
BDR : Laboratoire no 2

Introduction

Pour ce laboratoire, nous avons comme donnée un schéma entité-association d'une base de donnée ainsi que le script permettant de la créer et de remplir ses tables. Nous avons ensuite 15 requêtes à effectuer dessus. Le but de ce travail était plus exactement de pratiquer les requêtes SELECT. Ce rapport présente les requêtes que nous avons écrites ainsi que des captures d'écran des résultats obtenus. Nous retranscrivons avant chaque requête ce qui était textuellement demandé.

Résultats

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 AS "Id Client",  
Client.nom AS "Nom Client",  
Client.prénom AS "Prénom Client"
```

FROM

```
Client  
INNER JOIN Réservation ON Réservation.idClient = Client.id  
INNER JOIN Hôtel ON Réservation.idChambre = Hôtel.id
```

WHERE

```
Client.idVille = Hôtel.idVille;
```

	Id Client integer	Nom Client character varying (50)	Prénom Client character varying (50)
1	1	Hernandez	Luis
2	4	Hunt	Finley
3	5	Plaisance	Isabella
4	8	Schmid	Hans

Figure 1: résultat

2. Le prix minimum et le prix maximum pour passer une nuit dans une chambre d'hôtel dans la ville d'Interlaken.

```
SELECT
  MIN(prixParNuit) AS "Prix Minimum",
  MAX(prixParNuit) AS "Prix Maximum"
FROM
  Chambre
  INNER JOIN Hôtel ON Chambre.idHôtel = Hôtel.id
  INNER JOIN Ville ON Hôtel.idVille = Ville.id
WHERE
  Ville.nom = 'Interlaken';
```

	Prix Minimum smallint	Prix Maximum smallint
1	200	3000

Figure 2: résultat

3. Pour l'hôtel "JungFrau Petrus Palace", lister le prix moyen des chambres par étage. Classer les résultats par ordre croissant du prix moyen.

```
SELECT
  Hôtel.nom AS "Nom Hôtel",
  étage,
  ROUND(AVG(prixparnuit), 2) AS "Prix moyen par étage"
FROM
  chambre
  INNER JOIN hôtel ON chambre.idhôtel = hôtel.id
WHERE
  hôtel.nom = 'JungFrau Petrus Palace'
GROUP BY
  Chambre.étage,
  Hôtel.nom
ORDER BY
  "Prix moyen par étage";
```

	Nom Hôtel character varying (30)	étage smallint	Prix moyen par étage numeric
1	JungFrau Petrus Palace	1	200.00
2	JungFrau Petrus Palace	2	220.00
3	JungFrau Petrus Palace	3	240.00
4	JungFrau Petrus Palace	4	300.00
5	JungFrau Petrus Palace	5	1333.33

Figure 3: résultat

4. Les hôtels proposant au moins une chambre disposant de plus d'une baignoire.

SELECT

DISTINCT Hôtel.nom **AS** "Hôtel ayant plus d'une baignoire dans une chambre"

FROM

Hôtel

INNER JOIN Chambre_Equipement **ON** Hôtel.id = Chambre_Equipement.idChambre

WHERE

Chambre_Equipement.nomEquipement = 'Baignoire'

AND Chambre_Equipement.quantité > 1;

	Hôtels ayant plus d'une baignoire dans une chambre character varying (30)
1	JungFrau Petrus Palace

Figure 4: résultat

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

WITH tarifHôtel **AS** (

SELECT

Hôtel.nom **AS** "Nom Hôtel",

COUNT(**DISTINCT** Chambre.prixParNuit) **AS** "nbtarif"

FROM

Chambre

INNER JOIN Hôtel **ON** Hôtel.id = Chambre.idHôtel

```

GROUP BY
    Hôtel.nom
)
SELECT *
FROM
    tarifHôtel
ORDER BY
    tarifHôtel.nbtarif DESC
LIMIT
    1;

```

	Nom Hôtel	nbtarif
	character varying (30)	bigint
1	JungFrau Petrus Palace	6

Figure 5: résultat

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

