Refreshing SQL

INFO-H417: Lab Session 1

2021-2022

Find an SQL SELECT query to answer each of the following questions.

1. What are the names (first and last) and emails of our customers?

```
SELECT first_name, last_name, email FROM customer;
```

2. How many movies are rated PG-13?

```
SELECT count(*) FROM film WHERE rating = 'PG-13';
```

3. What is the first name of the staff member that did not store a picture in the database?

```
SELECT first_name FROM staff WHERE picture IS NULL;
```

4. Who are the customers (first and last name) who returned a DVD on May 27th 2005?

```
SELECT DISTINCT first_name, last_name
FROM customer
INNER JOIN rental USING(customer_id)
WHERE return_date::date = '2005-05-27';
```

5. Who is both an actor and a customer?

```
(SELECT first_name, last_name FROM customer)
INTERSECT
(SELECT first_name, last_name FROM actor);
```

6. Give all pairs of actors with the same last name, ordered by last name.

```
SELECT a.first_name as actor_1, b.first_name as actor_2, b.last_name as last_name
FROM actor a, actor b
WHERE a.actor_id < b.actor_id AND a.last_name = b.last_name
ORDER BY a.last_name;
```

7. What are the titles of the movies in which an actor with a last name ending in 'son' has played?

```
SELECT DISTINCT title
FROM film
INNER JOIN film_actor USING(film_id)
INNER JOIN actor USING(actor_id)
WHERE last_name LIKE ’%son’;
```

8. Who were the 10 first customers to rent a DVD?

```
SELECT first_name, last_name
FROM customer
INNER JOIN rental USING(customer_id)
GROUP BY first_name, last_name
ORDER BY min(rental_date)
LIMIT 10;
```

9. Who is the customer with the most DVD rentals?

```
SELECT c.first_name, c.last_name
FROM customer c
INNER JOIN rental r USING(customer_id)
GROUP BY c.customer_id
HAVING count(r.rental_id) > ALL (
    SELECT count(*)
    FROM rental r2
    WHERE r2.customer_id <> c.customer_id
    GROUP BY r2.customer_id
);
```

or

```
SELECT c.first_name, c.last_name
FROM customer c
INNER JOIN rental r USING(customer_id)
GROUP BY c.customer_id
ORDER BY count(r.rental_id) DESC
LIMIT 1;
```

10. Which customers rented at least one DVD lasting 3h or more?

```
SELECT first_name, last_name
```

```
FROM customer c
WHERE EXISTS (
    SELECT *
    FROM rental r
    INNER JOIN inventory USING(inventory_id)
    INNER JOIN film USING(film_id)
    WHERE c.customer_id = r.customer_id AND length >= 180
);

or

SELECT DISTINCT first_name, last_name
FROM customer
INNER JOIN rental USING(customer_id)
INNER JOIN inventory USING(inventory_id)
INNER JOIN film USING(film_id)
WHERE length >= 180;
```

11. Which customers did not rent a DVD lasting 3h or more?

```
SELECT first_name, last_name
FROM customer c
WHERE NOT EXISTS (
    SELECT *
    FROM rental r
    INNER JOIN inventory USING(inventory_id)
    INNER JOIN film USING(film_id)
    WHERE c.customer_id = r.customer_id AND length >= 180
);

or

SELECT first_name, last_name
FROM customer
WHERE customer_id NOT IN (
    SELECT DISTINCT customer_id
    FROM rental
    INNER JOIN inventory USING(inventory_id)
    INNER JOIN film USING(film_id)
    WHERE length >= 180
);
```

12. Who are the actors that have played in more movies than ’Harvey Hope’?

```
SELECT first_name, last_name
FROM actor
INNER JOIN film_actor USING(actor_id)
GROUP BY actor_id
HAVING count(film_id) > (
    SELECT count(*)
    FROM actor
    INNER JOIN film_actor USING(actor_id)
    WHERE first_name = ’Harvey’ AND last_name = ’Hope’
);
```

13. What are the titles of the movies that have never been rented yet?

```
SELECT title
FROM film f
WHERE NOT EXISTS (
    SELECT *
    FROM rental r
    INNER JOIN inventory i USING(inventory_id)
    WHERE f.film_id = i.film_id
);
```

14. What countries have at least one city with more than one customer?

```
WITH city_count(country, city, cnt) AS (
    SELECT country, city, count(customer_id)
    FROM customer
    INNER JOIN address USING(address_id)
    INNER JOIN city USING(city_id)
    INNER JOIN country USING(country_id)
    GROUP BY country, city
)
SELECT country
FROM city_count
GROUP BY country
```

```
HAVING max(cnt) > 1;

or

SELECT country
FROM country a
WHERE EXISTS (
    SELECT city
    FROM city b
    INNER JOIN address USING(city_id)
    INNER JOIN customer USING(address_id)
    WHERE a.country_id = b.country_id
    GROUP BY city
    HAVING count(customer_id) > 1
);
```

15. List the number of rentals per film, including films that have never been rented yet (count=zero).

```
SELECT title, count(rental_id)
FROM film
LEFT JOIN inventory USING(film_id)
LEFT JOIN rental USING(inventory_id)
GROUP BY title;
```

16. Who are the customers who returned 4 or more DVD's on the same day more than once?

```
WITH customer_return(customer_id, return_date, cnt) AS (
    SELECT customer_id, return_date::date, count(rental_id)
    FROM rental
    GROUP BY customer_id, return_date::date
    HAVING count(rental_id) >= 4
)
SELECT first_name, last_name
FROM customer_return
INNER JOIN customer USING(customer_id)
GROUP BY customer_id
HAVING count(return_date) > 1;
```

17. Who are the customers who have rented at least one movie from every category?

```
SELECT c.first_name, c.last_name
FROM customer c
WHERE NOT EXISTS (
    SELECT *
    FROM category ca
    WHERE NOT EXISTS (
        SELECT *
        FROM rental r
        INNER JOIN inventory i USING(inventory_id)
        INNER JOIN film f USING(film_id)
        INNER JOIN film_category fc USING(film_id)
        WHERE ca.category_id = fc.category_id
            AND r.customer_id = c.customer_id
    )
);
```