SQL-FinalProject

Alberta “Albi” Kovatcheva

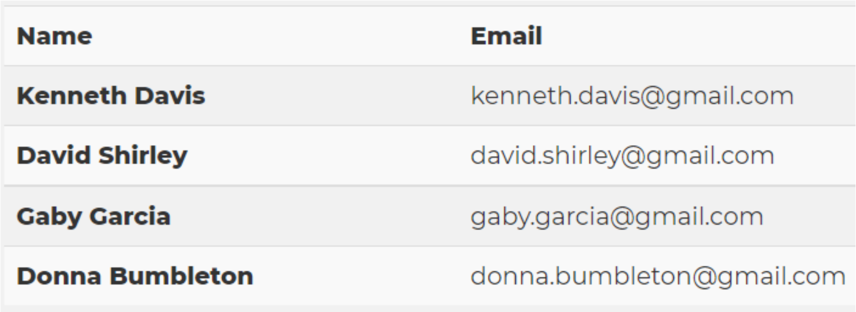
Part 1

1. Run a query that creates a table named viewer that has the following columns: viewer_id, first_name, last_name, email. Make sure the viewer_id is the primary key and auto increments.

Query:

```
CREATE TABLE viewer (
viewer_id INTEGER PRIMARY KEY AUTO_INCREMENT,
first_name VARCHAR(45) NOT NULL,
last_name VARCHAR(45) NOT NULL,
email VARCHAR(50) NOT NULL
);
```

2. Add the following customers:



Name	Email
Kenneth Davis	kenneth.davis@gmail.com
David Shirley	david.shirley@gmail.com
Gaby Garcia	gaby.garcia@gmail.com
Donna Bumbleton	donna.bumbleton@gmail.com

Query:

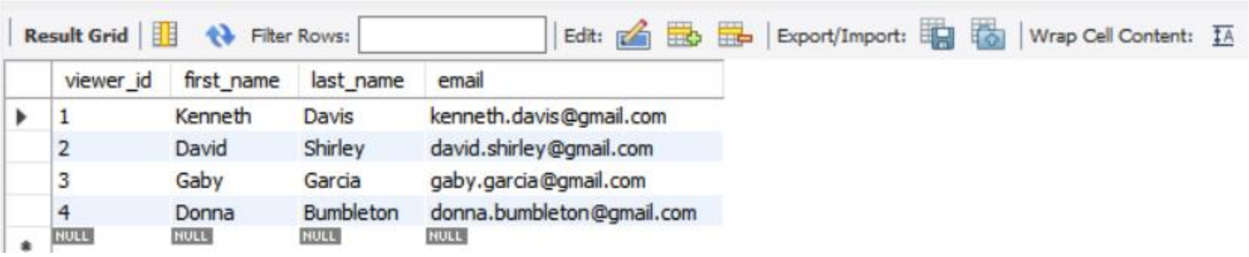
```
INSERT INTO viewer (first_name, last_name, email)
VALUES
("Kenneth", "Davis", "kenneth.davis@gmail.com"),
("David", "Shirley", "david.shirley@gmail.com"),
("Gaby", "Garcia", "gaby.garcia@gmail.com"),
("Donna", "Bumbleton", "donna.bumbleton@gmail.com");
```

3. Run a query to see all the new customers within the database.

Query:

```
SELECT * FROM viewer;
```

Results (4 rows returned):



viewer_id	first_name	last_name	email
1	Kenneth	Davis	kenneth.davis@gmail.com
2	David	Shirley	david.shirley@gmail.com
3	Gaby	Garcia	gaby.garcia@gmail.com
4	Donna	Bumbleton	donna.bumbleton@gmail.com
NULL	NULL	NULL	NULL

SQL-FinalProject

Alberta “Albi” Kovatcheva

Part 2

1. Run the following SQL query to add a new table into the database:

```
CREATE TABLE sakila.StreamingServiceQueue(
   queue_id INTEGER PRIMARY KEY AUTO_INCREMENT,
   genre NVARCHAR(50) NOT NULL,
   filmTitle NVARCHAR(50) NOT NULL,
   streamAvailable BOOLEAN CHECK(streamAvailable IN (true, false))
);
```

Task complete.