```

SELECT
    Client.id AS "Id Client",
    Client.nom AS "Nom Client",
    Client.prénom AS "Prénom Client",
    Hôtel.nom AS "Nom Hôtel",
    Réservation.numéroChambre AS "Numéro Chambre"
FROM
    Client
    INNER JOIN Réservation ON Client.id = Réservation.idClient
    INNER JOIN Hôtel ON Réservation.idChambre = Hôtel.id
GROUP BY
    Client.id,
    Réservation.numéroChambre,
    Hôtel.id
HAVING
    COUNT(*) > 1;

```

	Id Client integer	Nom Client character varying (50)	Prénom Client character varying (50)	Nom Hôtel character varying (30)	Numéro Chambre integer
1	5	Plaisance	Isabella	Antique Boutique Hôtel	2
2	1	Hernandez	Luis	Hôtel Royal	33
3	8	Schmid	Hans	Antique Boutique Hôtel	3
4	5	Plaisance	Isabella	Kurz Alpinhotel	2

Figure 6: résultat

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

```

WITH MembresAlpine AS(
  SELECT
    DISTINCT Client.id AS id,
    Client.nom AS nom,
    Client.prénom AS prénom,
    Membre.depuis AS membreDepuis,
    Membre.idHôtel AS idHôtel
  FROM
    Membre
    INNER JOIN Hôtel ON Membre.idHôtel = Hôtel.id
    INNER JOIN Client ON Membre.idClient = Client.id
  WHERE
    Hôtel.nom = 'Kurz Alpinhotel'
)
SELECT
  MembresAlpine.id,
  MembresAlpine.nom,
  MembresAlpine.prénom
FROM
  MembresAlpine
EXCEPT
SELECT
  MembresAlpine.id,
  MembresAlpine.nom,
  MembresAlpine.prénom
FROM
  MembresAlpine

```

```

INNER JOIN Réservation ON (MembresAlpine.id, MembresAlpine.idHôtel) =
  ↳ (Réservation.idClient, Réservation.idChambre)

```

WHERE

```

Réservation.dateRéservation >= MembresAlpine.membreDepuis;

```

	id integer	nom character varying (50)	prénom character varying (50)
1	7	Deeann	Hibbert
2	4	Hunt	Finley

Figure 7: résultat

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

SELECT

```

Ville.nom AS "Villes",
SUM(Lit.nbPlaces * Chambre_Equipement.quantité) AS "Capacité accueil"

```

FROM

```

Ville

```

```

INNER JOIN Hôtel ON Ville.id = Hôtel.idVille

```

```

INNER JOIN Chambre_Equipement ON Hôtel.id = Chambre_Equipement.idChambre

```

```

INNER JOIN Lit ON Chambre_Equipement.nomEquipement = Lit.nomEquipement

```

GROUP BY

```

Ville.nom

```

ORDER BY

```

SUM(Lit.nbPlaces * Chambre_Equipement.quantité) DESC;

```

	Villes character varying (30)	Capacité accueil bigint
1	Interlaken	94
2	Lausanne	28
3	Montreux	11

Figure 8: résultat

9. Les villes dans lesquelles ont été faites le plus grand nombre de réservations.

```
WITH nbSéjourParVille AS (  
  SELECT  
    Hôtel.idVille,  
    COUNT(Hôtel.idVille) AS nbSéjour  
  FROM  
    Hôtel  
    INNER JOIN Réservation ON Hôtel.id = Réservation.idChambre  
  GROUP BY  
    Hôtel.idVille  
)  
SELECT  
  Ville.nom AS "Ville",  
  nbSéjourParVille.nbSéjour  
FROM  
  Ville  
  INNER JOIN nbSéjourParVille ON Ville.id = nbSéjourParVille.idVille  
WHERE  
  nbSéjourParVille.nbSéjour >= ALL (  
    SELECT  
      DISTINCT nbSéjourParVille.nbSéjour  
    FROM  
      nbSéjourParVille  
  );
```

	Ville character varying (30)	nbséjour bigint
1	Interlaken	7
2	Lausanne	7

Figure 9: résultat

10. Les chambres réservées pour la nuit du 24 décembre (de cette année).

```
WITH CurrentYear AS (  
  SELECT  
    *  
  FROM  
    MAKE_DATE(EXTRACT(YEARFROMCURRENT_DATE):: SMALLINT,12,24) AS christmas
```

```

)
SELECT
    DISTINCT Hôtel.nom AS "Nom Hôtel",
    Chambre.Numéro
FROM
    Chambre
    INNER JOIN Hôtel ON Chambre.idHôtel = Hôtel.id
    INNER JOIN Réservation ON Réservation.idChambre = Chambre.idHôtel
    AND Réservation.numéroChambre = Chambre.numéro
WHERE
    Réservation.dateArrivée <= (
        SELECT
            christmas
        FROM
            CurrentYear
    )
    AND Réservation.dateArrivée + Réservation.nbNuits >= (
        SELECT
            christmas
        FROM
            CurrentYear
    )

```

	Nom Hôtel character varying (30)	numéro smallint
1	Antique Boutique Hôtel	2
2	Antique Boutique Hôtel	3

Figure 10: résultat

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

