

SQL - DBeaver

DBeaver est un outil de gestion de base de données universel et open source. Il offre une interface graphique conviviale permettant aux utilisateurs de se connecter, explorer, et gérer diverses bases de données, qu'elles soient relationnelles ou non. DBeaver prend en charge de nombreux systèmes de gestion de bases de données populaires tels que MySQL, PostgreSQL, Oracle, SQLite, et bien d'autres. Les fonctionnalités clés incluent l'exécution de requêtes SQL, la visualisation des schémas de base de données, la gestion des utilisateurs et la manipulation des données. Grâce à sa polyvalence, DBeaver est largement utilisé par les développeurs, administrateurs de bases de données et analystes pour simplifier le travail avec différentes bases de données.





SQL

Installation de DBeaver

03 / 01 / 2024
Version : 1
Page : 1 / 20

OBJECTIF : Cette section de la procédure vise à détailler la mise en place du logiciel DBeaver.

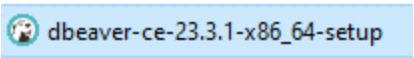
MODE OPÉRATOIRE :

1 – Première étape

Pour installer DBeaver, commencez par vous rendre sur le site officiel : <https://dbeaver.io/download/>. Une fois sur la page de téléchargement, choisissez le type d'installation adapté à votre système d'exploitation.

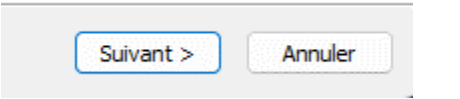
2 – Deuxième étape

Maintenant que vous avez téléchargé le programme d'installation, vous pouvez exécuter le fichier.



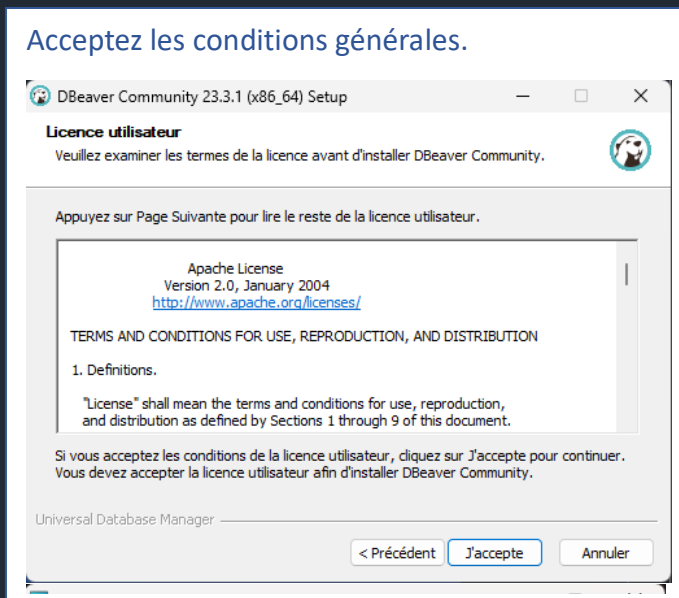
3 – Troisième étape

Une fois lancé, cliquez sur «Suivant».



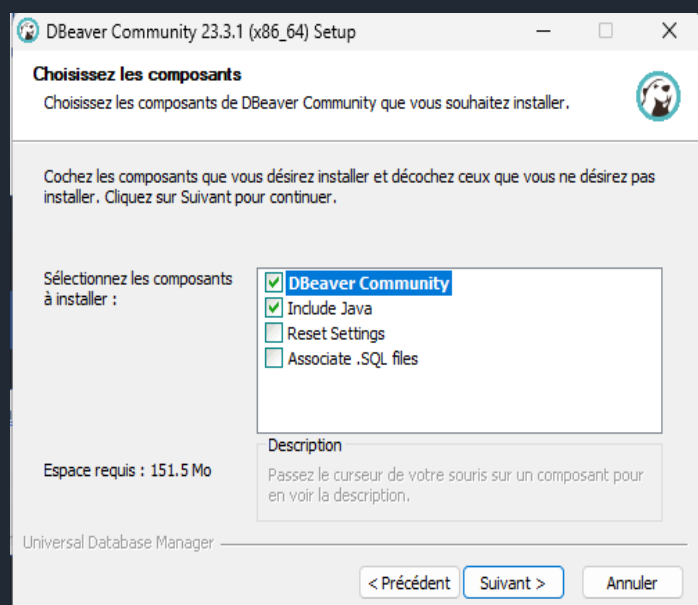
4 – Quatrième étape

Acceptez les conditions générales.



