = 1245;
Query OK, 1 row affected (0.37 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

2. Write the SQL command to change the price code for all Action movies to price code 3.



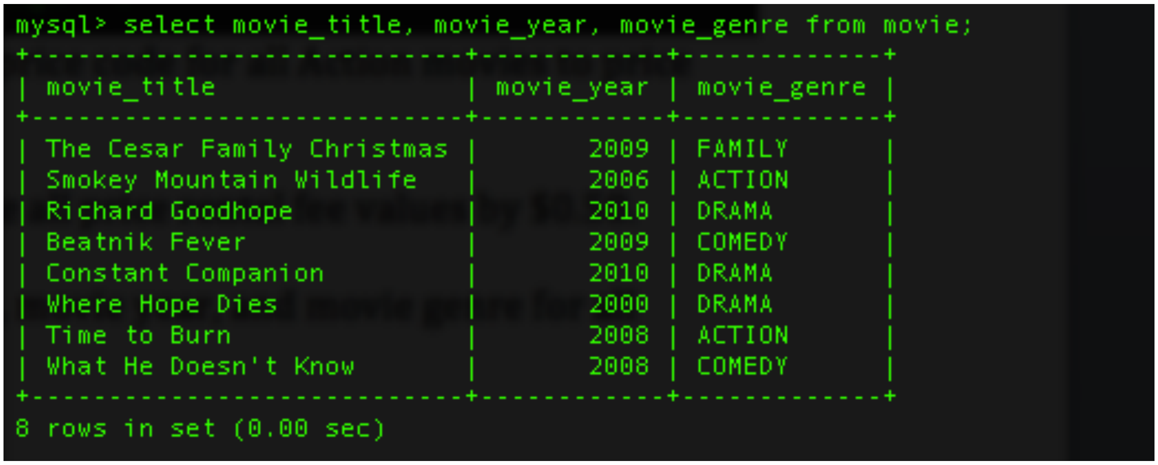
```
mysql> update movie set price_code = 3 where movie_genre = 'action';
Query OK, 2 rows affected (0.38 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

3. Write a single SQL command to increase all price rental fee values by $0.50.



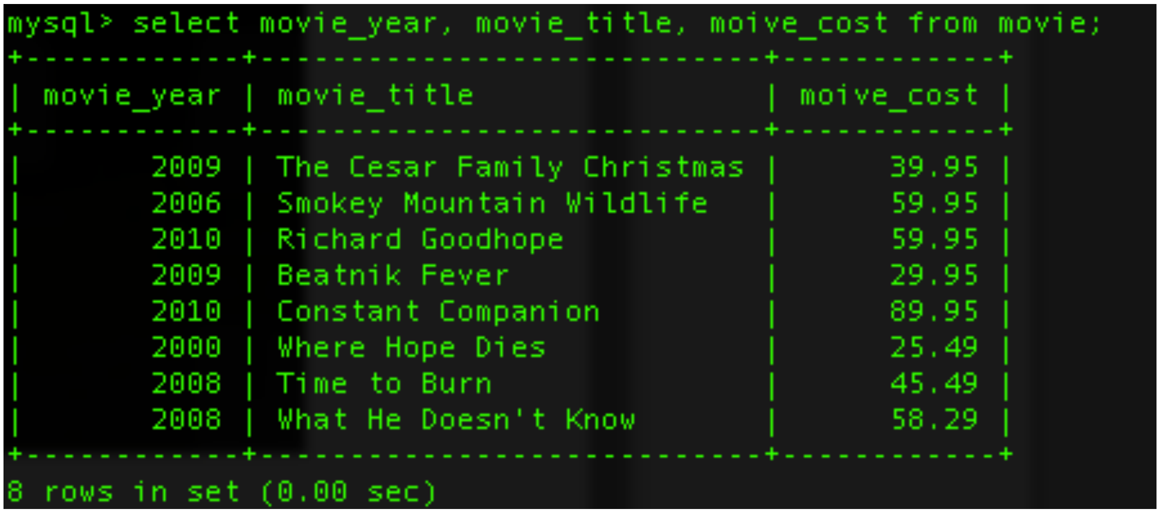
```
mysql> update price set price_rentfee = price_rentfee + 0.5;
Query OK, 4 rows affected (0.03 sec)
Rows matched: 4  Changed: 4  Warnings: 0
```

