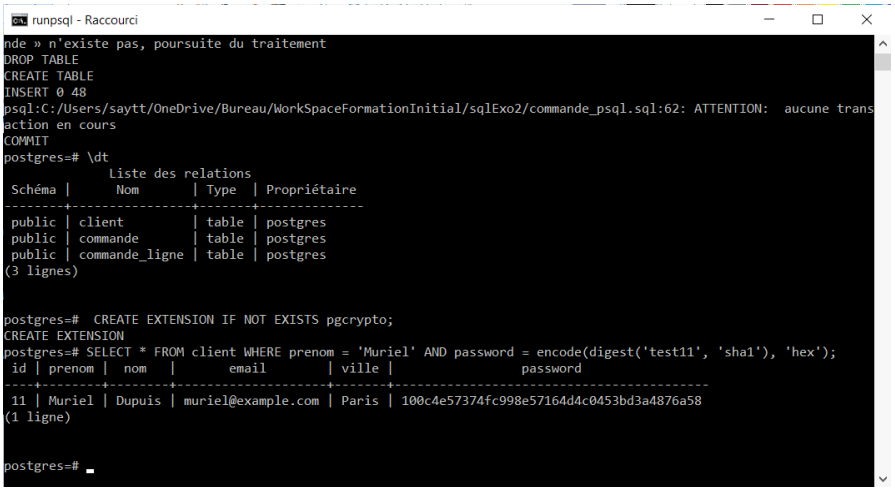
EXOS BD.STORE

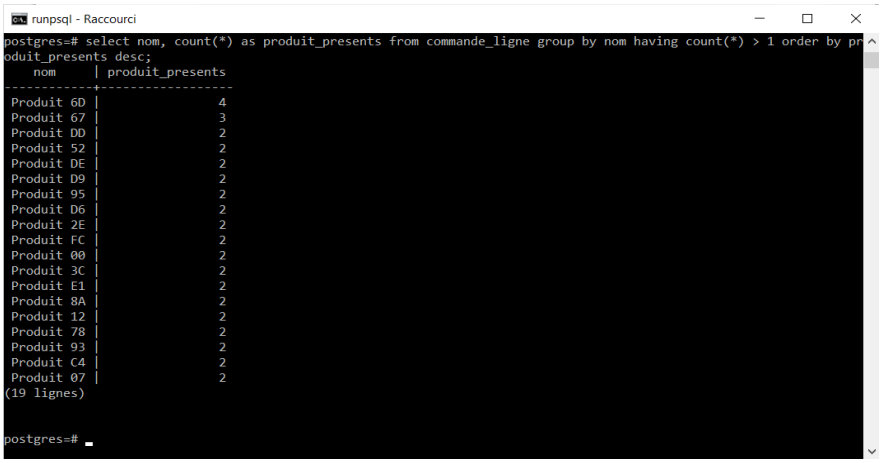
1. Obtenir l'utilisateur ayant le prénom "Muriel" et le mot de passe "test11", sachant que l'encodage du mot de passe est effectué avec l'algorithme Sha1.

-> SELECT * FROM client WHERE prenom = 'Muriel' AND password = encode(digest('test11', 'sha1'), 'hex');



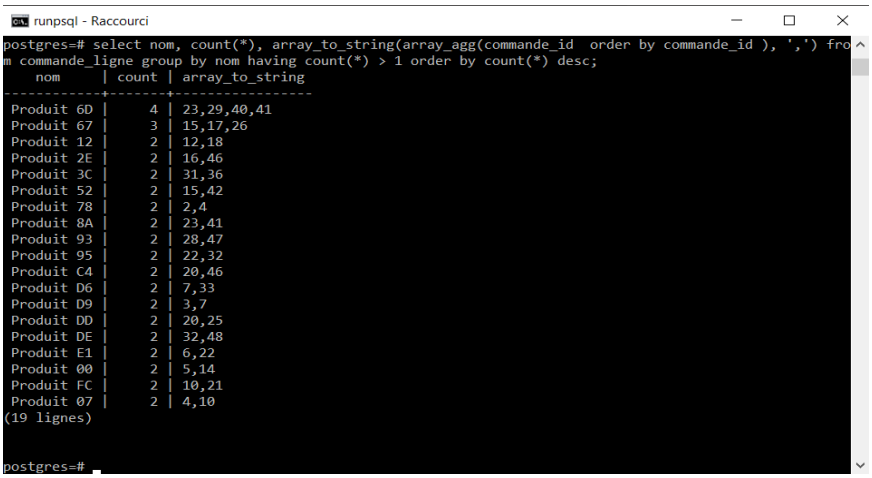
2. Obtenir la liste de tous les produits qui sont présent sur plusieurs commandes.

