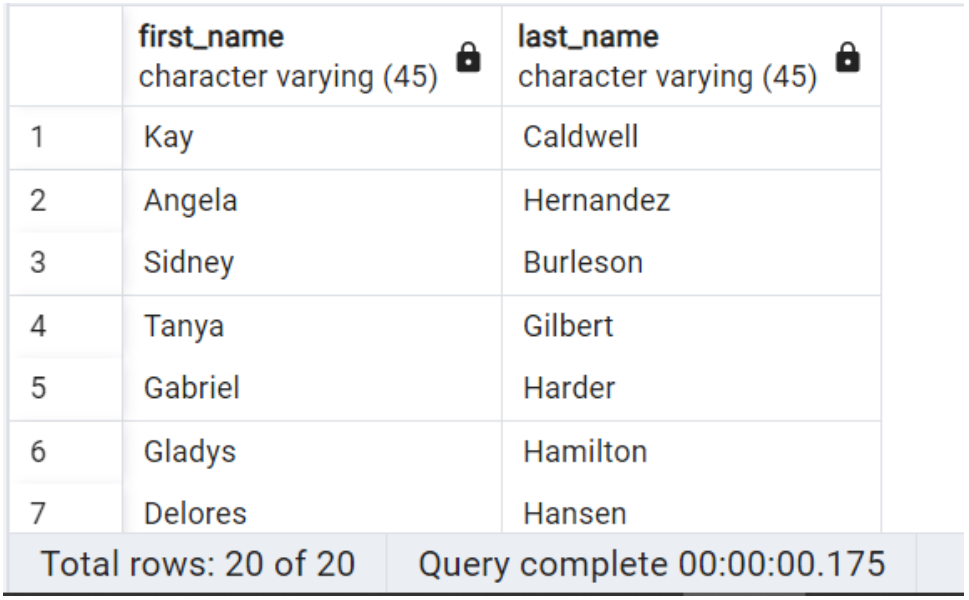
Java module 4

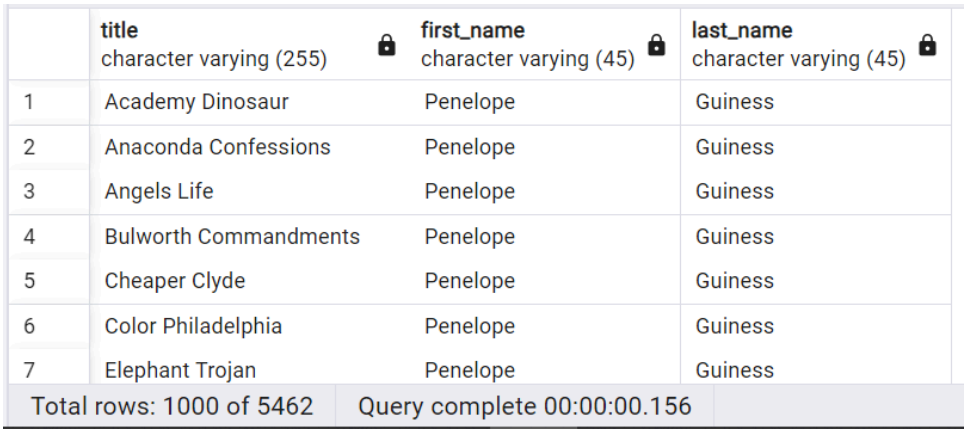
Exercises Day 3

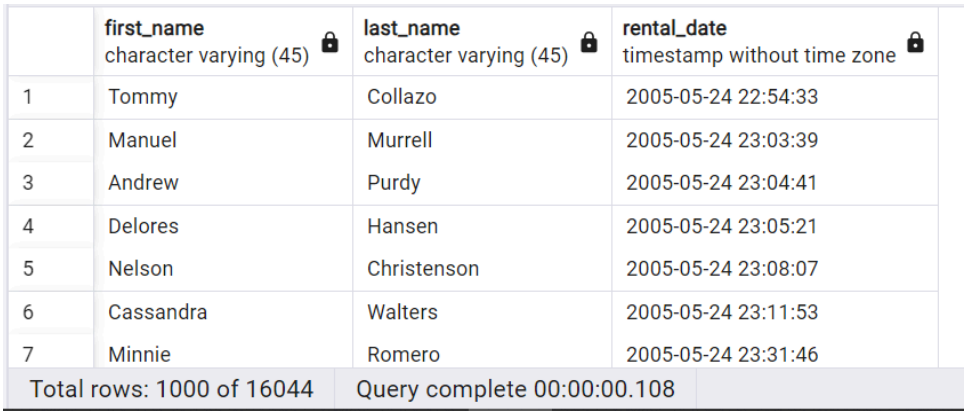
1 - Join	SQL Join
Instructions	List the first name and last name of all customers who have rented the film "Hunchback Impossible". You will need to join **customer**, **rental**, **inventory**, and **film** tables.
Expected output	
Solution	SELECT c.first_name, c.last_name FROM customer c JOIN rental r ON c.customer_id = r.customer_id JOIN inventory i ON r.inventory_id = i.inventory_id JOIN film f ON i.film_id = f.film_id WHERE f.title = 'Hunchback Impossible';

