Laboratoire 2 - DDL DML

Groupe: numéro 7

Auteurs: Piemontesi Gwendal, Quinn Calum, Trüeb Guillaume

Date: 17 décembre 2023

Sommaire

- Introduction
- Partie 1 Requêtes
 - o Point 1
 - o Point 2
 - Point 3
 - o Point 4
 - Point 5
 - Point 6
 - o Point 7
 - Point 8
 - o Point 9
 - o Point 10
 - o Point 11
 - o Point 12
 - o Point 13
 - o Point 14
- Partie 2 Cl
 - Script pour la Cl
 - Tests effectués
- Conclusion

Introduction

L'objectif de ce laboratoire était de pratiquer les commandes SQL en répondant à une série de questions. Ces questions se basaient sur un schéma EA ainsi que le script permettant de créer la base de données et d'y insérer des tupples. Il nous a été demandé de fournir un rapport celui-ci est constitué de deux parties.

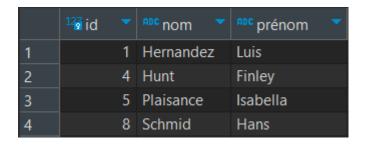
La première résumant les différentes points demandés avec les requêtes que nous avons écrites ainsi que le résultat obtenu pour chaque requête. La deuxième partie contient un script qui vient s'ajouter aux requêtes pour implémenter une contrainte d'intégrité qui n'autorise pas d'équipement lit avec un nom ne contenant pas le mot lit ainsi que les tests que nous avons effectués.

Partie 1 - Requêtes

Point 1

Les clients ayant fait au moins une réservation dans un hôtel se trouvant dans la ville dans laquelle ils habitent.

```
SELECT DISTINCT Client.id,Client.nom,Client.prénom FROM Client
    INNER JOIN Réservation ON Client.id = Réservation.idClient
    INNER JOIN Hôtel ON Réservation.idChambre = Hôtel.id
WHERE Client.idVille = Hôtel.idVille;
```

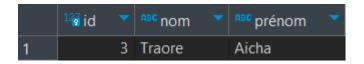


Point 2

Le prix minimum, maximum et moyen pour passer une nuit dans une chambre d'hôtel dans la ville de Montreux.



Les clients qui n'ont fait des réservations que dans des hôtels de 2 étoiles ou moins.



Point 4

Le nom des villes avec au moins un hôtel qui n'a aucune réservation.

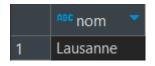
```
SELECT DISTINCT Ville.nom

FROM Hôtel

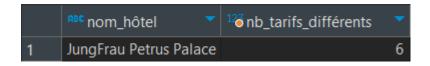
INNER JOIN Ville ON Ville.id = Hôtel.idVille

LEFT JOIN Réservation ON Réservation.idChambre = Hôtel.id

WHERE Réservation IS NULL;
```



L'hôtel qui a le plus de tarifs de chambres différents.



Point 6

Les clients ayant réservé plus d'une fois la même chambre. Indiquer les clients et les chambres concernées.

	¹² 8 id ▼	nom •	nom_hotel	¹¼ numérochambre ▼
1	1	Hernandez	Hôtel Royal	33 ௴
2	8	Schmid	Antique Boutique Hôtel	3 ௴
3	5	Plaisance	Kurz Alpinhotel	2 ☑
4	5	Plaisance	Antique Boutique Hôtel	2 ☑

Les membres de l'hôtel "Kurz Alpinhotel" qui n'y ont fait aucune réservation depuis qu'ils en sont devenus membre.



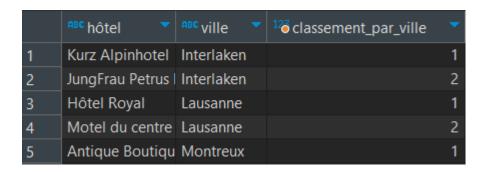
Point 8

Les villes, classées dans l'ordre décroissant de leur capacité d'accueil totale (nombre de places des lits de leurs hôtels).



