Refreshing SQL and Relational Algebra

INFO-H417: Lab Session 1

2023-2024

$$\boldsymbol{\sigma}\boldsymbol{\pi}\boldsymbol{\rho}\gamma \bowtie \cup \cap - ⟕ ⟖ ⟗ \leftarrow$$

Find an SQL SELECT query and an equivalent RA query to answer each of the following questions.[1]

Note: The list of symbols defined above are not sufficient to write relational algebra answers for all queries, as the ordering and limit operators are missing. In these cases, the RA answer corresponds to the SQL query without the ORDER BY and/or LIMIT operators.

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

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

   $\boldsymbol{\pi}_{first\_name,last\_name,email}(customer)$

2. How many movies are rated PG-13?

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

   $\gamma_{count(*)}(\boldsymbol{\sigma}_{\text{rating=PG-13}}(film))$

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

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

   $\boldsymbol{\pi}_{first\_name}\boldsymbol{\sigma}_{picture\text{ IS NULL}}(staff)$

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';
   ```

   $\gamma_{first\_name,last\_name}\boldsymbol{\sigma}_{return\_date='2005-05-27}$

   $\boldsymbol{\pi}_{return\_date=date(return\_date),first\_name,last\_name}(customer \bowtie_{customer\_id} rental)$

[1] https://www.postgresqltutorial.com/postgresql-getting-started/postgresql-sample-database/

5. Who is both an actor and a customer?

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

$\pi_{first\_name,last\_name}\ (customer) \cap \pi_{first\_name,last\_name}(actor)$

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;
```

$$\pi_{a.first\_name->actor\_1,b.first\_name\rightarrow actor\_2,b.last\_name}$$
$$\sigma_{a.actor\_id<b.actor\_id}(\rho_a(actor) \bowtie_{a.last\_name=b.last\_name}\ \rho_b(actor))$$

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’;
```

$\gamma_{title}\sigma_{last\_name\ \text{LIKE}\ \%son}(film \bowtie_{film\_id}\ film\_actor \bowtie_{actor\_id}\ actor)$

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;
```

$\gamma_{first\_name,last\_name}(customer \bowtie_{customer\_id}\ rental)$

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;
```

$\boldsymbol{\pi}_{first\_name,last\_name}\gamma_{customer\_id}(customer \bowtie_{customer\_id} rental)$

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;
```

$$\gamma_{first\_name,last\_name}\boldsymbol{\sigma}_{length>=180}(
\quad customer \bowtie_{customer\_id} rental \bowtie_{inventory\_id} inventory \bowtie_{film\_id} film)$$