

Name: Katie Salas

Project: SQL – One to Many Relationships

- 1.) 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

```
1 CREATE table videos(  
2 unique_id integer primary key NOT NULL,  
3 title VARCHAR(255) NOT NULL,  
4 length VARCHAR(255) NOT NULL,  
5 url VARCHAR(255) NOT NULL  
6 );
```

```
1 insert into videos  
2 values (001, 'The Scientific Way to Cut a Cake - Numberphile', 5.08, 'https://www.youtube.com/watch?v=wBU9N3  
3 insert into videos  
4 values(002, 'The Remarkable Way We Eat Pizza - Numberphile', 7.41, 'https://www.youtube.com/watch?v=gi-TBlh4  
5 insert into videos  
6 values (003, 'Ham Sandwich Problem - Numberphile', 5.53, 'https://www.youtube.com/watch?v=YCXmUi56rao');  
7 |
```

```
1 select * from videos;
```

Data Output Explain Messages Notifications					
	unique_id [PK] integer	title character varying (255)	length character varying (255)	url character varying (255)	
1	1	The Scientific Way to Cut a Ca...	5.08	https://www.youtube.com/wa...	
2	2	The Remarkable Way We Eat ...	7.41	https://www.youtube.com/wa...	
3	3	Ham Sandwich Problem - Nu...	5.53	https://www.youtube.com/wa...	

- 2.) 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", "John", 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.

```
1 create table reviewers(  
2 unique_id integer primary key not null,  
3 username varchar (255) not null,  
4 rating integer not null,  
5 view varchar (255) not null);
```

Data Output Explain Messages Notifications

CREATE TABLE

Query returned successfully in 65 msec.

```
1 insert into reviewers  
2 values (1, 'Lei Rose', 5, 'bold of you to assume I have leftover cake');  
3 insert into reviewers  
4 values (2, 'John Chasant', 4.5, 'Props to Germany for putting Gauss on their money.');
```





Data Output Explain Messages Notifications

INSERT 0 1

Query returned successfully in 95 msec.

```
1 select * from reviewers;  
2
```

Data Output Explain Messages Notifications

	 unique_id [PK] integer	 username character varying (255)	 rating integer	 view character varying (255)
1	1	Lei Rose	5	bold of you to assume I have leftover cake
2	2	John Chasant	4.5	Props to Germany for putting Gauss on their money.
3	3	DHGameStudios	4	I think the only problem here is that theres only ham in that sandwich.

3.) Report on Video Reviews. Write a JOIN statement that shows information from both tables.

```
2 select * from videos
3 full outer join reviewers
4 on videos.unique_id = reviewers.unique_id
```

Data Output Explain Messages Notifications

	unique_id integer	title character varying (255)	length character	url character varying (255)	unique_id integer	username character varying (255)	rating integer	view character varying (255)
1	1	The Scientific Way to Cut a Ca...	5.08	https://www.youtube.com/wa...	1	Lei Rose	5	bold of you to assume I have leftover cake
2	2	The Remarkable Way We Eat ...	7.41	https://www.youtube.com/wa...	2	John Chasant	5	Props to Germany for putting Gauss on their money.
3	3	Ham Sandwich Problem - Nu...	5.53	https://www.youtube.com/wa...	3	DHGameStudios	4	I think the only problem here is that theres only ham in that sandwich