## Movie rating system

ACME Entertainment have requested that a chart be added to the application displaying the top ten movies as rated by users. Users should be able to assign a rating to each movie. Ratings will range from one star to five stars. The individual counts will be used to calculate an average rating.

To facilitate this functionality an additional table will be required to store the user ratings. The table should store the number of times each movie has been assigned each rating value along with an overall average rating.

The following table definition could be used to store the data:

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Field Type | Default | Comment |
| IDrating | Int |  | ID number used to link to tblDataMovies via the IDmovie column |
| 1starCount | Int | 0 | Number of 1-star ratings |
| 2starCount | Int | 0 | Number of 2-star ratings |
| 3starCount | Int | 0 | Number of 3-star ratings |
| 4starCount | Int | 0 | Number of 4-star ratings |
| 5starCount | Int | 0 | Number of 5-star ratings |
| movieRating | Float |  | Calculated field the average rating is calculated using the following formula:  ((1starCount) + (2starCount\*2) + (3starCount\*3) + (4starCount\*4) + (5starCount\*5)) / (1starCount + 2starCount + 3starCount + 4starCount + 5starCount) |

Table : movie rating table definition

The following SQL command can be used to create the table:

CREATE TABLE tbldatamovierating (

`IDrating` INT NOT NULL,

`1starCount` INT NOT NULL DEFAULT '0',

`2starCount` INT NOT NULL DEFAULT '0',

`3starCount` INT NOT NULL DEFAULT '0',

`4starCount` INT NOT NULL DEFAULT '0',

`5starCount` INT NOT NULL DEFAULT '0',

`movieRating` FLOAT AS ((1starCount + (2starCount\*2)

+ (3starCount\*3) + (4starCount\*4) + (5starCount\*5))

/ (1starCount + 2starCount + 3starCount + 4starCount + 5starCount))

);

Figure : rating table SQL

An SQL query can be used to initialise the records in the ratings table setting all values to 0:

INSERT INTO tbldatamovierating

(IDrating, `1starCount`, `2starCount`, `3starCount`, `4starCount`, `5starCount`)

SELECT IDmovie, 0 AS `1starCount`, 0 AS `2starCount`, 0 AS `3starCount`,

0 AS `4starCount`, 0 AS `5starCount`

FROM `tbldatamovies`

Figure : SQL to initialise the ratings table

It is anticipated that users will assign ratings to movies by selecting the title value. The assigned rating will then be added to the count of the assigned rating value. For example, if a user assigns a five star rating the number of five star ratings will be incremented by one.

To facilitate this interaction a view should be created linking the movie title to the ratings table. This will allow the ratings table to be updated using the movie title as the control.

The following SQL can be used to create the view:

CREATE VIEW vwTitleRating AS

SELECT Title, 1starCount, 2starCount, 3starCount, 4starCount, 5starCount, movieRating

FROM `tbldatamovies`

LEFT OUTER JOIN tbldatamovierating ON IDmovie = IDrating;

Figure : rating update view SQL

The ratings can be updated using a simple query such as the following to assign a 5-star rating to the film “Clockwork Orange”:

UPDATE `vwtitlerating`

SET `5starCount` = `5starCount` + 1

WHERE Title = "Clockwork Orange"

Figure : example rating query

The updated table is presented below showing “Clockwork Orange” with a five-star rating based on one evaluation:

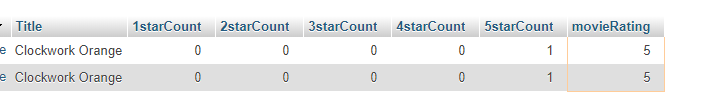


Figure : ratings applied to table

For testing an evaluation, the ratings table should be initialised with values. This can be completed by assigning random values to the ratings counts. The following SQL statement demonstrates the process setting a random value between 0 and 2000 for each rating count:

UPDATE `tbldatamovierating`

SET `1starCount`= FLOOR( RAND() \* 2000),

`2starCount`= FLOOR( RAND() \* 2000),

`3starCount`= FLOOR( RAND() \* 2000),

`4starCount`= FLOOR( RAND() \* 2000),

`5starCount`= FLOOR( RAND() \* 2000)

Figure : setting the ratings to random values

The following screen dump shows the records for “Clockwork Orange” after several runs of the random values script:

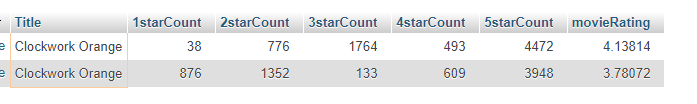


Figure : the ratings table with random values

A view should be created to select the top 10 rated movies for display in the chart. The view should select the Movie name and its rating selectin only the 10 highest rating movies. The following SQL can be used to create the view:

CREATE VIEW vwGetTop10Ratings AS (

SELECT Title, movieRating

FROM `vwtitlerating`

ORDER BY movieRating DESC

LIMIT 10);

Figure : creating the top 10 rated movie view

The records required to create the top 10 rating chart can now be retrieved through the view.

A snapshot of the output from the view is included below:

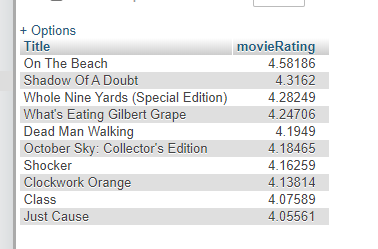


Figure : top 10 rated movie view