5 – Cinquième étape

Lorsque vous installez DBeaver, l'étape "Choisissez les composants" vous permet de sélectionner les fonctionnalités spécifiques que vous souhaitez installer avec DBeaver. Cette étape vous donne un contrôle sur les composants additionnels qui peuvent être inclus dans l'installation.





SQL

Installation de DBeaver

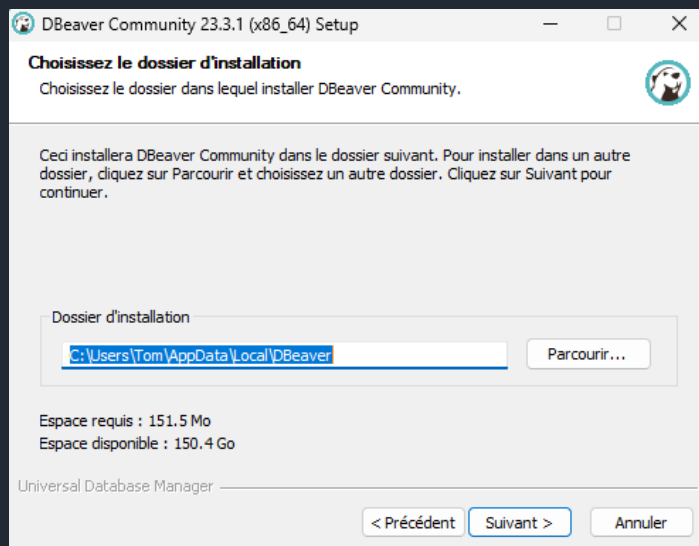
03 / 01 / 2023

Version : 1

Page : 3 / 20

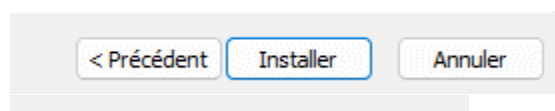
6 – Sixième étape

Une fois l'étape 5 terminée, DBeaver vous donne la possibilité de choisir l'emplacement d'installation du logiciel sur votre machine. Par défaut, l'application s'installe dans C:\Users\Tom\AppData\Local\DBeaver. Cependant, si vous souhaitez modifier l'emplacement du dossier, cliquez sur le bouton « Parcourir... » puis sélectionnez le nouvel emplacement où vous souhaitez télécharger le dossier.



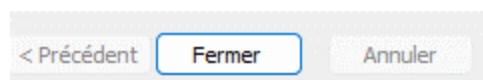
7 – Septième étape

Cliquez sur « Installer » pour confirmer l'installation.



8 – Huitième étape

Une fois l'installation terminée vous pouvez valider.



Vous pouvez également créer un raccourci sur le bureau.