4. Write a query to display the movie title, movie year, and movie genre for all movies (result shown in Figure P4.4).



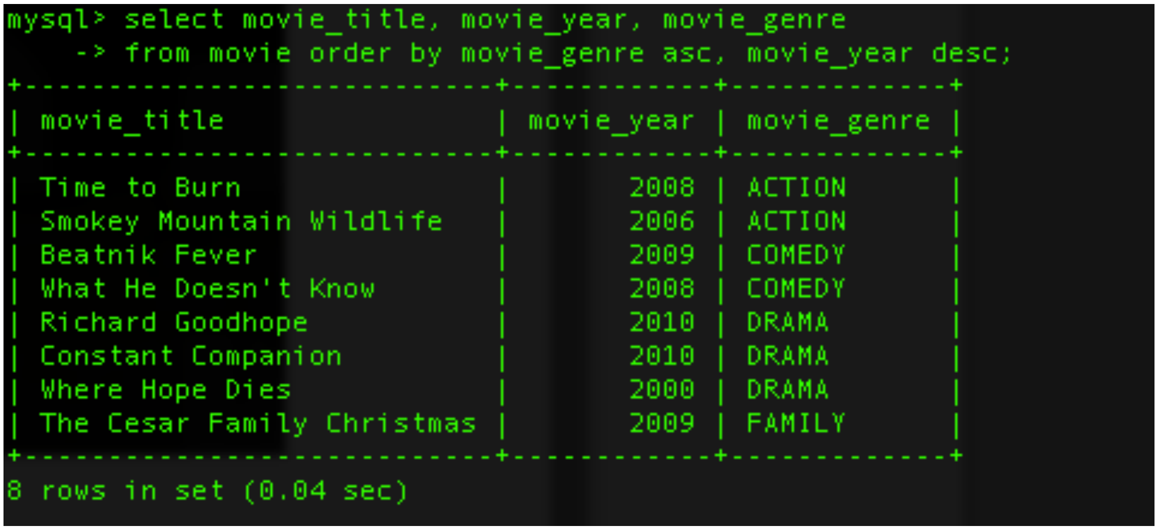
```
mysql> select movie_title, movie_year, movie_genre from movie;
+----------------------------+------------+-------------+
| movie_title                | movie_year | movie_genre |
+----------------------------+------------+-------------+
| The Cesar Family Christmas |       2009 | FAMILY      |
| Smokey Mountain Wildlife   |       2006 | ACTION      |
| Richard Goodhope           |       2010 | DRAMA       |
| Beatnik Fever              |       2009 | COMEDY      |
| Constant Companion         |       2010 | DRAMA       |
| Where Hope Dies            |       2000 | DRAMA       |
| Time to Burn               |       2008 | ACTION      |
| What He Doesn't Know       |       2008 | COMEDY      |
+----------------------------+------------+-------------+
8 rows in set (0.00 sec)
```

5. Write a query to display the movie year, movie title, and movie cost sorted by movie year in descending order (result shown in Figure P4.5). Figure P4.5 Movies by year



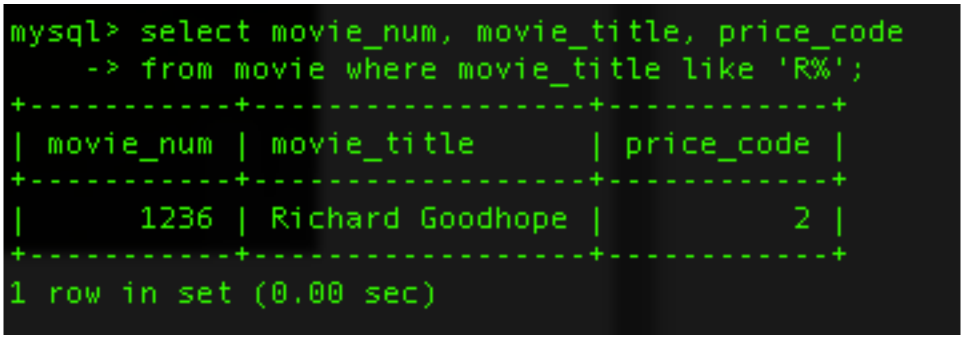
```
mysql> select movie_year, movie_title, moive_cost from movie;
+------------+----------------------------+------------+
| movie_year | movie_title                | moive_cost |
+------------+----------------------------+------------+
|       2009 | The Cesar Family Christmas |      39.95 |
|       2006 | Smokey Mountain Wildlife   |      59.95 |
|       2010 | Richard Goodhope           |      59.95 |
|       2009 | Beatnik Fever              |      29.95 |
|       2010 | Constant Companion         |      89.95 |
|       2000 | Where Hope Dies            |      25.49 |
|       2008 | Time to Burn               |      45.49 |
|       2008 | What He Doesn't Know       |      58.29 |
+------------+----------------------------+------------+
8 rows in set (0.00 sec)
```