2. Next, run the following insert statements to add some data to the Products table:

```
-- query 1
INSERT INTO sakila.StreamingServiceQueue (genre, filmTitle, streamAvailable)
VALUES ("Movies based on books","The Breadwinner", true);

-- query 2
INSERT INTO sakila.StreamingServiceQueue (genre, filmTitle, streamAvailable)
VALUES ("Emotional","Roma", false);

-- query 3
INSERT INTO sakila.StreamingServiceQueue (genre, filmTitle, streamAvailable)
VALUES ("Campy","To Wong Foo, Thanks for Everything Julie Newmar", true);

-- query 4
INSERT INTO sakila.StreamingServiceQueue (genre, filmTitle, streamAvailable)
VALUES ("Critically Acclaimed","A Single Man", true);
```

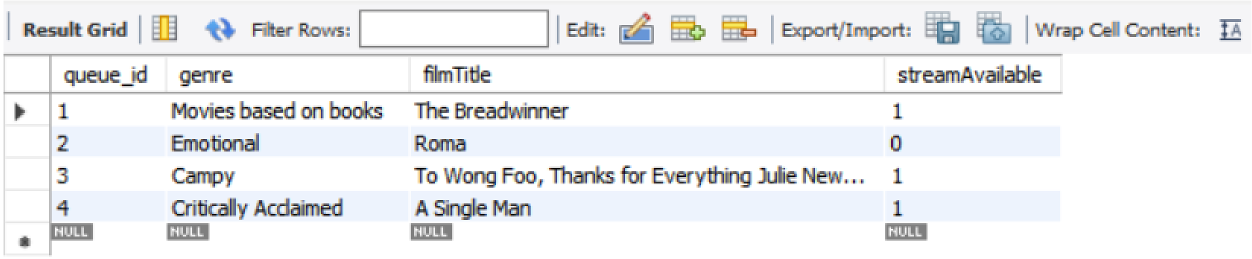
Task complete.

3. Run a query to see all of the films in your Streaming Service queue.

Query:

```
SELECT * FROM sakila.StreamingServiceQueue;
```

Results (4 rows returned):



	queue_id	genre	filmTitle	streamAvailable
▶	1	Movies based on books	The Breadwinner	1
	2	Emotional	Roma	0
	3	Campy	To Wong Foo, Thanks for Everything Julie New...	1
	4	Critically Acclaimed	A Single Man	1
*	NULL	NULL	NULL	NULL

SQL-FinalProject

Alberta “Albi” Kovatcheva

Part 3

1. Run the following SQL query to add a new table into the database:

```
CREATE TABLE sakila.WatchedList(
    watch_id INTEGER PRIMARY KEY AUTO_INCREMENT,
    queue_id INTEGER NOT NULL,
    viewer_id INTEGER NOT NULL,
    FOREIGN KEY(viewer_id) REFERENCES sakila.viewer(viewer_id),
    FOREIGN KEY(queue_id) REFERENCES sakila.StreamingServiceQueue(queue_id)
);
```

Task complete.

2. Next, run the following insert statements to add some data to the WatchedList table:

```
-- query 1
INSERT INTO sakila.WatchedList (viewer_id, queue_id)
VALUES (1, 2);

-- query 2
INSERT INTO sakila.WatchedList (viewer_id, queue_id)
VALUES (2, 1);

-- query 3
INSERT INTO sakila.WatchedList (viewer_id, queue_id)
VALUES (3, 2);

-- query 4
INSERT INTO sakila.WatchedList (viewer_id, queue_id)
VALUES (4, 1);
```

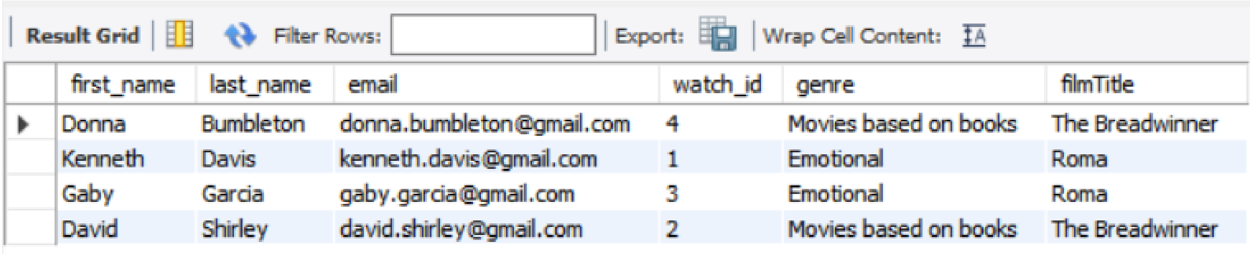
Task complete.

3. Lastly, run a query to see the customer's full name, their email address, and their watch_id as well as the genre and title of the film they watched. The list of customers should be ordered in alphabetical order by their last name.

   Query:

   ```
   SELECT first_name, last_name, email, watch_id, genre, filmTitle
   FROM sakila.viewer
   JOIN sakila.WatchedList
   USING(viewer_id)
   JOIN sakila.StreamingServiceQueue
   USING(queue_id)
   ORDER BY last_name;
   ```

   Results (4 rows returned):



first_name	last_name	email	watch_id	genre	filmTitle
Donna	Bumbleton	donna.bumbleton@gmail.com	4	Movies based on books	The Breadwinner
Kenneth	Davis	kenneth.davis@gmail.com	1	Emotional	Roma
Gaby	Garcia	gaby.garcia@gmail.com	3	Emotional	Roma
David	Shirley	david.shirley@gmail.com	2	Movies based on books	The Breadwinner