```

SELECT
    Réservation.idClient AS "Id Client",
    Client.nom AS "Nom Client",
    Client.prénom AS "Prénom Client",
    Hôtel.nom AS "Nom Hôtel",
    Réservation.numéroChambre AS "Numéro Chambre",
    TO_CHAR(Réservation.dateRéservation, 'DD/MM/YYYY') AS "Date Réservation",

```

```
TO_CHAR(Réservation.dateArrivée, 'DD/MM/YYYY') AS "Date Arrivée",
Réservation.nbNuits,
Réservation.nbPersonnes
```

FROM

```
Réservation
INNER JOIN Client ON Réservation.idClient = Client.id
INNER JOIN Hôtel ON Réservation.idChambre = Hôtel.id
INNER JOIN Chambre_Equipement ON Réservation.numéroChambre =
↳ Chambre_Equipement.numéroChambre
AND Réservation.idChambre = Chambre_Equipement.idChambre
INNER JOIN Lit ON Chambre_Equipement.nomEquipement = Lit.nomEquipement
```

GROUP BY

```
réservation.idclient,
Client.nom,
Client.prénom,
Hôtel.nom,
réservation.numérochambre,
réservation.dateréservation,
réservation.datearrivée,
Réservation.nbNuits,
Réservation.nbPersonnes
```

HAVING

```
SUM(Chambre_Equipement.quantité) > Réservation.nbPersonnes;
```


	 Id Client integer	 Nom Client character varying (50)	 Prénom Client character varying (50)	 Nom Hôtel character varying (30)	 Numéro Chambre integer	 Date Réservation text	 Date Arrivée text	 nbnuits smallint	 nbpersonnes smallint
1	1	Hernandez	Luis	Hôtel Royal	3	02/02/2020	02/03/2020	3	1
2	1	Hernandez	Luis	Hôtel Royal	33	22/02/2020	23/02/2020	2	1
3	5	Plaisance	Isabella	Antique Boutique Hôtel	2	07/06/2021	23/12/2021	4	1
4	5	Plaisance	Isabella	Antique Boutique Hôtel	2	10/10/2022	23/12/2022	4	1
5	5	Plaisance	Isabella	Hôtel Royal	3	16/05/2010	18/05/2010	1	1
6	8	Schmid	Hans	Antique Boutique Hôtel	3	22/10/2021	24/12/2021	2	2
7	8	Schmid	Hans	Antique Boutique Hôtel	3	30/01/2022	24/12/2022	2	2

Figure 11: résultat

12. Les hôtels dont pas toutes les chambres sont équipées d'une TV. N'utiliser ni EXCEPT, ni INTERSECT.

SELECT

```
Hôtel.nom AS "Hôtels avec chambres sans TV"
```

FROM

```
Hôtel
INNER JOIN Chambre ON Chambre.idHôtel = Hôtel.id
```

WHERE

(Chambre.idHôtel, Chambre.numéro) **NOT IN** (

SELECT

Chambre_Equipement.idChambre,
Chambre_Equipement.numéroChambre

FROM

Chambre_Equipement

WHERE

Chambre_Equipement.nomEquipement = 'TV'

)

GROUP BY

Hôtel.nom;

	Hôtels ayant des chambres sans TV character varying (30)
1	Hôtel Royal
2	Motel du centre urbain

Figure 12: résultat

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

WITH ChambresLausanne **AS** (

SELECT

Hôtel.nom **AS** nomHôtel,
Chambre_Equipement.numéroChambre **AS** numéroChambre,
Chambre_Equipement.nomEquipement **AS** nomEquipement,
Chambre_Equipement.quantité **AS** quantitéEquipement,
Lit.nbPlaces **AS** nbPlacesLit

FROM

Hôtel

INNER JOIN Ville **ON** Ville.id = Hôtel.idVille

INNER JOIN Chambre_Equipement **ON** Chambre_Equipement.idChambre =
↪ Hôtel.id

LEFT JOIN Lit **ON** Lit.nomEquipement = Chambre_Equipement.nomEquipement

WHERE

Ville.nom = 'Lausanne'

)

SELECT

ChambresLausanne.nomHôtel **AS** "Nom Hôtel",