6. Write a query to display the movie title, movie year, and movie genre for all movies sorted by movie genre in ascending order, then sorted by movie year in descending order within genre (result shown in Figure P4.6). Figure P4.6 Movies with multicolumn sort



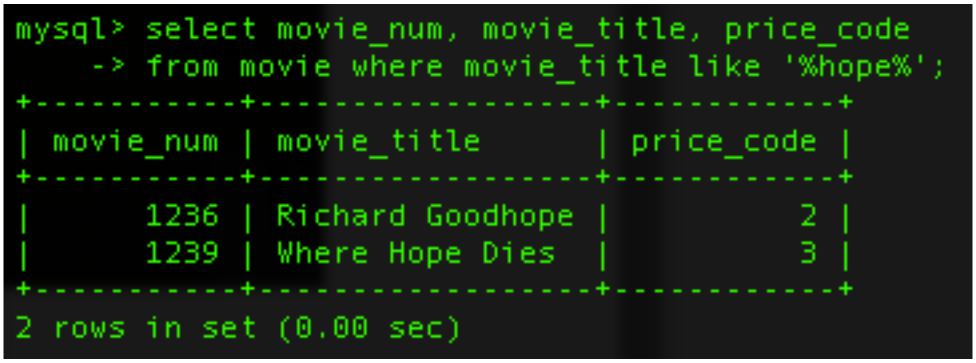
```
mysql> select movie_title, movie_year, movie_genre
    -> from movie order by movie_genre asc, movie_year desc;
+----------------------------+------------+-------------+
| movie_title                | movie_year | movie_genre |
+----------------------------+------------+-------------+
| Time to Burn               |       2008 | ACTION      |
| Smokey Mountain Wildlife   |       2006 | ACTION      |
| Beatnik Fever              |       2009 | COMEDY      |
| What He Doesn't Know       |       2008 | COMEDY      |
| Richard Goodhope           |       2010 | DRAMA       |
| Constant Companion         |       2010 | DRAMA       |
| Where Hope Dies            |       2000 | DRAMA       |
| The Cesar Family Christmas |       2009 | FAMILY      |
+----------------------------+------------+-------------+
8 rows in set (0.04 sec)
```

7. Write a query to display the movie number, movie title, and price code for all movies with a title that starts with the letter “R” (result shown in Figure P4.7). Figure P4.7 Movies starting with RCS 325 MySQL Project #4 3



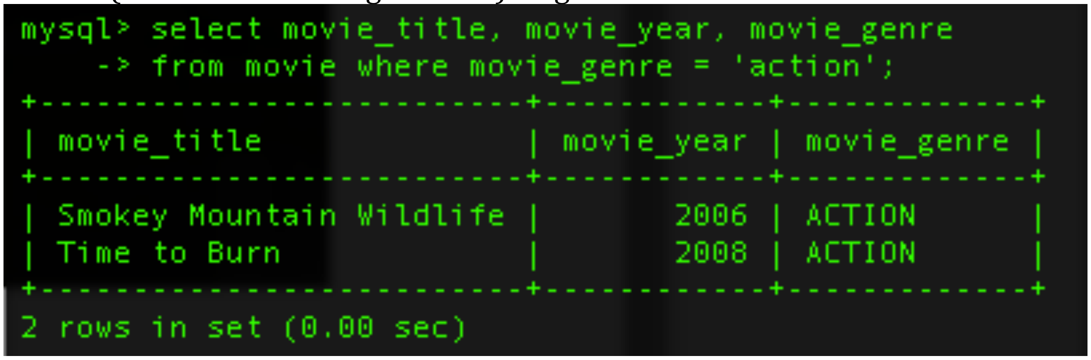
```
mysql> select movie_num, movie_title, price_code
    -> from movie where movie_title like 'R%';
+-----------+------------------+------------+
| movie_num | movie_title      | price_code |
+-----------+------------------+------------+
|      1236 | Richard Goodhope |          2 |
+-----------+------------------+------------+
1 row in set (0.00 sec)
```

8. Write a query to display the movie title, movie year, and movie cost for all movies that contain the word "hope" anywhere in the title. Sort the results in ascending order by title (result shown in figure P4.8).
Figure P4.8 Movies with "Hope" in the title



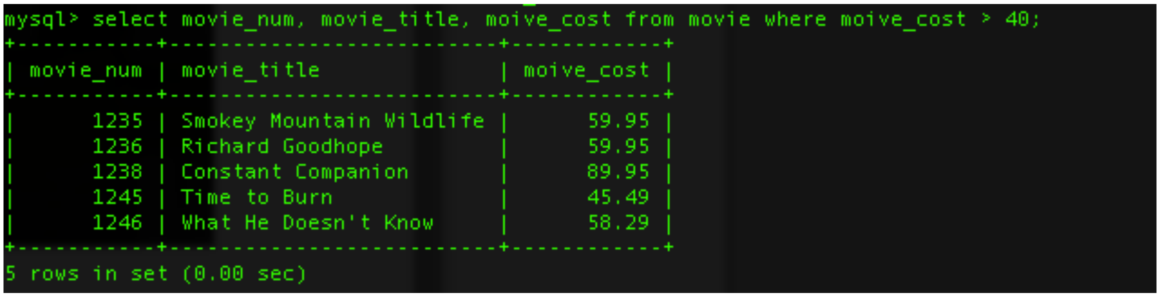
```
mysql> select movie_num, movie_title, price_code
