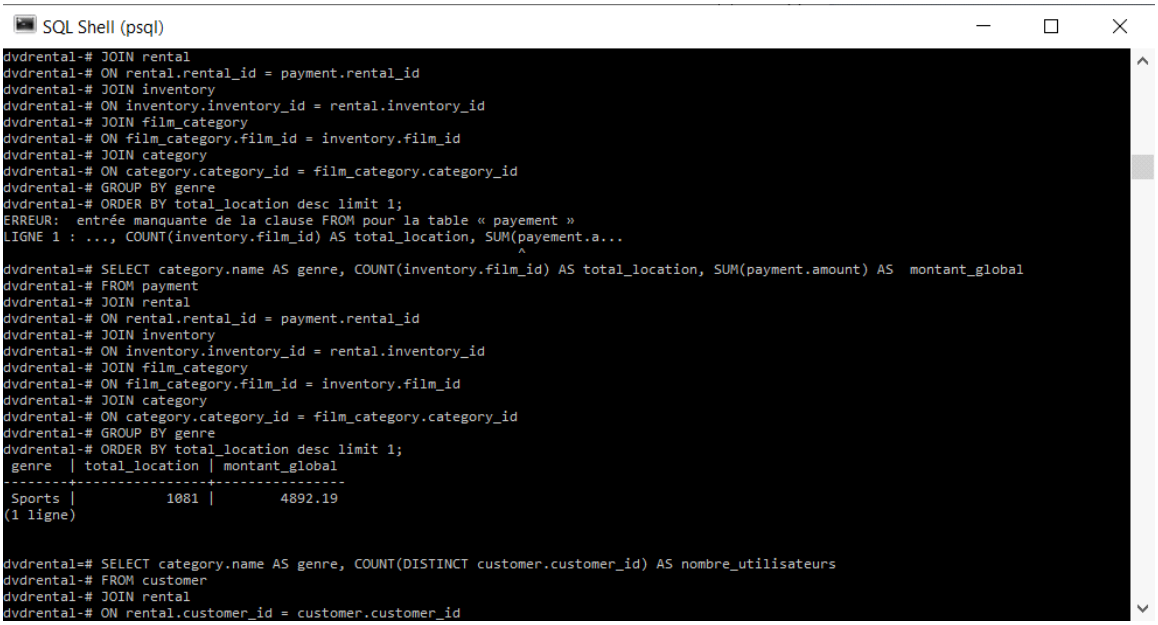
BASE DE DONNEE dvdrental:

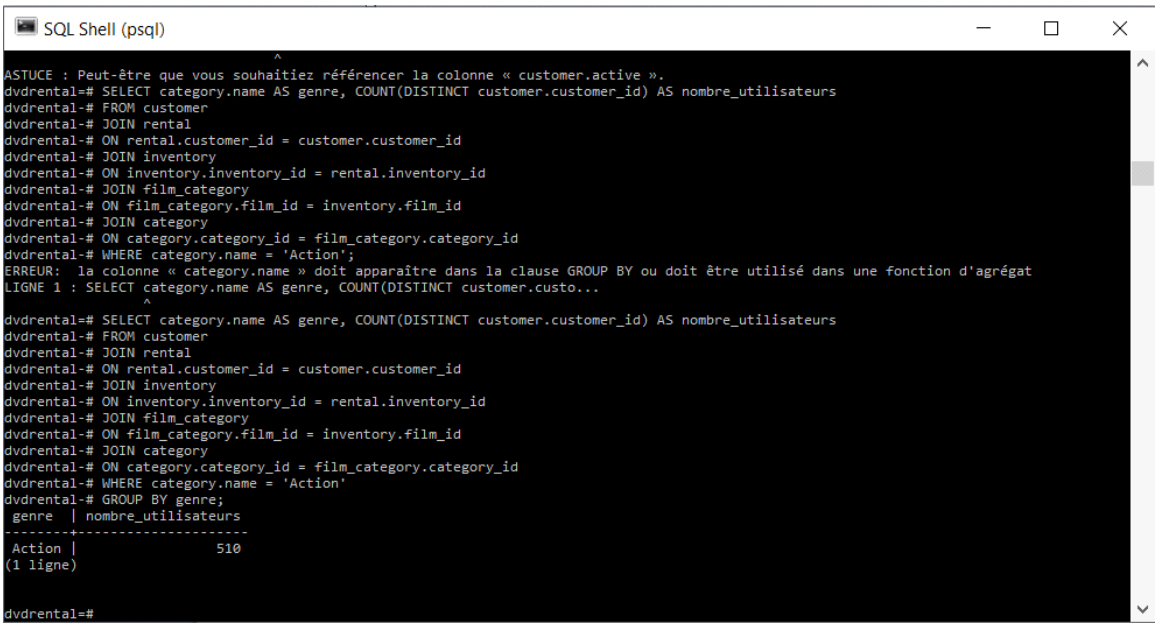
1. Quel est la catégorie de film la plus loué, quel est son chiffre d'affaire.