-> select nom, count(*) as produit_presents from commande_ligne group by nom having count(*) > 1 order by produit_presents desc;



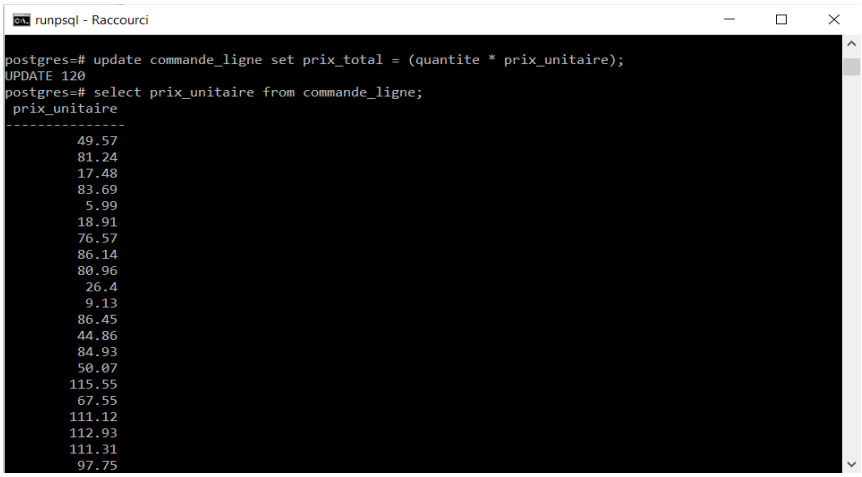
3. Obtenir la liste de tous les produits qui sont présent sur plusieurs commandes et y ajouter une colonne qui liste les identifiants des commandes associées.

-> select nom, count(*), array_to_string(array_agg(commande_id order by commande_id), ',') from commande_ligne group by nom having count(*) > 1 order by count(*) desc;



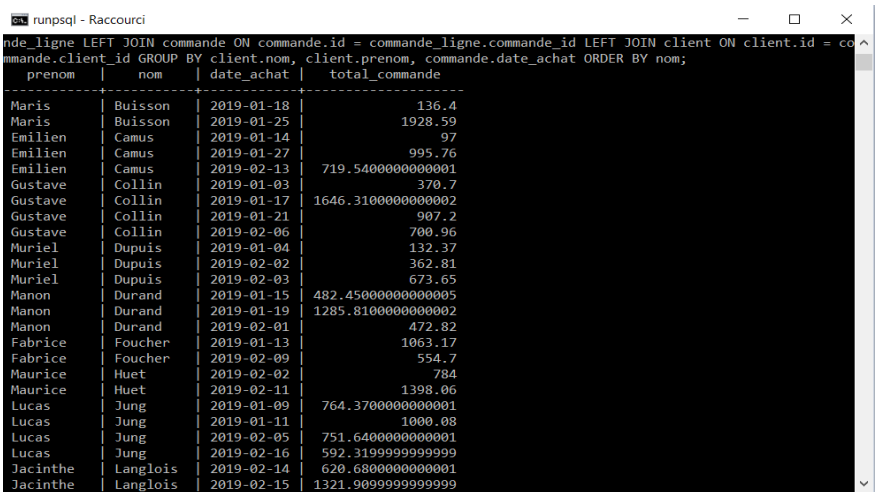
4. Enregistrer le prix total à l'intérieur de chaque ligne des commandes, en fonction du prix unitaire et de la quantité