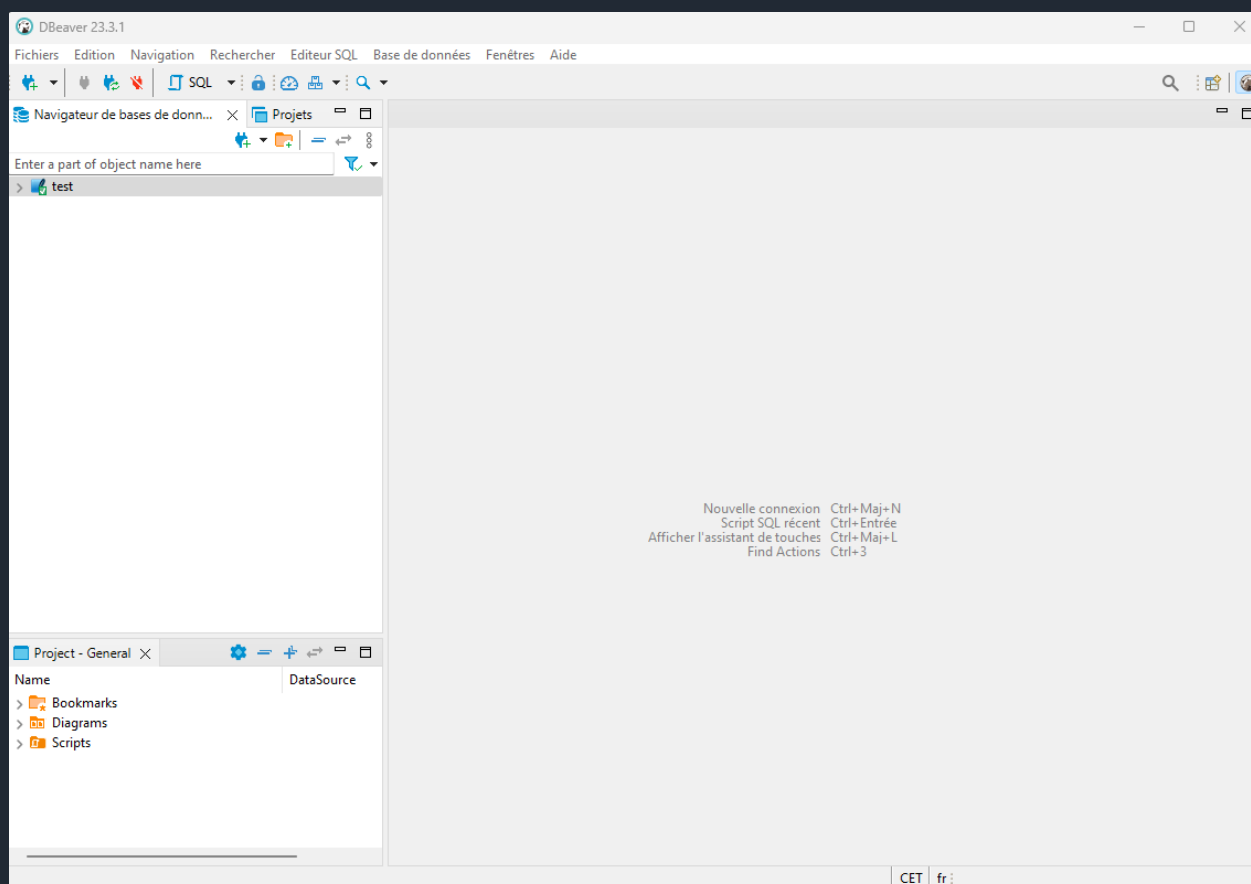
	<u>SQL</u> Mise en place de bases de données.	03 / 01 / 2023 Version : 1 Page : 4 / 20
---	---	--

OBJECTIF : Cette section de la procédure vise à détailler la mise en place de bases de données.

MODE OPÉRATOIRE :

1 – Première étape : Création d'une nouvelle base de données

Une fois que le logiciel DBeaver ouvert, vous devez sélectionner "Créer une Nouvelle connexion" pour pouvoir importer ou crée une nouvelle base de donner.





SQL

Mise en place de bases de données.