-> SELECT category.name AS genre, COUNT(inventory.film_id) AS total_location, SUM(payment.amount) AS montant_global FROM payment JOIN rental ON rental.rental_id = payment.rental_id JOIN inventory ON inventory.inventory_id = rental.inventory_id JOIN film_category ON film_category.film_id = inventory.film_id JOIN category ON category.category_id = film_category.category_id GROUP BY genre ORDER BY total_location desc limit 1



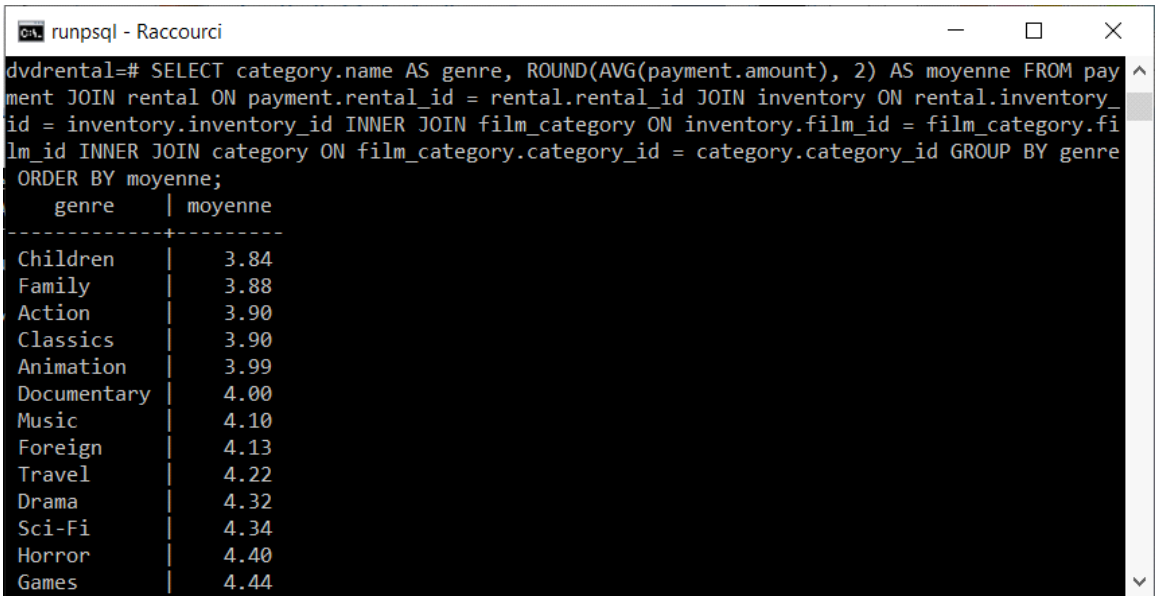
2. Combien d'utilisateurs "distincte" ont loué des films d'actions.