    -> from movie where movie_title like '%hope%';
+-----------+------------------+------------+
| movie_num | movie_title      | price_code |
+-----------+------------------+------------+
|      1236 | Richard Goodhope |          2 |
|      1239 | Where Hope Dies  |          3 |
+-----------+------------------+------------+
2 rows in set (0.00 sec)
```

9. Write a query to display the movie title, movie year, and movie genre for all action movies (result shown in Figure P4.9). Figure P4.9 Action movies



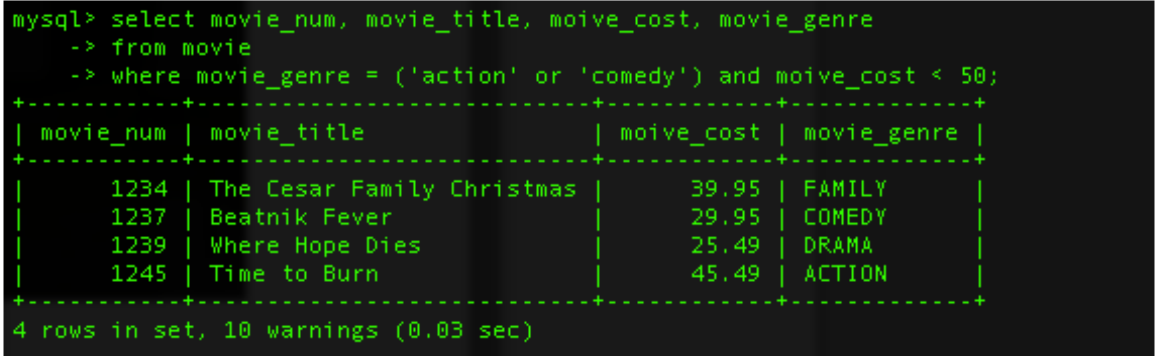
```
mysql> select movie_title, movie_year, movie_genre
    -> from movie where movie_genre = 'action';
+--------------------------+------------+-------------+
| movie_title              | movie_year | movie_genre |
+--------------------------+------------+-------------+
| Smokey Mountain Wildlife |       2006 | ACTION      |
| Time to Burn             |       2008 | ACTION      |
+--------------------------+------------+-------------+
2 rows in set (0.00 sec)
```

10. Write a query to display the movie number, movie title, and movie cost for all movies with a cost greater than $40 (result shown in Figure P4.10).
P4.10 Movies costing more than $40



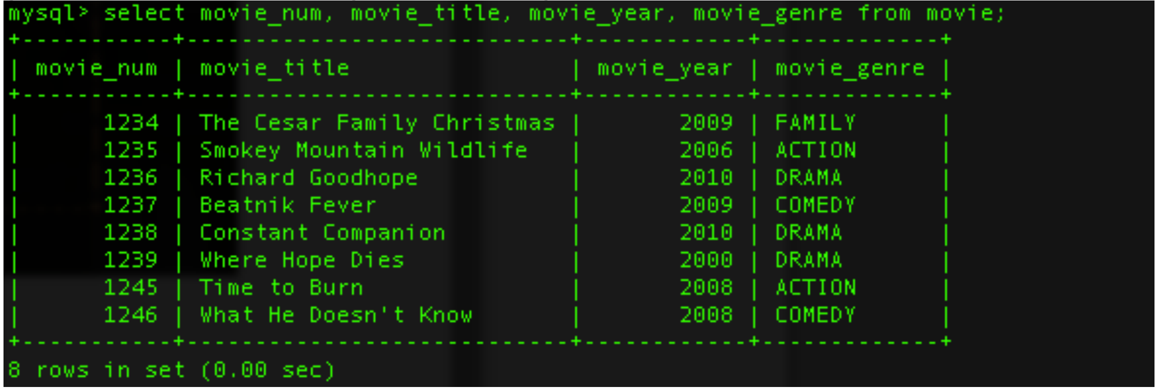
```
mysql> select movie_num, movie_title, moive_cost from movie where moive_cost > 40;
+-----------+--------------------------+------------+
| movie_num | movie_title              | moive_cost |
+-----------+--------------------------+------------+
|      1235 | Smokey Mountain Wildlife |      59.95 |
|      1236 | Richard Goodhope         |      59.95 |
|      1238 | Constant Companion       |      89.95 |
|      1245 | Time to Burn             |      45.49 |
|      1246 | What He Doesn't Know     |      58.29 |
+-----------+--------------------------+------------+
5 rows in set (0.00 sec)
```

11. Write a query to display the movie number, movie title, movie cost, and movie genre for all movies that are either action or comedy movies and that have a cost that is less than $50. Sort the results in ascending order by genre. (Result shown in Figure P4.11.)Figure P4.11 Action or comedy movies costing less than $50CS 325 MySQL Project #4 4