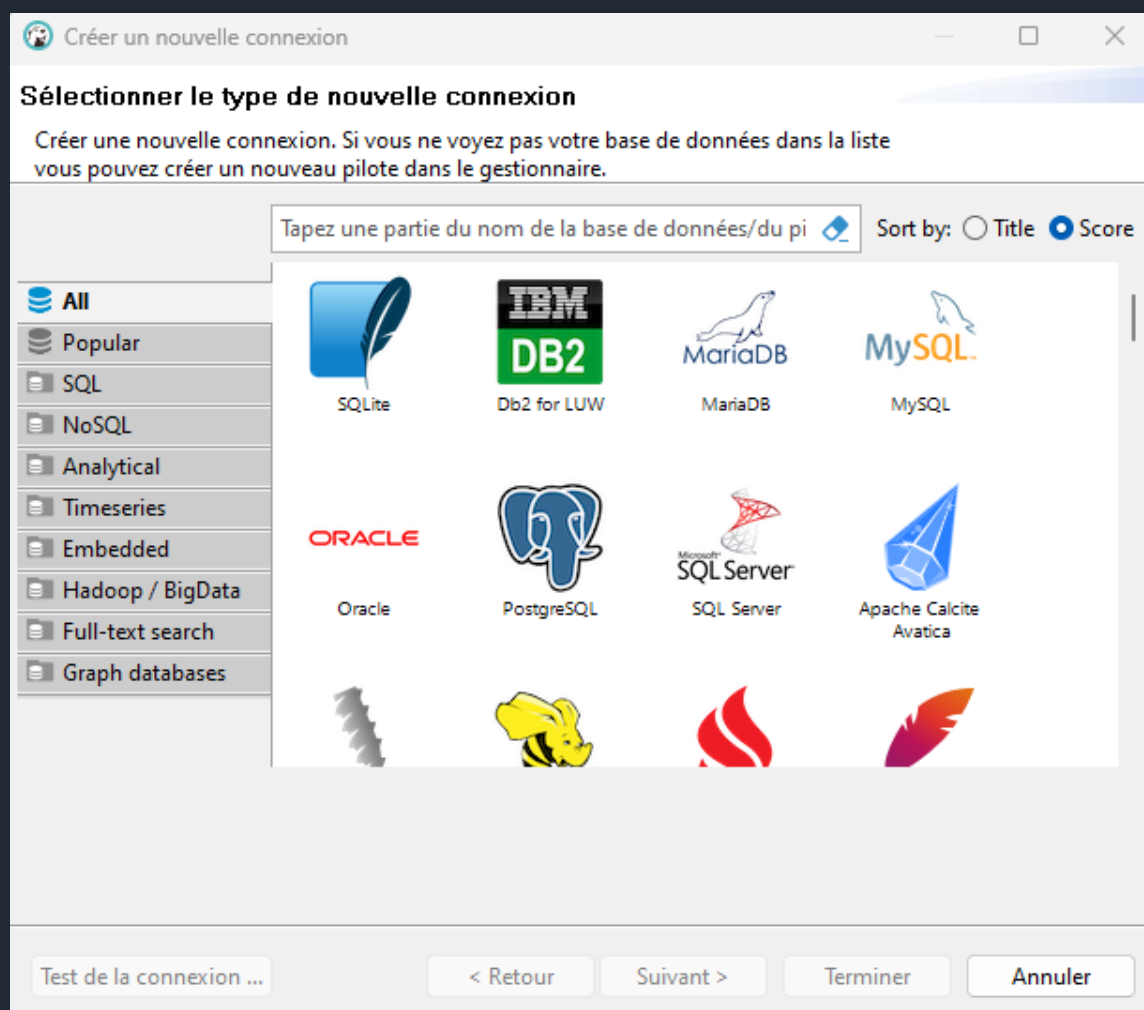
03 / 01 / 2023

Version : 1

Page : 5 / 20

1 – Première étape : Création d'une nouvelle base de données

Le logiciel DBeaver va alors vous proposer différents choix de connexion pour vos bases de données.





SQL

Mise en place de bases de données.

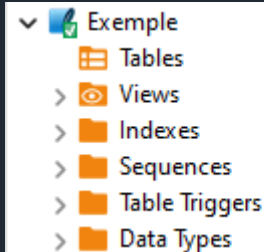
03 / 01 / 2023

Version : 1

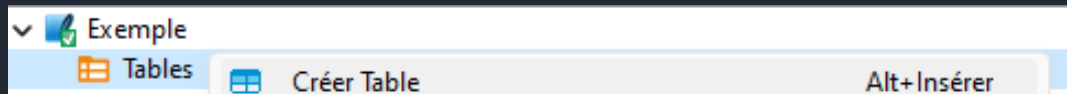
Page : 6 / 20

2 – Deuxième étape : Création d'une table

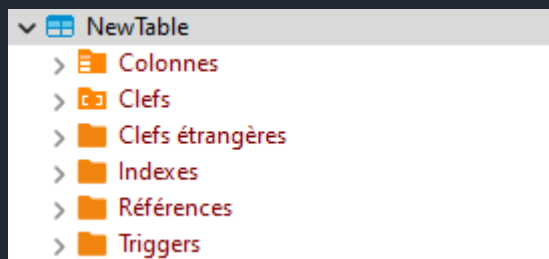
Une fois que vous avez sélectionné le type de connexion, vous pouvez alors créer une table de données. Une fois votre table de données créée, DBeaver générera six fichiers : tables, vues, index, séquences, déclencheurs de table et types de données.



Dans cet exemple, nous avons créé une base de données nommée « exemple ». Pour créer des tables, il vous suffit de faire un clic droit sur le fichier « tables », puis de sélectionner l'option « Créer Table ».



Une fois la table créée, vous pouvez la renommer (par défaut, son nom est "NewTable"). Avec cette création de table, DBeaver propose également la création de six fichiers : Colonnes Clés, Clés étrangères, Indexes, Références, Triggers.



Nom de la table:	NewTable
Table Description:	



SQL

Mise en place de bases de données.