-> update commande_ligne set prix_total = (quantite * prix_unitaire);



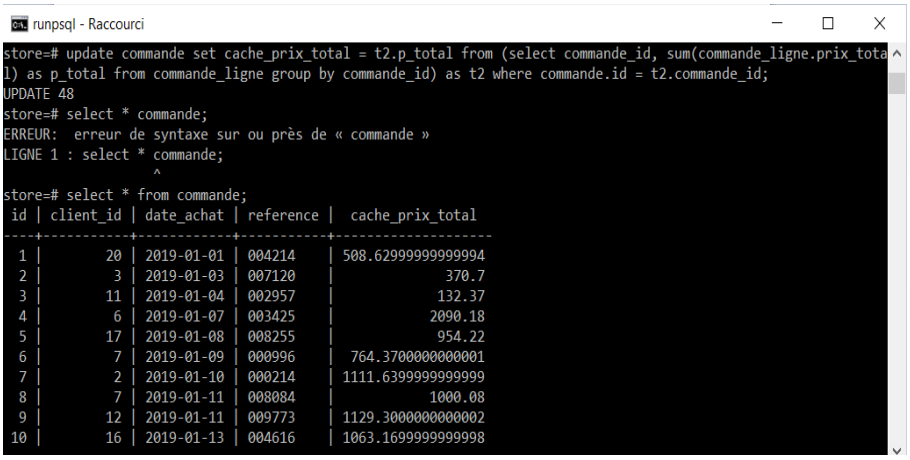
5. Obtenir le montant total pour chaque commande et y voir facilement la date associée à cette commande ainsi que le prénom et nom du client associé

-> SELECT client.prenom, client.nom, commande.date_achat, SUM(prix_total) AS total_commande FROM commande_ligne LEFT JOIN commande ON commande.id = commande_ligne.commande_id LEFT JOIN client ON client.id = commande.client_id GROUP BY client.nom, client.prenom, commande.date_achat ORDER BY nom;



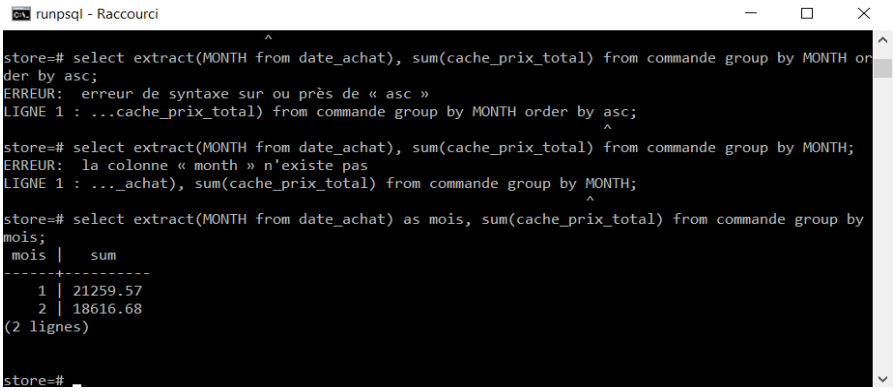
6. (difficulté très haute) Enregistrer le montant total de chaque commande dans le champ intitulé “cache_prix_total”

-> update commande set cache_prix_total = t2.p_total from (select commande_id, sum(commande_ligne.prix_total) as p_total from commande_ligne group by commande_id) as t2 where commande.id = t2.commande_id;



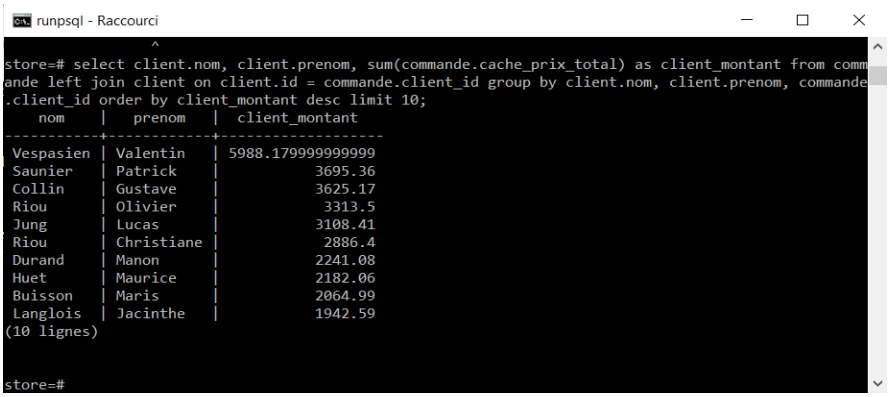
7. Obtenir le montant global de toutes les commandes, pour chaque mois