```
mysql> select movie_num, movie_title, moive_cost, movie_genre
    -> from movie
    -> where movie_genre = ('action' or 'comedy') and moive_cost < 50;
+-----------+-----------------------------+------------+-------------+
| movie_num | movie_title                 | moive_cost | movie_genre |
+-----------+-----------------------------+------------+-------------+
|      1234 | The Cesar Family Christmas  |      39.95 | FAMILY      |
|      1237 | Beatnik Fever               |      29.95 | COMEDY      |
|      1239 | Where Hope Dies             |      25.49 | DRAMA       |
|      1245 | Time to Burn                |      45.49 | ACTION      |
+-----------+-----------------------------+------------+-------------+
4 rows in set, 10 warnings (0.03 sec)
```

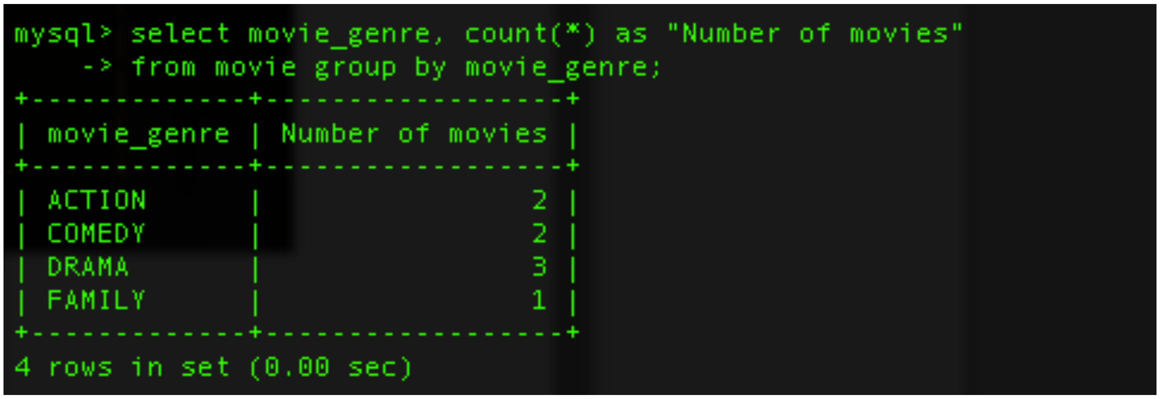
12. Write a query to display the movie number, movie title, movie year, and movie genre (result shown in Figure P4.12).Figure P4.12



```
mysql> select movie_num, movie_title, movie_year, movie_genre from movie;
+-----------+----------------------------+------------+-------------+
| movie_num | movie_title                | movie_year | movie_genre |
+-----------+----------------------------+------------+-------------+
|      1234 | The Cesar Family Christmas |       2009 | FAMILY      |
|      1235 | Smokey Mountain Wildlife   |       2006 | ACTION      |
|      1236 | Richard Goodhope           |       2010 | DRAMA       |
|      1237 | Beatnik Fever              |       2009 | COMEDY      |
|      1238 | Constant Companion         |       2010 | DRAMA       |
|      1239 | Where Hope Dies            |       2000 | DRAMA       |
|      1245 | Time to Burn               |       2008 | ACTION      |
|      1246 | What He Doesn't Know       |       2008 | COMEDY      |
+-----------+----------------------------+------------+-------------+
8 rows in set (0.00 sec)
```