03 / 01 / 2023

Version : 1

Page : 7 / 20

3 – Troisième étape : Remplir une table de données

Une fois votre table créée, il est nécessaire de remplir vos données. Pour ce faire, commencez par créer des colonnes. Vous pouvez le faire en cliquant sur l'icône d'édition de colonne :

Une fois que vous avez cliqué sur l'édition de colonne, DBeaver affichera une fenêtre dans laquelle vous pourrez modifier le nom de la colonne ainsi que ses différentes propriétés.

Nom	Valeur
Type de donnée	INTEGER
Length	
Non Null	[]
Auto Increment	[]
Défaut	

OK Annuler

Une fois que vous avez créé vos colonnes, vous pouvez alors choisir le type de données qui sera stocké dans chaque colonne.

Type de donnée

INTEGER

BLOB

INTEGER

NUMERIC

REAL

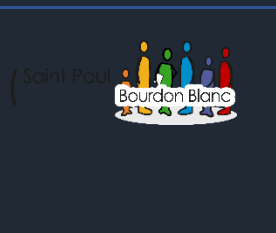
TEXT

Enfin, pour pouvoir créer des clés étrangères et des clés primaires, il vous suffira de faire un clic droit pour cliquer sur « créer une clé » puis de sélectionner les différentes clés entre les tables.

Table: Clients

Référer la table:

Clients	
Films	
Locations	



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 8 / 20

4 – Exercice sur colibri : Partie 1

Base de données n°1 :

Flims

123 codefilm	123 nomfilm
1	C'est arrivé près de chez vous
2	Bernie
3	Dans la peau de John Malkovitch
4	Intouchables
5	Ong Bak
6	Shoot' Em UP
7	Tigres et dragons
8	Matrix 1
9	Machete
10	Boulevard de la mort
11	Brain dead

Locations

123 codecli	123 codefilm	123 datedebut	123 duree
1	2	2013-04-11	1
1	4	2013-04-12	3
1	5	2013-04-13	3
2	1	2013-04-09	2
3	2	2013-04-15	5
4	1	2013-04-17	1
4	6	2013-04-21	2
5	2	2013-04-25	3
6	8	2013-05-01	2
7	7	2013-04-09	1
7	9	2012-12-31	4

Client

123 codecli	ABC prenomcli	ABC nomcli	123 ruecli	123 cpcli	ABC villecli
1	Alberto	Dubois	3 Rue du Pont	57 500	Saint-Avold
2	Mi	Volond	4 rue de la liberté	57 500	Saint-Avold
3	Roger	Botas	5 place du marché	57 500	Saint-Avold
4	Edouard	Noulas	41 rue de l'église	57 600	Forbach
5	Paul	Lontague	21 Boulevard des oiseaux	57 800	Freyming
6	Eric	Pondier	14, rue des Agates	57 600	Forbach
7	Thomas	Malon	12, rue des lapins	57 600	Forbach
8	Rénato	Point	451, rue de légalité	57 500	Saint-Avold
9	Michel	Botas	17, rue des hochets	57 500	Saint-Avold
10	David	Collague	14, rue Utrillo	57 600	Forbach
11	Simon	Potillon	17, rue des marguerittes	57 800	Freyming



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 9 / 20

4 – Exercice sur colibri : Partie 1

Partie 1 - chapitre 1 Avec la base de données n°1 :

Afficher les noms des villes où résident des clients sans prendre en compte les doublons :

```
SELECT villecli  
FROM Clients ;
```

Afficher le nom des clients :

```
SELECT nomcli  
FROM Clients ;
```

Afficher tous les films :

```
SELECT *  
FROM Films ;
```

La liste des codes des films loués sans doublon :

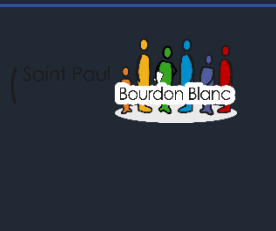
```
SELECT *  
FROM Films ;
```

Afficher sans doublon, les codes des films loués dans une colonne portant le titre "Le code des films loués" :

```
SELECT DISTINCT codefilm AS « Le code des films louées »  
FROM Locations;
```

Afficher le code des films loués et la durée de location dans des colonnes qui portent le titre "Le code des films loués" et "Durée de la location" :

```
SELECT DISTINCT codefilm AS « Le code des films louées », duree AS « Durée »  
FROM Locations;
```