-> SELECT category.name AS genre, COUNT(DISTINCT customer.customer_id) AS nombre_utilisateurs FROM customer JOIN rental ON rental.customer_id = customer.customer_id JOIN inventory ON inventory.inventory_id = rental.inventory_id JOIN film_category ON film_category.film_id = inventory.film_id JOIN category ON category.category_id = film_category.category_id WHERE category.name = 'Action' GROUP BY genre;



3. Déterminer la moyenne de revenu par catégorie ordonnée dans l'ordre décroissant.

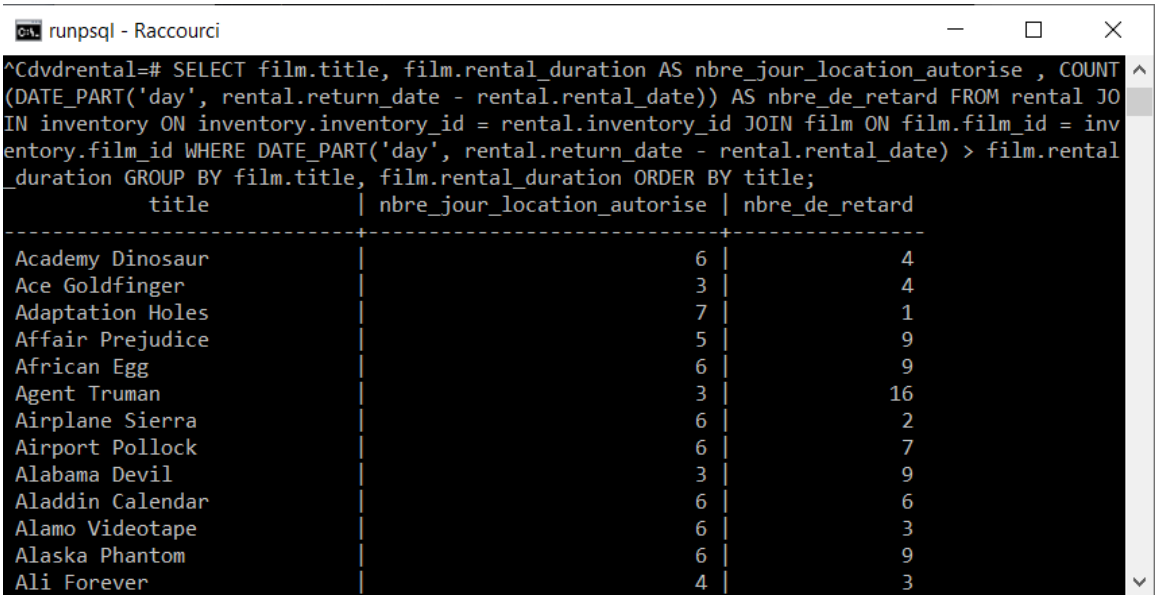
-> SELECT category.name AS genre, ROUND(AVG(payment.amount), 2) AS moyenne FROM payment JOIN rental ON payment.rental_id = rental.rental_id JOIN inventory ON rental.inventory_id = inventory.inventory_id INNER JOIN film_category ON inventory.film_id = film_category.film_id INNER JOIN category ON film_category.category_id = category.category_id GROUP BY genre ORDER BY moyenne;