-> select extract(MONTH from date_achat) as mois, sum(cache_prix_total) from commande group by mois;



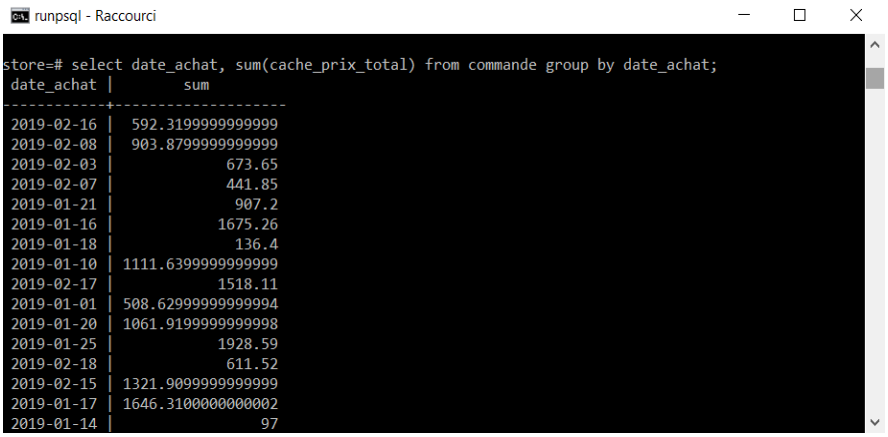
8. Obtenir la liste des 10 clients qui ont effectué le plus grand montant de commandes, et obtenir ce montant total pour chaque client.

-> select client.nom, client.prenom, sum(commande.cache_prix_total) as client_montant from commande left join client on client.id = commande.client_id group by client.nom, client.prenom, commande.client_id order by client_montant desc limit 10;



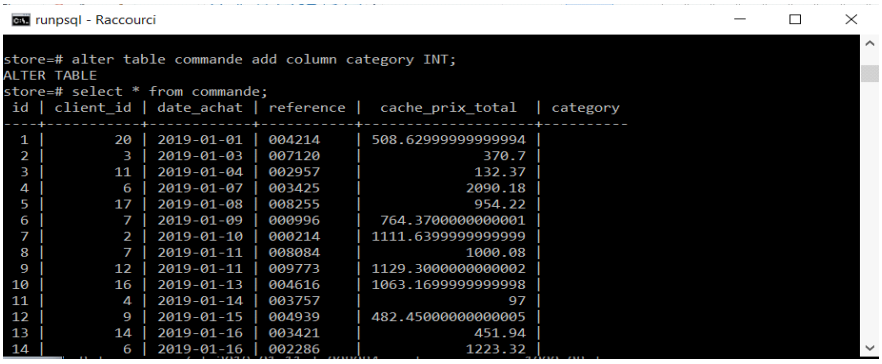
9. Obtenir le montant total des commandes pour chaque date

-> select date_achat, sum(cache_prix_total) as montant_date from commande group by date_achat order by date_achat asc;



10. Ajouter une colonne intitulée “category” à la table contenant les commandes. Cette colonne contiendra une valeur numérique

-> alter table commande add column category INT;



