

SQL-HandsOn2

Alberta “Albi” Kovatcheva

Part 1

1. Write a query to find the first and last name, customer ID and rental ID for customers who have rented a film.

Query:

```
SELECT first_name, last_name, customer_id, rental_id  
FROM sakila.customer  
INNER JOIN sakila.rental  
USING (customer_id);
```

Results (1000 rows returned):

	first_name	last_name	customer_id	rental_id
►	MARY	SMITH	1	76
	MARY	SMITH	1	573
	MARY	SMITH	1	1185
	MARY	SMITH	1	1422
	MARY	SMITH	1	1476
	MARY	SMITH	1	1725
	MARY	SMITH	1	2308
	MARY	SMITH	1	2363
	MARY	SMITH	1	3284
	MARY	SMITH	1	4526
	MARY	SMITH	1	4611
	MARY	SMITH	1	5244

- Write a query that finds all films with actors that have an actor_id 5.

Query:

```
SELECT actor_id, first_name, last_name, film_id, title
FROM sakila.actor
INNER JOIN sakila.film_actor
USING(actor_id)
INNER JOIN sakila.film
USING(film_id)
WHERE actor_id = 5;
```

Results (29 rows returned):

actor_id	first_name	last_name	film_id	title
5	JOHNNY	LOLLOBRIGIDA	19	AMADEUS HOLY
5	JOHNNY	LOLLOBRIGIDA	54	BANGER PINOCCHIO
5	JOHNNY	LOLLOBRIGIDA	85	BONNIE HOLOCAUST
5	JOHNNY	LOLLOBRIGIDA	146	CHITTY LOCK
5	JOHNNY	LOLLOBRIGIDA	171	COMMANDMENTS EXPRESS
5	JOHNNY	LOLLOBRIGIDA	172	CONEHEADS SMOOCHY
5	JOHNNY	LOLLOBRIGIDA	202	DADDY PITTSBURGH
5	JOHNNY	LOLLOBRIGIDA	203	DAISY MENAGERIE
5	JOHNNY	LOLLOBRIGIDA	286	ENOUGH RAGING
5	JOHNNY	LOLLOBRIGIDA	288	ESCAPE METROPOLIS
5	JOHNNY	LOLLOBRIGIDA	316	FIRE WOLVES
5	JOHNNY	LOLLOBRIGIDA	340	FRONTIER CABIN

- Write a query that lists out all information of every film along with the name of the language for each film, even if a language does not exist for that film.

Query:

```
SELECT * FROM sakila.film
LEFT OUTER JOIN sakila.language
USING (language_id);
```

Results (1000 rows returned):



language_id	film_id	title	description	release_year	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	special_features
1	1	ACADEMY DINOSAUR	A Epic Drama of a Feminist And a Mad Scientist ...	2006	1	6	0.99	86	20.99	PG	Deleted Scenes,Behind the Scenes
1	2	ACE GOLDFINGER	A Astounding Epistle of a Database Administrat...	2006	3	4.99	48	12.99	G	Trailers,Deleted Scenes	
1	3	ADAPTATION HOLES	A Astounding Reflection of a Lumberjack And a ...	2006	7	2.99	50	18.99	NC-17	Trailers,Deleted Scenes	
1	4	AFFAIR PREJUDICE	A Painful Documentary of a Frisbee And a Lum...	2006	5	2.99	117	26.99	G	Commentaries,Behind the Scenes	
1	5	AFRICAN EGG	A Fast-Paced Documentary of a Pastry Chef An...	2006	6	2.99	130	22.99	G	Deleted Scenes	
1	6	AGENT TRUMAN	A Intrepid Panorama of a Robot And a Boy who...	2006	3	2.99	169	17.99	PG	Deleted Scenes	
1	7	AIRPLANE SIERRA	A Touching Saga of a Hunter And a Butler who ...	2006	6	4.99	62	28.99	PG-13	Trailers,Deleted Scenes	
1	8	AIRPORT POLLOCK	A Epic Tale of a Moose And a Girl who must Con...	2006	6	4.99	54	15.99	R	Trailers	
1	9	ALABAMA DEVIL	A Thoughtful Panorama of a Database Administr...	2006	3	2.99	114	21.99	PG-13	Trailers,Deleted Scenes	
1	10	ALADDIN CALENDAR	A Action-Packed Tale of a Man And a Lumberjac...	2006	6	4.99	63	24.99	NC-17	Trailers,Deleted Scenes	
1	11	ALAMO VIDEOTAPE	A Boring Epistle of a Butler And a Cat who must ...	2006	6	0.99	126	16.99	G	Commentaries,Behind the Scenes	

4. Write a query that lists out the title of films and the name of the actors who starred in those films. Additionally, only list films that starred artists whose first names start with a vowel.

Query:

```
SELECT title, first_name, last_name FROM sakila.film
JOIN sakila.film_actor
USING (film_id)
JOIN sakila.actor
USING (actor_id)
WHERE first_name LIKE "a%"
OR first_name LIKE "e%"
OR first_name LIKE "i%"
OR first_name LIKE "o"
OR first_name LIKE "u";
```

Results (577 rows returned):

Result Grid   Filter Rows: <input type="text"/>			
	title	first_name	last_name
▶	ALONE TRIP	ED	CHASE
	ARMY FLINTSTONES	ED	CHASE
	ARTIST COLDBLOODED	ED	CHASE
	BOONDOCK BALLROOM	ED	CHASE
	CADDYSHACK JEDI	ED	CHASE
	COWBOY DOOM	ED	CHASE
	EVE RESURRECTION	ED	CHASE
	FORREST SONS	ED	CHASE
	FRENCH HOLIDAY	ED	CHASE
	FROST HEAD	ED	CHASE
	HALLOWEEN NUTS	ED	CHASE
	HUNTER ALTER	ED	CHASE

Part 2

1. You have just been hired as a Data Analyst for a company that rents films to customers. They would like an inventory list of films that were rented for more than \$4.99.

Query:

```
SELECT film_id, title, rental_rate  
FROM sakila.film  
JOIN sakila.inventory  
USING(film_id)  
WHERE rental_rate > 4.99;
```

Results (0 rows returned):



The screenshot shows a database query results interface. At the top, there is a toolbar with a back arrow, a 'Result Grid' button, a 'Filter Rows' button with a dropdown menu, an 'Export' button, and a 'Wrap Cell Content' button. Below the toolbar is a table with three columns: 'film_id', 'title', and 'rental_rate'. The table is currently empty, indicating that no rows were returned by the query.

film_id	title	rental_rate
---------	-------	-------------