Gabriel Maldonado

SQL Bridge Week 2

Assignment – SQL One to Many Relationships

Solutions can also be found on my Github.

Query contains tables, keys and constraints:

https://github.com/maldonag/KVGMV9AW8U/blob/master/CUNY-SPS/SQL%20Bridge/Week2/howtovids.sql﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿

Query contains OneToMany, to display information from both tables:

https://github.com/maldonag/KVGMV9AW8U/blob/master/CUNY-SPS/SQL%20Bridge/Week2/OnetoMany.sql

1. In this assignment, we’ll practice working with one-to-many relationships in SQL. Suppose you are tasked with keeping track of a database that contain the best “how-to” videos on MySQL. You may want to first create a new database (schema) for this assignment. Videos table. Create one table to keep track of the videos. This table should include a unique ID, the title of the video, the length in minutes, and the URL. Populate the table with at least three related videos from YouTube or other publicly available resources.

CREATE TABLE `Videos` (

`Video\_Id` int(255) NOT NULL,

`Video\_Title` varchar(255) NOT NULL,

`Video\_Description` varchar(255) NOT NULL,

`Video\_URL` varchar(255) NOT NULL,

`Video\_Length` datetime NOT NULL

);

ALTER TABLE `Videos`

ADD PRIMARY KEY (`Video\_Id`);

ALTER TABLE `Videos`

MODIFY `Video\_Id` int(255) NOT NULL AUTO\_INCREMENT;

INSERT INTO `Videos` (`Video\_Title`, `Video\_Description`, `Video\_URL`, `Video\_Length`) VALUES

('Find Eigenvalues and Eigenvectors of a 2x2 Matrix', 'We work through two methods of finding the characteristic equation for λ, then use this to find two eigenvalues. We use Ax=λx to calculate two eigenvectors, then look at a visualisation.', 'http://www.youtube.com/watch?v=tXlMbAxbUI4', '2016-02-23 18:36:00'),

('How to Factor any Quadratic Equation', 'How to factor anything with x squared in it.', 'http://www.youtube.com/watch?v=ZQ-NRsWhOGI', '2011-09-12 09:37:00'),

('How to calculate the singular values of a matrix', 'In this video you will learn how to calculate the singular values of a matrix by finding the eigenvalues of A transpose A. We will also do a worked example to show you how it is done.', 'http://www.youtube.com/watch?v=4g-zS32oKEw', '2018-07-14 03:20:00'),

('Make Your Own Raycaster Game', 'I\'m happy to share my raycaster video with you! I hope you learn something, or find it interesting and stay tuned for more fun videos! ', 'https://www.youtube.com/watch?v=gYRrGTC7GtA&t=201s', '2020-03-13 16:51:00'),

('FIRST PERSON MOVEMENT in Unity - FPS Controller', 'Let\'s see how to get an FPS Character Controller up and running in no time!\r\n\r\n', 'https://www.youtube.com/watch?v=\_QajrabyTJc', '2019-10-27 23:52:00');

1. Create and populate Reviewers table. Create a second table that provides at least two user reviews for each of at least two of the videos. These should be imaginary reviews that include columns for the user’s name (“Asher”, “Cyd”, etc.), the rating (which could be NULL, or a number between 0 and 5), and a short text review (“Loved it!”). There should be a column that links back to the ID column in the table of videos.

CREATE TABLE `Reviews` (

`Review\_Id` int(255) NOT NULL,

`Review\_UserName` varchar(64) NOT NULL,

`Review\_Rating` enum('0','1','2','3','4','5') DEFAULT NULL,

`Review\_ShortText` enum('Loved it!','Liked it.','It''s okay.','Don''t like it.','Hated it!') NOT NULL,

`Reviews\_Video\_Id` int(255) NOT NULL

);

ALTER TABLE `Reviews`

ADD PRIMARY KEY (`Review\_Id`),

ADD KEY `Reviews\_fk\_Videos` (`Reviews\_Video\_Id`);

ALTER TABLE `Reviews`

MODIFY `Review\_Id` int(255) NOT NULL AUTO\_INCREMENT;

ALTER TABLE `Reviews`

ADD CONSTRAINT `Reviews\_fk\_Videos` FOREIGN KEY (`Reviews\_Video\_Id`) REFERENCES `Videos` (`Video\_Id`) ON DELETE CASCADE ON UPDATE CASCADE;

INSERT INTO `Reviews` (`Review\_UserName`, `Review\_Rating`, `Review\_ShortText`, `Reviews\_Video\_Id`) VALUES

('Avocado Pirate', '5', 'Loved it!', 1),

('John Richards', '5', 'Loved it!', 1),

('Zhang Dave', NULL, 'Loved it!', 2),

('Velez Darlene', '3', 'It\'s okay.', 3),

('Suzie Mahase', '5', 'Loved it!', 3),

('Levin Michael', NULL, 'It\'s okay.', 3),

('Alexis Santos', '4', 'Liked it.', 3),

('Dave Seliger', '1', 'Don\'t like it.', 2),

('Robles Jessica', '2', 'Hated it!', 2),

('Rodriquez Bianca', '3', 'It\'s okay.', 2),

('John Wick', '0', 'Hated it!', 2),

('Matt Booty', '3', 'It\'s okay.', 4),

('Tim Cook', '2', 'Hated it!', 5),

('Jane Doe', '5', 'Loved it!', 4),

('Bill Gates', '5', 'Loved it!', 2),

('John Doe', '3', 'It\'s okay.', 5);

1. Report on Video Reviews. Write a JOIN statement that shows information from both tables. Assignment – SQL One to Many Relationships.

SELECT videos.Video\_Id, videos.Video\_Title as 'Video Title', videos.Video\_Description as Description, videos.Video\_URL AS URL, TIME\_FORMAT(video\_Length, "%H:%I") AS 'Video Length',

reviews.Review\_Id, reviews.Review\_UserName AS Reviewer, CONCAT(reviews.Review\_Rating, " Stars") AS Rating, reviews.Review\_ShortText AS Review, reviews.Reviews\_Video\_Id as VideoFK

FROM howtovids.videos

RIGHT JOIN howtovids.reviews -- Right join used to display all reviews associated with videos on the Left table

ON videos.video\_Id = reviews.Review\_Id;