11. Enregistrer la valeur de la catégorie, en suivant les règles suivantes :

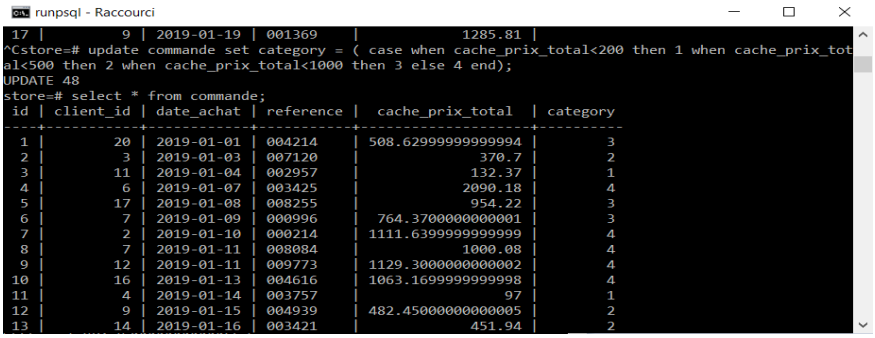
○ “1” pour les commandes de moins de 200€

○ “2” pour les commandes entre 200€ et 500€

○ “3” pour les commandes entre 500€ et 1.000€

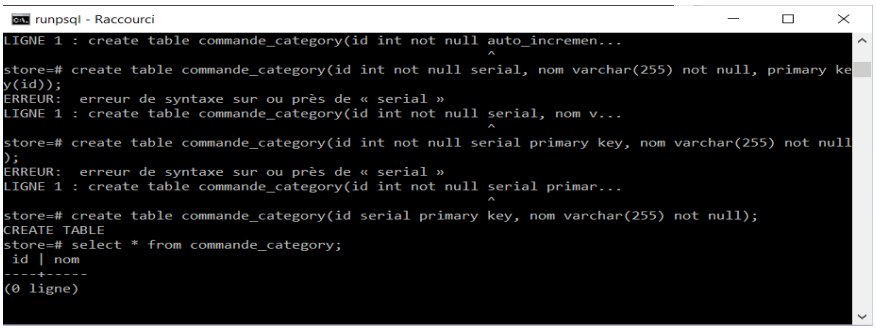
○ “4” pour les commandes supérieures à 1.000€

-> update commande set category = (case when cache_prix_total<200 then 1 when cache_prix_total<500 then 2 when cache_prix_total<1000 then 3 else 4 end);



12. Créer une table intitulée “commande_category” qui contiendra le descriptif de ces catégories

-> create table commande_category(id serial primary key, nom varchar(255) not null);



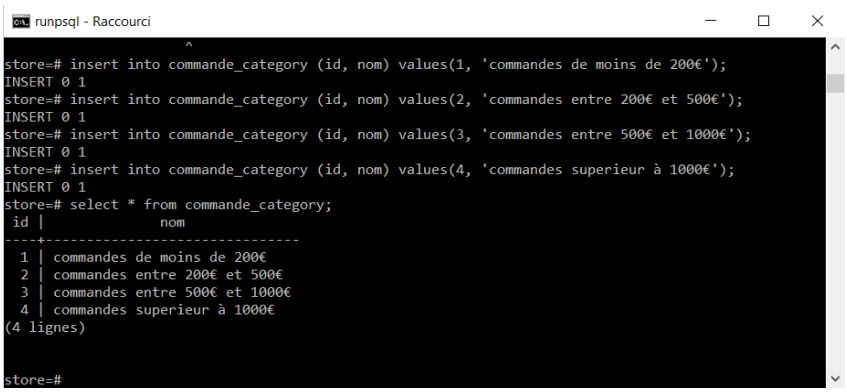
13. Insérer les 4 descriptifs de chaque catégorie au sein de la table précédemment créée

-> insert into commande_category (id, nom) values(1, 'commandes de moins de 200€');

-> insert into commande_category (id, nom) values(2, 'commandes entre 200€ et 500€');

-> insert into commande_category (id, nom) values(3, 'commandes entre 500€ et 1000€');

-> insert into commande_category (id, nom) values(4, 'commandes superieur à 1000€');



14. Supprimer toutes les commandes (et les lignes des commandes) inférieur au 1er février 2019. Cela doit être effectué en 2 requêtes maximum

-> delete from commande_ligne where commande_id in (select id from commande where date_achat < '2019-02-01');

-> delete from commande where date_achat < '2019-02-01';