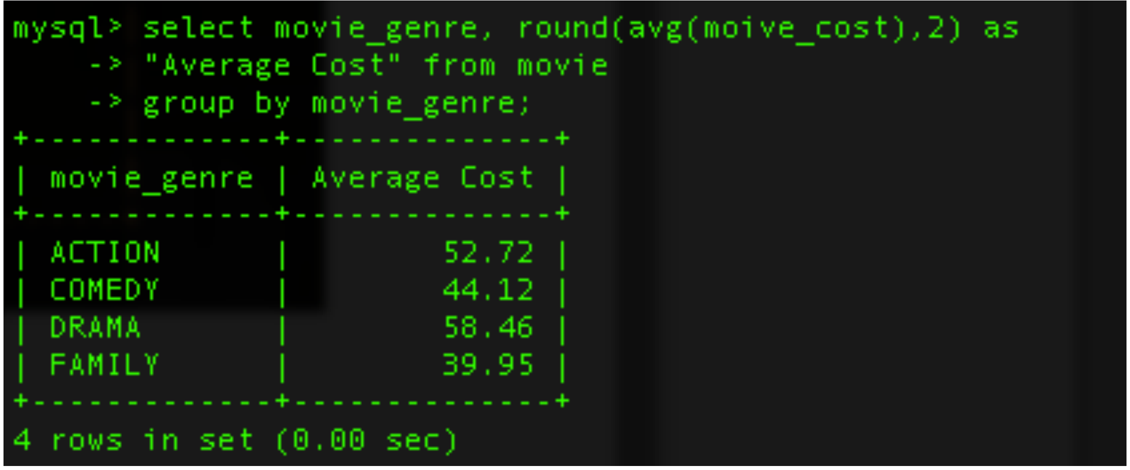
13. Write a query to display the movie genre and the number of movies in each genre (result shown in Figure P4.13).

Figure P4.13 Number of movies in genre



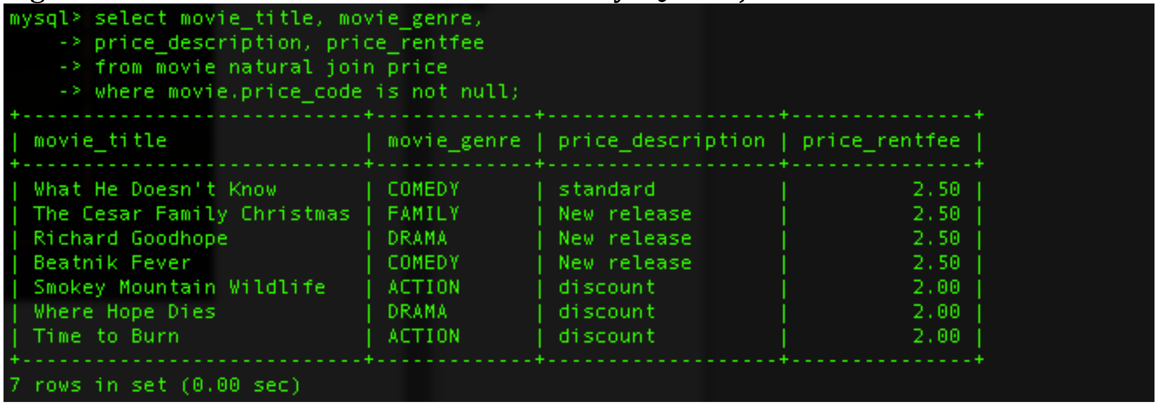
```
mysql> select movie_genre, count(*) as "Number of movies"
    -> from movie group by movie_genre;
+-------------+------------------+
| movie_genre | Number of movies |
+-------------+------------------+
| ACTION      |                2 |
| COMEDY      |                2 |
| DRAMA       |                3 |
| FAMILY      |                1 |
+-------------+------------------+
4 rows in set (0.00 sec)
```

14. Write a query to display the movie genre and average cost of movies in each genre (result shown in Figure P4.14). Figure P4.14 Average movie cost by genre



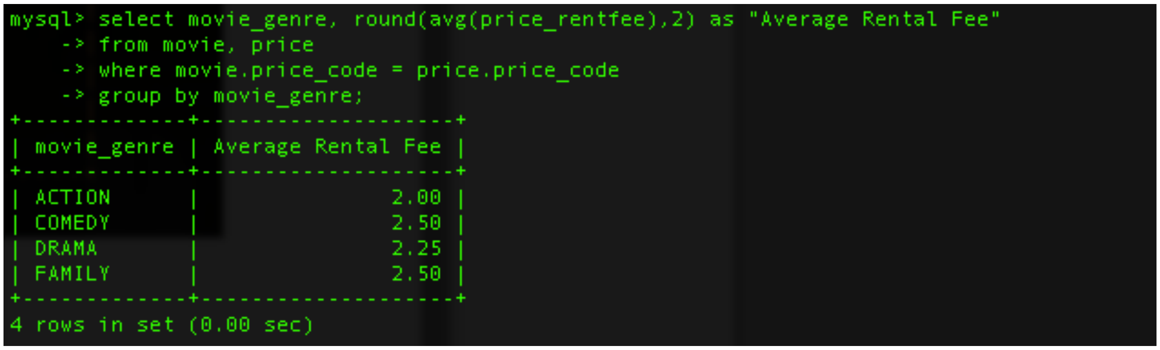
```
mysql> select movie_genre, round(avg(moive_cost),2) as
    -> "Average Cost" from movie
    -> group by movie_genre;
+-------------+--------------+
| movie_genre | Average Cost |
+-------------+--------------+
| ACTION      |        52.72 |
| COMEDY      |        44.12 |
| DRAMA       |        58.46 |
| FAMILY      |        39.95 |
+-------------+--------------+
4 rows in set (0.00 sec)
```

15. Write a query to display the movie title, movie genre, price description, and price rental fee for all movies with a price code (result shown in Figure P4.15). Figure P4.15 Rental fees for moviesCS 325 MySQL Project #4 5



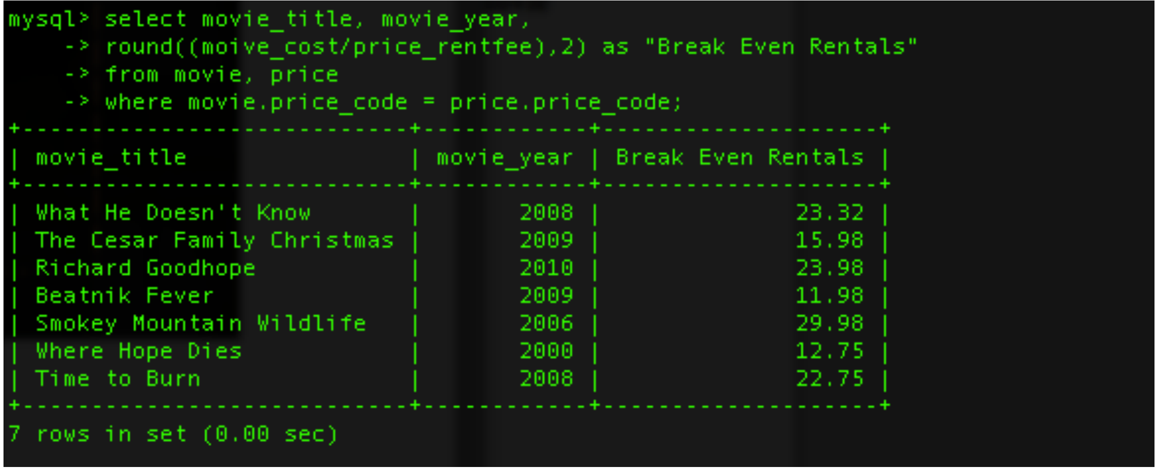
```
mysql> select movie_title, movie_genre,
    -> price_description, price_rentfee
    -> from movie natural join price
    -> where movie.price_code is not null;
+----------------------------+-------------+-------------------+---------------+
| movie_title                | movie_genre | price_description | price_rentfee |
+----------------------------+-------------+-------------------+---------------+
| What He Doesn't Know       | COMEDY      | standard          |          2.50 |
| The Cesar Family Christmas | FAMILY      | New release       |          2.50 |
| Richard Goodhope           | DRAMA       | New release       |          2.50 |
| Beatnik Fever              | COMEDY      | New release       |          2.50 |
| Smokey Mountain Wildlife   | ACTION      | discount          |          2.00 |
| Where Hope Dies            | DRAMA       | discount          |          2.00 |