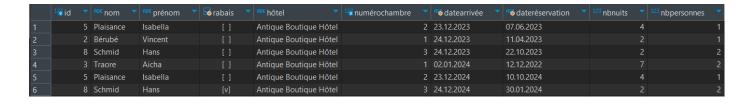
Les hôtels avec leur classement par ville en fonction du nombre de réservations.

```
SELECT Hôtel.nom, Ville.nom,
RANK() OVER (PARTITION BY Ville.nom
ORDER BY SUM(
CASE
WHEN Réservation.idchambre = Hôtel.id
THEN 1
ELSE 0
END)
DESC) AS Classement_par_ville
FROM Hôtel
INNER JOIN Ville ON Hôtel.idville = Ville.id
LEFT JOIN Réservation ON Hôtel.id = Réservation.idchambre
GROUP BY Ville.nom, Hôtel.nom
ORDER BY Ville.nom, Classement_par_ville;
```



Lister, par ordre d'arrivée, les prochaines réservations pour l'hôtel "Antique Boutique Hôtel" en indiquant si le client a obtenu un rabais.

```
SELECT Client.id, Client.nom, Client.prénom,
  (Membre IS NOT NULL AND Réservation.dateréservation > Membre.depuis) AS
Rabais, Hôtel.nom AS Hôtel, Chambre.numéro AS numérochambre,
  TO_CHAR(Réservation.datearrivée, 'DD.MM.YYYY') AS datearrivée,
  TO_CHAR(Réservation.dateréservation,'DD.MM.YYYY') AS dateréservation,
Réservation.nbnuits, Réservation.nbpersonnes
FROM Réservation
        INNER JOIN Client ON Réservation.idclient = Client.id
        INNER JOIN Chambre ON Réservation.idchambre = Chambre.idhôtel AND
Réservation.numérochambre = Chambre.numéro
        INNER JOIN Hôtel ON Réservation.idchambre = Hôtel.id AND Hôtel.nom =
'Antique Boutique Hôtel'
        LEFT JOIN Membre ON Client.id = Membre.idclient AND Hôtel.id =
Membre.idhôtel
WHERE Réservation.datearrivée > now()
ORDER BY Réservation.datearrivée;
```



Les réservations faites dans des chambres qui ont un nombre de lits supérieur au nombre de personnes de la réservation.

```
SELECT Client.id, Client.nom, Client.prénom, Hôtel.nom AS Hôtel,
Chambre.numéro AS numérochambre,
   TO_CHAR(Réservation.datearrivée,'DD.MM.YYYY') AS datearrivée,
   TO_CHAR(Réservation.dateréservation,'DD.MM.YYYY') AS dateréservation,
   Réservation.nbnuits, Réservation.nbpersonnes
FROM Réservation
        INNER JOIN Client ON Réservation.idclient = Client.id
        INNER JOIN Chambre ON Réservation.idchambre = Chambre.idhôtel

AND Réservation.numérochambre = Chambre.numéro
        INNER JOIN chambre_equipement ON Chambre.idhôtel =
Chambre_equipement.idchambre

AND Chambre.numéro = chambre_equipement.numérochambre

AND Chambre_equipement.quantité > Réservation.nbpersonnes
        INNER JOIN Hôtel ON Réservation.idchambre = Hôtel.id
ORDER BY Hôtel, Chambre.numéro;
```

	12% id	•	^{ABC} nom ▼	prénom •	ABC hôtel	12 numérochambre	•	⁸ 6 datearrivée ▼	⁸⁷ 6 dateréservation ▼	123 nbnuits	123 nbpersonnes
1		5	Plaisance	Isabella	Antique Boutique Hôtel		2	23.12.2023	07.06.2023	4	
2			Plaisance	Isabella	Antique Boutique Hôtel			23.12.2024	10.10.2024	4	
3			Hernandez	Luis	Hôtel Royal			02.03.2022	02.02.2022	3	
4			Plaisance	Isabella	Hôtel Royal			18.05.2012	16.05.2012	1	
5			Bérubé	Vincent	JungFrau Petrus Palace	5	51	28.12.2022	02.03.2022	5	

Les chambres à Lausanne ayant au moins une TV et un lit à 2 places.

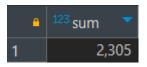


Pour l'hôtel "Hôtel Royal", lister toutes les réservations en indiquant de combien de jours elles ont été faites à l'avance (avant la date d'arrivée) ainsi que si la réservation a été faite en tant que membre de l'hôtel. Trier les résultats par ordre des réservations (en 1er celles faites le plus à l'avance), puis par clients (ordre croissant du nom puis du prénom).

```
SELECT Client.id, Client.nom, Client.prénom,
  (Membre IS NOT NULL AND Réservation.dateréservation > Membre.depuis) AS
Membre, Hôtel.nom AS Hôtel, Chambre.numéro AS numérochambre,
  TO_CHAR(Réservation.datearrivée, 'DD.MM.YYYY') AS datearrivée,
  TO CHAR(Réservation.dateréservation,'DD.MM.YYYY') AS dateréservation,
  (Réservation.datearrivée - Réservation.dateréservation) AS Avance,
Réservation.nbnuits, Réservation.nbpersonnes
FROM Réservation
        INNER JOIN Client ON Réservation.idclient = Client.id
        INNER JOIN Chambre ON Réservation.idchambre = Chambre.idhôtel AND
Réservation numérochambre = Chambre numéro
        INNER JOIN Hôtel ON Réservation.idchambre = Hôtel.id AND Hôtel.nom =
'Hôtel Royal'
        LEFT JOIN Membre ON Client.id = Membre.idclient AND Hôtel.id =
Membre.idhôtel
ORDER BY Avance DESC, Client.nom, Client.prénom;
```