	first_name character varying (45)	last_name character varying (45)
1	Kay	Caldwell
2	Angela	Hernandez
3	Sidney	Burleson
4	Tanya	Gilbert
5	Gabriel	Harder
6	Gladys	Hamilton
7	Delores	Hansen

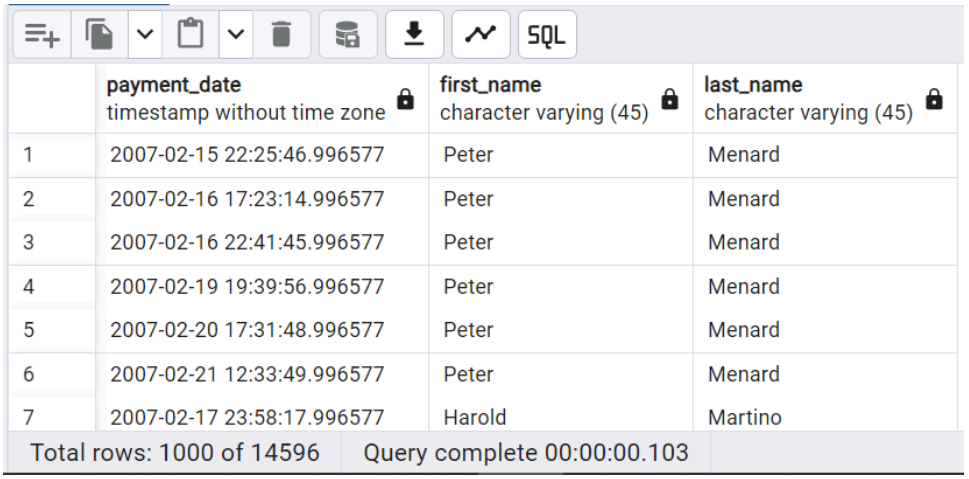
Total rows: 20 of 20 Query complete 00:00:00.175

2 - Join	SQL Join
Instructions	Find the titles of films and the names of the actors in them. Hint: Use INNER JOIN

Expected output	
Solution	-- Select film titles and actor names SELECT film.title, actor.first_name, actor.last_name FROM film -- Join film with film_actor to get the actor IDs for each film INNER JOIN film_actor ON film.film_id = film_actor.film_id -- Join actor to get the names of the actors from their IDs INNER JOIN actor ON film_actor.actor_id = actor.actor_id

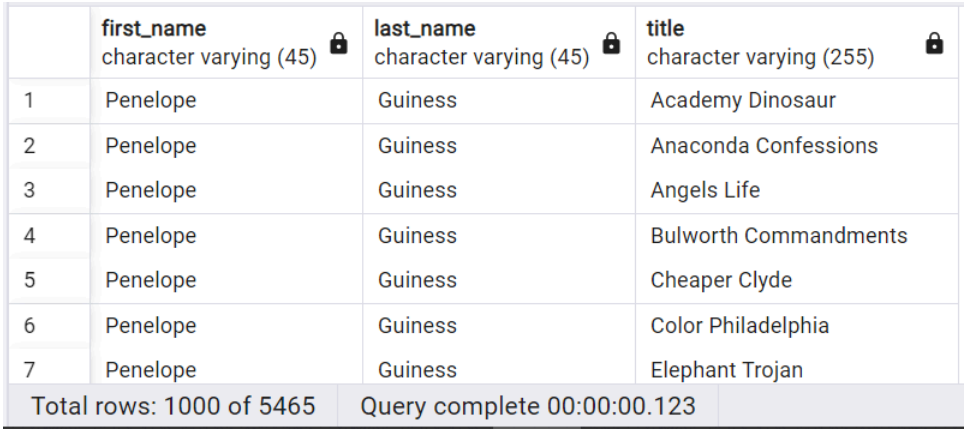
3 - Join	SQL Join
Instructions	List all customers and their rental information, including those who have never rented a movie. Hint: Use LEFT OUTER JOIN
Expected output	
Solution	-- Select customer names and their rental dates SELECT customer.first_name, customer.last_name, rental.rental_date FROM customer -- Left join to include all customers, even those without rentals LEFT JOIN rental ON customer.customer_id = rental.customer_id