| Time to Burn               | ACTION      | discount          |          2.00 |
+----------------------------+-------------+-------------------+---------------+
7 rows in set (0.00 sec)
```

16. Write a query to display the movie genre and average price rental fee for movies in each genre that have a price (result shown in Figure P4.16). Figure P4.16 Average rental fee by genre



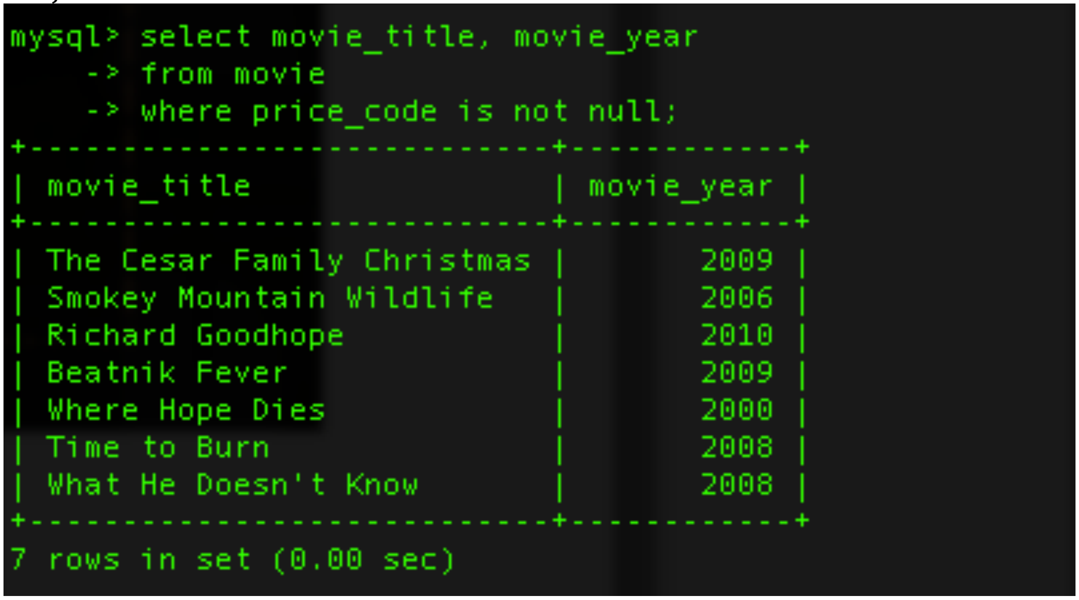
```
mysql> select movie_genre, round(avg(price_rentfee),2) as "Average Rental Fee"
    -> from movie, price
    -> where movie.price_code = price.price_code
    -> group by movie_genre;
+-------------+--------------------+
| movie_genre | Average Rental Fee |
+-------------+--------------------+
| ACTION      |               2.00 |
| COMEDY      |               2.50 |
| DRAMA       |               2.25 |
| FAMILY      |               2.50 |
+-------------+--------------------+
4 rows in set (0.00 sec)
```

17. Write a query to display the movie title, movie year, and the movie cost divided by the price rental fee for each movie that has a price to determine the number of rentals it will take to break even on the purchase of the movie (result shown in Figure P4.17). Figure P4.17 Breakeven rentals



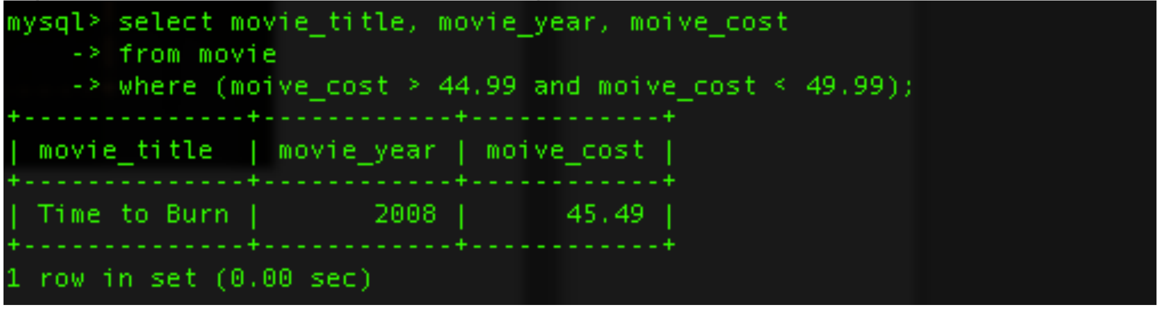
```
mysql> select movie_title, movie_year,
    -> round((moive_cost/price_rentfee),2) as "Break Even Rentals"
    -> from movie, price
    -> where movie.price_code = price.price_code;
+-----------------------------+------------+--------------------+
| movie_title                 | movie_year | Break Even Rentals |
+-----------------------------+------------+--------------------+
| What He Doesn't Know        |       2008 |              23.32 |
| The Cesar Family Christmas  |       2009 |              15.98 |
| Richard Goodhope            |       2010 |              23.98 |
| Beatnik Fever               |       2009 |              11.98 |
| Smokey Mountain Wildlife    |       2006 |              29.98 |
| Where Hope Dies             |       2000 |              12.75 |
| Time to Burn                |       2008 |              22.75 |
+-----------------------------+------------+--------------------+
7 rows in set (0.00 sec)
```

18. Write a query to display the movie title and movie year for all movies that have a price code (result shown in Figure P4.18). P4.18 Movies with a priceCS 325 MySQL Project #4 6



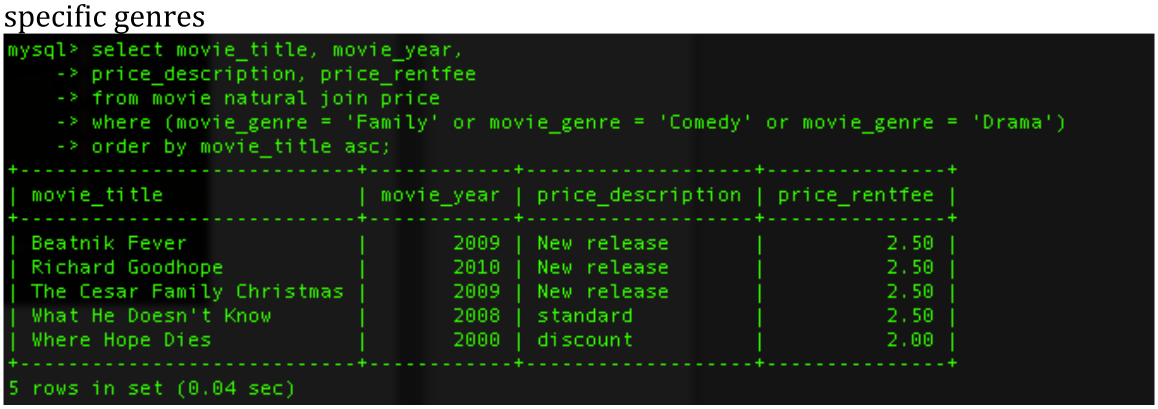
```
