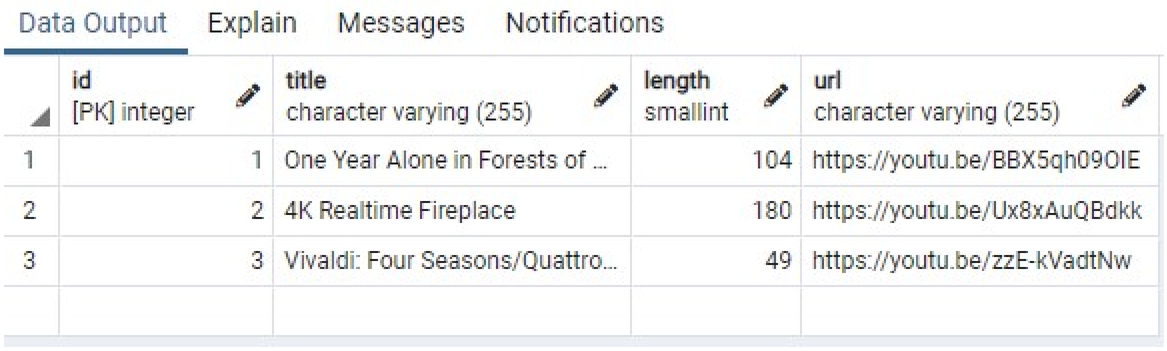
Name: Kenneth Bentley

Data Analytics 2

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

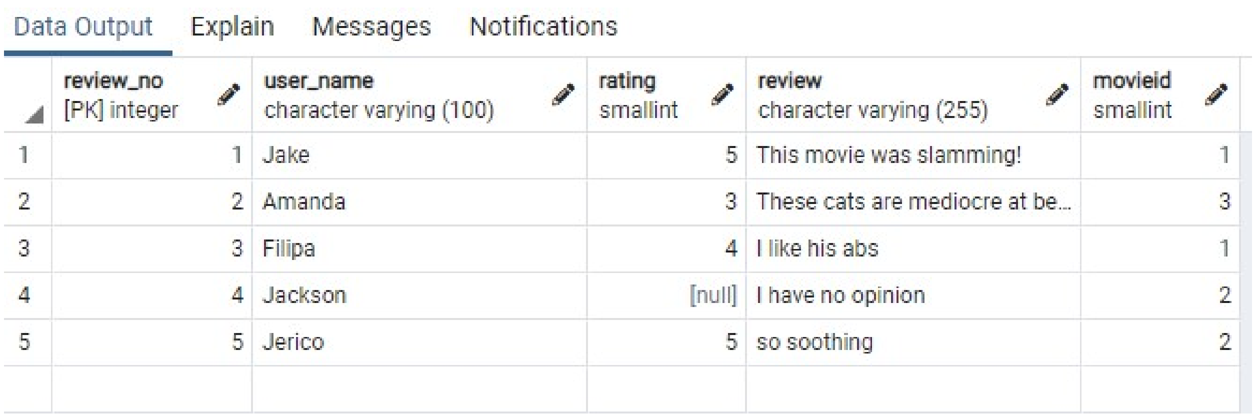
```
DROP TABLE IF EXISTS videos;

CREATE TABLE videos (

        ID SERIAL PRIMARY KEY,

        title VARCHAR(255),

        length SMALLINT,

        url VARCHAR(255));


INSERT INTO videos

        (title, length, url)

VALUES

        ('One Year Alone in Forests of Sweden', 104, 'https://youtu.be/BBX5qh09OIE'),

        ('4K Realtime Fireplace', 180, 'https://youtu.be/Ux8xAuQBdkk'),

        ('Vivaldi: Four Seasons/Quattro Stagioni', 49, 'https://youtu.be/zzE-kVadtNw');


SELECT * FROM videos;
```



	id [PK] integer	title character varying (255)	length smallint	url character varying (255)
1	1	One Year Alone in Forests of ...	104	https://youtu.be/BBX5qh09OIE
2	2	4K Realtime Fireplace	180	https://youtu.be/Ux8xAuQBdkk
3	3	Vivaldi: Four Seasons/Quattro...	49	https://youtu.be/zzE-kVadtNw

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.

```
DROP TABLE IF EXISTS reviews;

CREATE TABLE reviews (

        review_no SERIAL PRIMARY KEY,

        user_name VARCHAR(100) NOT NULL,

        rating SMALLINT,

        review VARCHAR(255),

        movieid SMALLINT);


INSERT INTO reviews

        (user_name, rating, review, movieid)

VALUES

        ('Jake', 5, 'This movie was slamming!', 1),

        ('Amanda', 3, 'These cats are mediocre at best', 3),

        ('Filipa', 4, 'I like his abs', 1),

        ('Jackson', null, 'I have no opinion', 2),

        ('Jerico', 5, 'so soothing', 2);


SELECT * FROM reviews;
```



Data Output | Explain | Messages | Notifications

	review_no [PK] integer	user_name character varying (100)	rating smallint	review character varying (255)	movieid smallint
1	1	Jake	5	This movie was slamming!	1
2	2	Amanda	3	These cats are mediocre at be...	3
3	3	Filipa	4	I like his abs	1
4	4	Jackson	[null]	I have no opinion	2
5	5	Jerico	5	so soothing	2

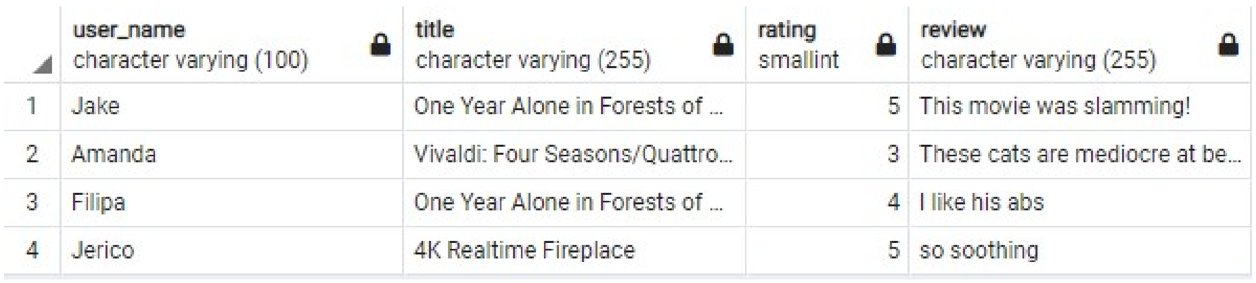
3.) Report on Video Reviews. Write a JOIN statement that shows information from both tables.

SELECT reviews.user_name, videos.title, reviews.rating, reviews.review

FROM reviews INNER JOIN videos

ON reviews.movieid=videos.id

WHERE reviews.rating IS NOT NULL;



	user_name character varying (100)	title character varying (255)	rating smallint	review character varying (255)
1	Jake	One Year Alone in Forests of ...	5	This movie was slamming!
2	Amanda	Vivaldi: Four Seasons/Quattro...	3	These cats are mediocre at be...
3	Filipa	One Year Alone in Forests of ...	4	I like his abs
4	Jerico	4K Realtime Fireplace	5	so soothing