4. Quels sont les films qui sont retournés en retard.

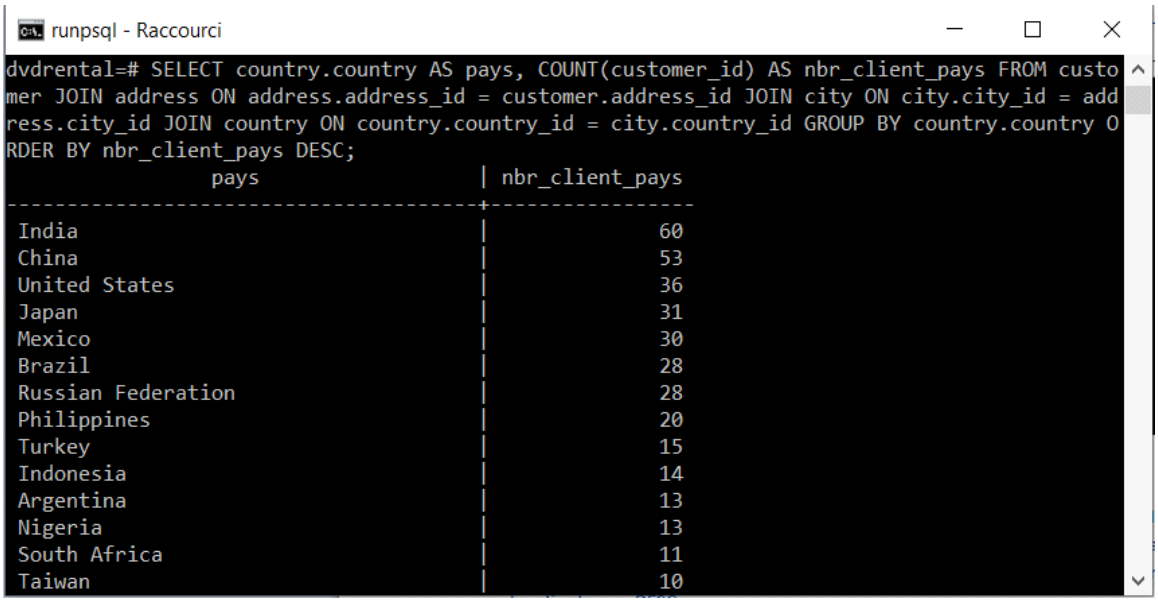
-> SELECT film.title, film.rental_duration AS nbre_jour_location_autorise , COUNT(DATE_PART('day',

rental.return_date - rental.rental_date)) AS nbre_de_retard FROM rental JOIN inventory ON inventory.inventory_id = rental.inventory_id JOIN film ON film.film_id = inventory.film_id WHERE DATE_PART('day', rental.return_date - rental.rental_date) > film.rental_duration GROUP BY film.title, film.rental_duration ORDER BY title;



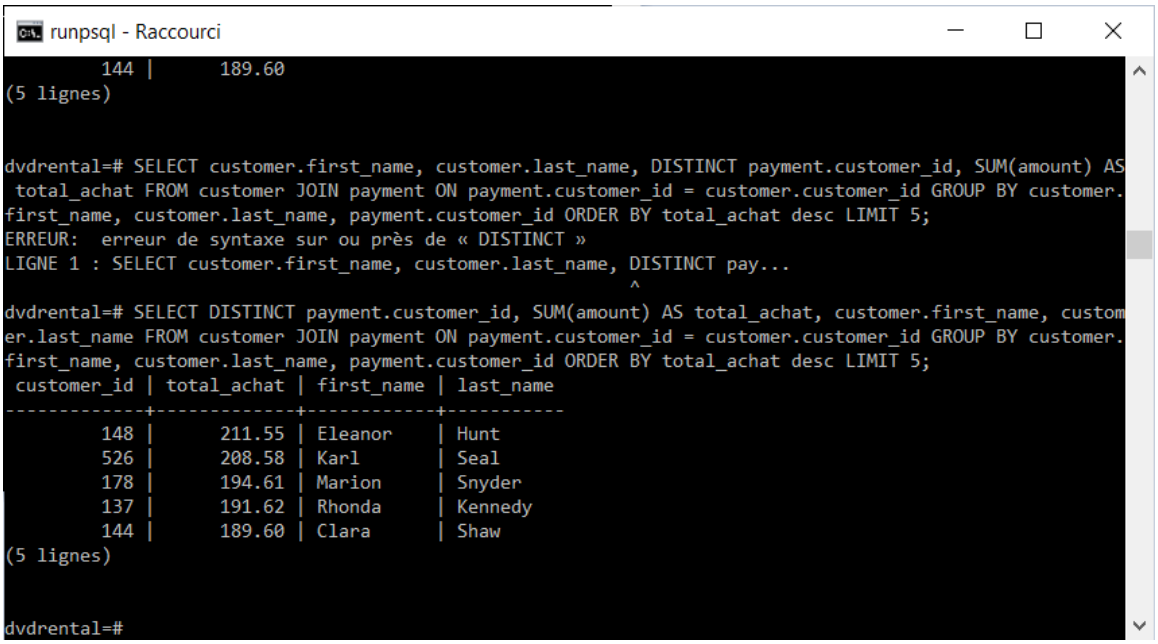
5. Déterminer le nombre de client par pays