mysql> select movie_title, movie_year
    -> from movie
    -> where price_code is not null;
+-----------------------------+------------+
| movie_title                 | movie_year |
+-----------------------------+------------+
| The Cesar Family Christmas  |       2009 |
| Smokey Mountain Wildlife    |       2006 |
| Richard Goodhope            |       2010 |
| Beatnik Fever               |       2009 |
| Where Hope Dies             |       2000 |
| Time to Burn                |       2008 |
| What He Doesn't Know        |       2008 |
+-----------------------------+------------+
7 rows in set (0.00 sec)
```

19. Write a query to display the movie title, movie year, and movie cost for all movies that have a cost between $44.99 and $49.99 (result shown in Figure P4.19). Figure P4.19 Movies costs within a range



```
mysql> select movie_title, movie_year, moive_cost
    -> from movie
    -> where (moive_cost > 44.99 and moive_cost < 49.99);
+--------------+------------+------------+
| movie_title  | movie_year | moive_cost |
+--------------+------------+------------+
| Time to Burn |       2008 |      45.49 |
+--------------+------------+------------+
1 row in set (0.00 sec)
```

20. Write a query to display the movie title (ascending order), movie year, price description, and price rental fee for all movies that are in the genres Family, Comedy, or Drama (result shown in Figure P4.20). Figure P4.20 Movies with specific genres



```
mysql> select movie_title, movie_year,
    -> price_description, price_rentfee
    -> from movie natural join price
    -> where (movie_genre = 'Family' or movie_genre = 'Comedy' or movie_genre = 'Drama')
    -> order by movie_title asc;
+----------------------------+------------+-------------------+---------------+
| movie_title                | movie_year | price_description | price_rentfee |
+----------------------------+------------+-------------------+---------------+
| Beatnik Fever              |       2009 | New release       |          2.50 |
| Richard Goodhope           |       2010 | New release       |          2.50 |
| The Cesar Family Christmas |       2009 | New release       |          2.50 |
| What He Doesn't Know       |       2008 | standard          |          2.50 |
| Where Hope Dies            |       2000 | discount          |          2.00 |
+----------------------------+------------+-------------------+---------------+
5 rows in set (0.04 sec)
```