```

    ChambresLausanne.numéroChambre AS "Numéro Chambre"
FROM
    ChambresLausanne
WHERE
    ChambresLausanne.nomEquipement = 'TV'
    AND ChambresLausanne.quantitéEquipement >= 1
    AND ChambresLausanne.numéroChambre IN (
        SELECT
            ChambresLausanne.numéroChambre
        FROM
            ChambresLausanne
        WHERE
            ChambresLausanne.nbPlacesLit = 2
            AND ChambresLausanne.quantitéEquipement >= 1
    );

```

	Nom Hôtel character varying (30)	Numéro Chambre integer
1	Hôtel Royal	33
2	Hôtel Royal	100

Figure 13: résultat

14. 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 1^{er} celles faites le plus à l'avance), puis par clients (ordre croissant du nom puis du prénom).

```

WITH clientRoyal AS (
    SELECT
        DISTINCT Réservation.*
    FROM
        Réservation
        INNER JOIN Hôtel ON Hôtel.id = Réservation.idChambre
        INNER JOIN Membre ON Membre.idClient = Réservation.idClient
    WHERE
        Hôtel.nom = 'Hôtel Royal'
)
SELECT

```

```

DISTINCT Réservation.dateArrivée - Réservation.dateRéservation AS Avance,
Membre.depuis AS "Est Membre",
Client.id AS "Id Client",
Client.nom AS "Nom Client",
Client.prénom AS "Prénom Client",
Réservation.NuméroChambre AS "Numéro Chambre",
Réservation.dateArrivée AS "Date arrivée",
Réservation.dateRéservation AS "Date réservation",
Réservation.nbnuits AS "Nombre nuits",
Réservation.nbPersonnes AS "Nombre personnes"
FROM
Hôtel
INNER JOIN Réservation ON Réservation.idChambre = Hôtel.id
INNER JOIN Client ON Réservation.idClient = Client.id
LEFT JOIN Membre ON Réservation.idClient = Membre.idClient
AND Réservation.idChambre = Membre.idHôtel
WHERE
Hôtel.nom = 'Hôtel Royal'
ORDER BY
Avance ASC,
Client.nom ASC,
Client.prénom DESC;

```

	avance integer	Est Membre date	Id Client integer	Nom Client character varying (50)	Prénom Client character varying (50)	Numéro Chambre integer	Date arrivée date	Date réservation date	Nombre nuits smallint	Nombre personnes smallint
1	0	[null]	4	Hunt	Finley	23	2021-10-06	2021-10-06	2	2
2	0	2018-01-05	5	Plaisance	Isabella	33	2018-01-07	2018-01-07	5	2
3	1	[null]	1	Hernandez	Luis	33	2020-02-23	2020-02-22	2	1
4	2	2018-01-05	5	Plaisance	Isabella	3	2010-05-18	2010-05-16	1	1
5	3	2018-01-05	5	Plaisance	Isabella	1	2018-01-07	2018-01-04	5	1
6	11	[null]	1	Hernandez	Luis	33	2021-10-12	2021-10-01	1	2
7	29	[null]	1	Hernandez	Luis	3	2020-03-02	2020-02-02	3	1

Figure 14: résultat

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

```

SELECT
SUM(
CASE
--Rabais Membre
WHEN Réservation.dateRéservation > membre.depuis
THEN Chambre.prixParNuit * Réservation.nbNuits * (100 -
↪ Hôtel.rabaisMembre) / 100

```

```

-- Sans Rabais
ELSE Chambre.prixParNuit * Réservation.nbNuits
END
) AS "Prix total réservations"
FROM
  Réservation
  INNER JOIN Hôtel ON Réservation.idChambre = Hôtel.id
  INNER JOIN Chambre ON (Réservation.idChambre, Réservation.numéroChambre)
    ↳ = (Chambre.idHôtel, Chambre.numéro)
  LEFT JOIN Membre ON (Réservation.idChambre, Réservation.idClient) =
    ↳ (Membre.idHôtel, Membre.idClient)
WHERE
  Hôtel.nom = 'Hôtel Royal';

```

	Prix total reservations
	bigint
1	2305

Figure 15: résultat

Conclusion

Ce laboratoire nous a permis d'explorer les multiples possibilités des requêtes SELECT en SQL. Le plus grand défi a été certaines fois d'obtenir des résultats cohérents, d'autre fois d'écrire des requêtes pas trop longues.