-> SELECT country.country AS pays, COUNT(customer_id) AS nbr_client_pays FROM customer JOIN address ON address.address_id = customer.address_id JOIN city ON city.city_id = address.city_id JOIN country ON country.country_id = city.country_id GROUP BY country.country ORDER BY nbr_client_pays DESC;



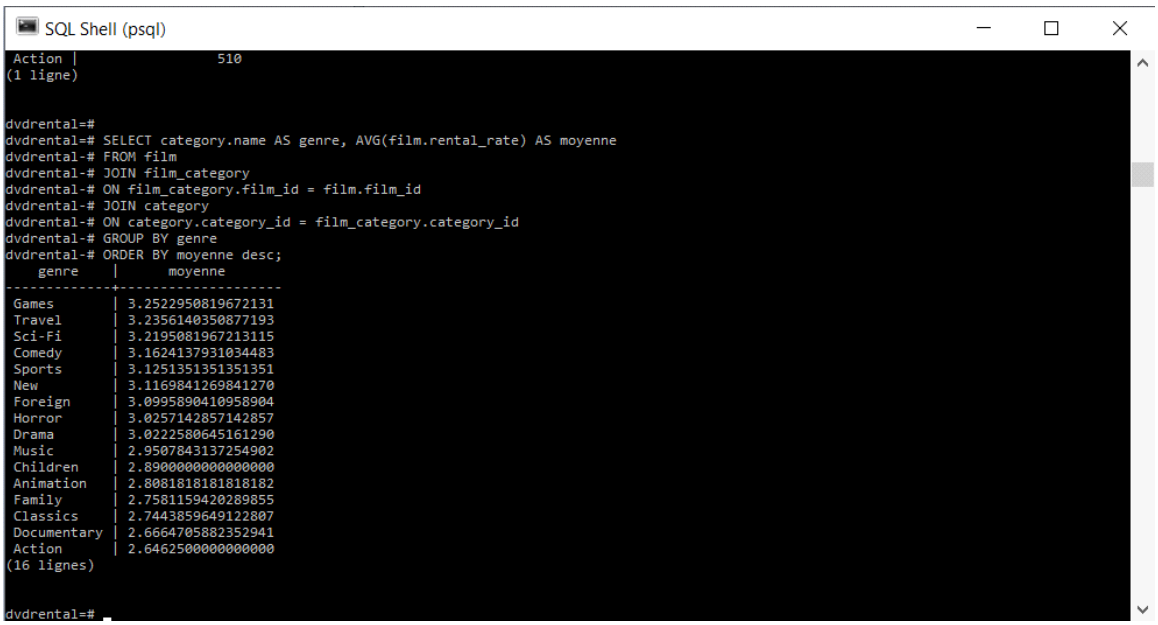
6. Trouver les 5 clients qui génèrent le plus de profits pour la sociétés.

-> SELECT DISTINCT customer.first_name, customer.last_name, payment.customer_id, SUM(amount) AS total_achat FROM customer JOIN payment ON payment.customer_id = customer.customer_id GROUP BY customer.first_name, customer.last_name, payment.customer_id ORDER BY total_achat desc LIMIT 5;