4 - Join	SQL Join
Instructions	List all payments and include customer information even if they have never made a payment (unlikely in a real scenario but useful for demonstration). Hint: Use RIGHT OUTER JOIN

Expected output	
Solution	-- Select payment dates and customer names SELECT payment.payment_date, customer.first_name, customer.last_name FROM payment -- Right join to include all customers, even if they haven't made payments RIGHT JOIN customer ON payment.customer_id = customer.customer_id

	payment_date timestamp without time zone	first_name character varying (45)	last_name character varying (45)
1	2007-02-15 22:25:46.996577	Peter	Menard
2	2007-02-16 17:23:14.996577	Peter	Menard
3	2007-02-16 22:41:45.996577	Peter	Menard
4	2007-02-19 19:39:56.996577	Peter	Menard
5	2007-02-20 17:31:48.996577	Peter	Menard
6	2007-02-21 12:33:49.996577	Peter	Menard
7	2007-02-17 23:58:17.996577	Harold	Martino

Total rows: 1000 of 14596 Query complete 00:00:00.103

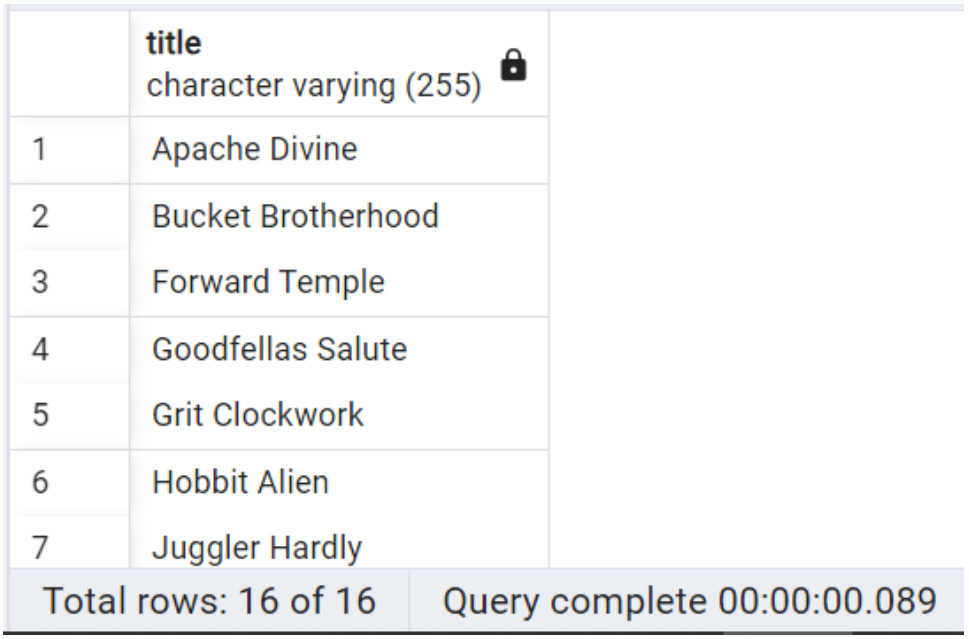
5 - Join	SQL Join
Instructions	List all actors and the films they've been in, including actors who've been in no films and films that have no actors listed. Hint: Use FULL OUTER JOIN
Expected output	
Solution	-- Select actor names and film titles SELECT actor.first_name, actor.last_name, film.title FROM actor -- Full outer join to include all actors and all films FULL OUTER JOIN film_actor ON actor.actor_id = film_actor.actor_id FULL OUTER JOIN film ON film_actor.film_id = film.film_id

	first_name character varying (45)	last_name character varying (45)	title character varying (255)
1	Penelope	Guiness	Academy Dinosaur
2	Penelope	Guiness	Anaconda Confessions
3	Penelope	Guiness	Angels Life
4	Penelope	Guiness	Bulworth Commandments
5	Penelope	Guiness	Cheaper Clyde
6	Penelope	Guiness	Color Philadelphia
7	Penelope	Guiness	Elephant Trojan

Total rows: 1000 of 5465 Query complete 00:00:00.123

6 - Join	SQL Join
Instructions	Find the titles of all films that have been rented out more than 30 times.

Expected output



	title character varying (255)
1	Apache Divine
2	Bucket Brotherhood
3	Forward Temple
4	Goodfellas Salute
5	Grit Clockwork
6	Hobbit Alien
7	Juggler Hardly

Total rows: 16 of 16 Query complete 00:00:00.089

Solution

```
SELECT title
FROM film
WHERE film_id IN
    ( SELECT film_id
    FROM inventory
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    GROUP BY film_id
    HAVING COUNT(rental.rental_id) > 30 );
```