	¹² id ▼	^{ABC} nom ▼	^{ABC} prénom ▼	™ membre	^{ABC} hôtel ▼	123 numérochambre	⁸⁶ datearrivée ▼	** dateréservation	¹ <mark>76</mark> avance ▼	123 nbnuits	123 nbpersonnes
1		Hernandez	Luis		Hôtel Royal		02.03.2022	02.02.2022	28		
2		Hernandez			Hôtel Royal		12.10.2023	01.10.2023	11		
3		Plaisance	Isabella		Hôtel Royal		07.01.2020	04.01.2020			
4		Plaisance	Isabella		Hôtel Royal		18.05.2012	16.05.2012			
5		Hernandez			Hôtel Royal		23.02.2022	22.02.2022			
6		Hunt	Finley		Hôtel Royal	23	06.10.2023	06.10.2023			
7		Plaisance	Isabella	[v]	Hôtel Royal		07.01.2020	07.01.2020			

Calculer le prix total de toutes les réservations faites pour l'hôtel "Hôtel Royal".



Partie 2 - CI

Script pour la Cl

```
CREATE OR REPLACE FUNCTION controleNomLit() RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN

IF NOT EXISTS (
SELECT 1 FROM Equipement
WHERE id = NEW.idEquipement AND lower(nom) LIKE '%lit%'
) THEN
RAISE EXCEPTION 'Le nom doit contenir le mot lit';
END IF;
RETURN NEW;
END $$;

CREATE TRIGGER CI_LIT BEFORE INSERT OR UPDATE ON Lit
FOR EACH ROW WHEN (NEW.idEquipement IS NOT NULL)
EXECUTE FUNCTION controleNomLit();
```

Tests effectués

Nous avons essayer d'ajouter des équipements avec des noms variés contenant parfois le mot lit et parfois pas.

```
INSERT INTO Equipement(nom) VALUES ('Li Queen size');
INSERT INTO Equipement(nom) VALUES ('Lits Queen size');
INSERT INTO Equipement(nom) VALUES ('Literie Queen size');
INSERT INTO Equipement(nom) VALUES ('liT Queen size');
INSERT INTO Equipement(nom) VALUES ('LIT Queen size');
INSERT INTO Equipement(nom) VALUES ('lit Queen size');
INSERT INTO Equipement(nom) VALUES ('Li t Queen size');
INSERT INTO Equipement(nom) VALUES ('Queen Lit size');
INSERT INTO Equipement(nom) VALUES ('Queen LIT size');
INSERT INTO Equipement(nom) VALUES ('Queen lit size');
INSERT INTO Equipement(nom) VALUES ('Queen litière size');
INSERT INTO Equipement(nom) VALUES ('Queen Li t size');
INSERT INTO Equipement(nom) VALUES ('Queen Li size');
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (8, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (9, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (10, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (11, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (12, 2);
```

```
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (13, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (14, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (15, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (16, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (17, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (18, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (19, 2);
INSERT INTO Lit(idEquipement, nbPlaces) VALUES (20, 2);
```

Les résultats des insertions suivants sont les suivantes:

- Erreur ("Li ")
- Ok
- Ok
- Ok
- Ok
- Ok
- Erreur ("Li t")
- Ok
- Ok
- Ok
- Ok
- Erreur ("Li t")
- Erreur ("Li ")

Conclusion

La première partie nous a permis de nous familiariser avec la base de données ainsi que son contenu. Notre démarche, consistant à écrire une requête et ensuite contrôler que son résultat soit correcte, nous a permis de bien avancer et d'être sûr de nos résultats.

Pour la deuxième partie nous avons employé un trigger pour le contrôle d'une insertion. Les tests ont vérifié le bon fonctionnement de ce contrôle avec des cas variés et ressemblant à des erreurs possibles. Les résultats de notre script de test sont parfaitement accordés avec nos attentes.

Ce laboratoire nous a donc permis d'approfondir nos connaissances en SQL ainsi que d'apprendre à repartir d'un travail déjà existant pour y apporter des améliorations ou nouvelles fonctionnalitées.