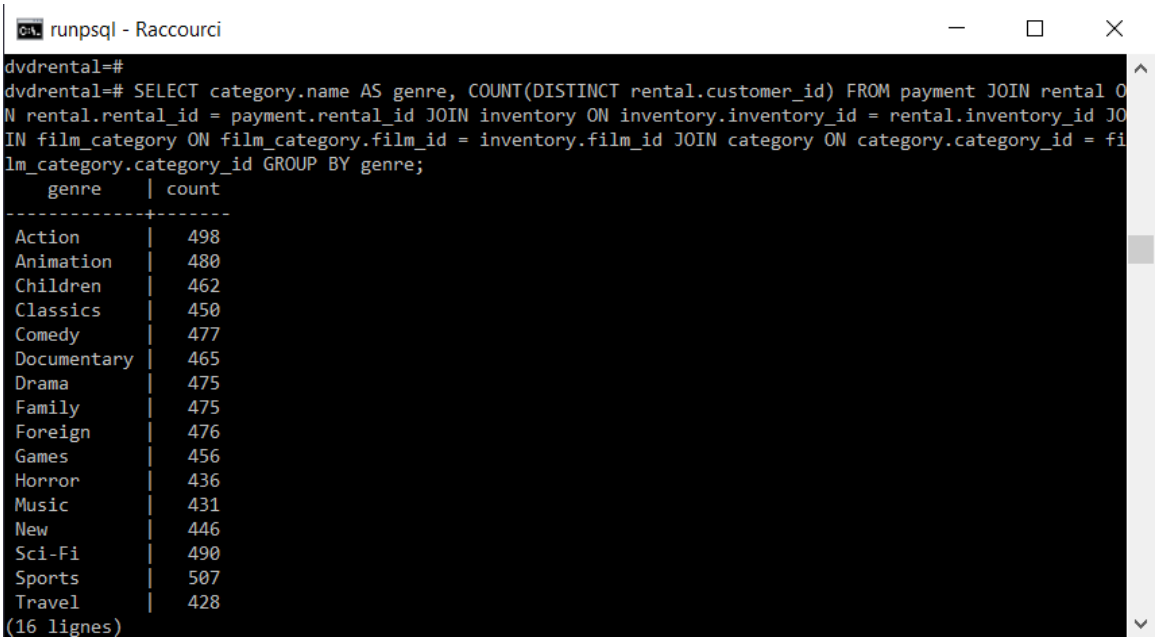
7. Quel est le tarif de location moyen pour chaque genre ? (du plus élevé au plus bas)

-> SELECT category.name AS genre, AVG(film.rental_rate) AS moyenne FROM film JOIN film_category ON film_category.film_id = film.film_id JOIN category ON category.category_id = film_category.category_id GROUP BY genre ORDER BY moyenne desc;



8. Peut-on savoir combien d'utilisateurs distincts ont loué chaque genre?

-> SELECT category.name AS genre, COUNT(DISTINCT rental.customer_id) FROM payment JOIN rental ON rental.rental_id = payment.rental_id JOIN inventory ON inventory.inventory_id = rental.inventory_id JOIN film_category ON film_category.film_id = inventory.film_id JOIN category ON category.category_id = film_category.category_id GROUP BY genre;



9. Combien de films loués ont été retournés tard, tôt et à temps ?

-> SELECT COUNT(CASE WHEN rental_duration > DATE_PART('day', return_date - rental_date) THEN 'Retourne en avance' END) AS en_avance, COUNT(CASE WHEN rental_duration = DATE_PART('day', return_date - rental_date) THEN 'Rendu a temps' END) AS a_temps, COUNT(CASE WHEN rental_duration < DATE_PART('day', return_date - rental_date) THEN 'Retourne en retard' END) AS en_retard FROM film JOIN inventory ON film.film_id = inventory.film_id JOIN rental ON inventory.inventory_id = rental.inventory_id;


Sélection runpsql - Raccourci
dvdrental=# SELECT COUNT(CASE WHEN rental_duration > DATE_PART('day', return_date - rental_
date) THEN 'Retourne en avance' END) AS en_avance, COUNT(CASE WHEN rental_duration = DATE_PA
RT('day', return_date - rental_date) THEN 'Rendu a temps' END) AS a_temps, COUNT(CASE WHEN r
ental_duration < DATE_PART('day', return_date - rental_date) THEN 'Retourne en retard' END)
AS en_retard FROM film JOIN inventory ON film.film_id = inventory.film_id JOIN rental ON inv
entory.inventory_id = rental.inventory_id;
 en_avance | a_temps | en_retard
-----------+---------+-----------
      7738 | 1720 | 6403
(1 ligne)
dvdrental=#