4 – Exercice sur colibri : Partie 1

Base de donnée n°2

123 codecde	123 codecli	123 datecde
1	1	2009-05-04
2	2	2009-05-05
3	1	2009-05-06
4	4	2009-05-07
5	6	2009-05-08
6	7	2009-05-09
7	10	2009-05-10
8	14	2009-05-11
9	5	2009-05-12
10	6	2009-05-13
11	4	2009-05-14
12	3	2009-05-14
13	10	2009-05-16
14	12	2009-05-17
15	14	2009-05-18
16	2	2009-05-19
17	1	2009-05-20
18	9	2009-05-21
19	7	2009-05-22
20	10	2009-05-23

123 codepro	ABC designationpro	123 prixpro
1	Vélo SS1	149
2	Vélo SS2	159
3	Banc Musculation TP1	210
4	Banc Musculation TP 2	230
5	Kimono Judo S	25
6	Kimono Judo M	25
7	Kimono Judo L	25
8	Kimono Judo X	25
9	Barre traction	12
10	Rameur FD1	141
11	Rameur FD2	169
12	Short Boxe Thaie	10
13	Gant Thaie	24
14	Gant Boxe anglaise	26
15	Gant Boxe Kick Boxing	26

Commande

Produit



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 11 / 20

4 – Exercice sur colibri : Partie 1

123 codecli	ABC nomcli	123 ruecli	ABC villecli	123 cpcli	ABC payscli	123 codecde	123 codepro	123 qtecde
1	Luffof SA	Olebe Str. 48	Berlin	12 210	Allemagne	1	2	36
2	Tropulo Zappoto	Av de la conception, 45	Mexico	54 110	Mexique	1	7	45
3	Poquilo Antonio	Caliente 147	Mexico	54 110	Mexique	1	10	21
4	Maxel John	Liberty Str 27	London	WX1 1DP	Royaume-Unis	2	4	45
5	Oburgen Kolof	Libuerte 4	Stockholm	S-475 22	Suède	2	11	32
6	Kaliof Patila	Piotr 24	Berlin	12 210	Allemagne	2	15	67
7	InterSport	14, rue de la liberté	Paris	75 010	France	3	7	14
8	Ounepetipeu	Poulopo 37	Madrid	28 023	Espagne	3	10	24
9	Gogosport	24, place du pont	Lyon	69 007	France	3	14	53
10	Chun-li	34, Fighter Str	Bejin	14 544	Chine	4	1	12
11	RyuKen	16, Place du chêne	Bejin	14 544	Chine	4	2	14
12	BS Sport	Picadely circus, 34	London	WX1 1DP	Royaume Unis	5	10	14
13	Dc Wong	47, rue Powo	Bejin	14 544	Chine	6	1	25
14	Power Sport	63, rue Honda	Shangai	12 444	Chine	6	2	46
15	Athlétique Sport 1	145, rue du près	Paris	75 012	France	6	4	14
16	Lilut gym	14, rue piccolo	Bejin	13 444	Chine	6	5	32
17	Judoka Center	56 rue Sankukai	Tokyo	4 789	Japon	6	10	57
18	Tao PaiPai Box	14, rue d'ongbak	Bankkok	26 547	Thailand	7	10	47
19	Krilin Dojo	39, place du pégase	Paris	75 010	France	7	11	42
20	Pogoland	41 donlon Str	London	WX1 DP	Royaume Unis	8	8	20
						9	10	36
						10	1	45
						11	2	21
						12	3	45
						12	4	32
						12	5	67
						13	7	14
						13	8	24
						14	10	53
						14	11	12
						14	12	14
						15	14	14
						15	15	25
						16	1	46
						17	2	14
						18	3	32
						18	4	57
						19	5	47
						20	6	42
						20	7	20

Client

Ligne_commande



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 12 / 20

4 – Exercice sur colibri : Partie 1

Partie 1 - chapitre 2 Avec la base de données n°2 :

Afficher les informations sur les produits dans l'ordre croissant des prix :

```
SELECT * FROM Produit  
ORDER By prixpro ASC ;
```

Afficher toutes les informations sur les produits dans l'ordre décroissant des prix. Attention cet exemple utilise une nouvelle base de données. Pensez à observer le schéma relationnel :

```
SELECT * FROM Produit  
ORDER By prixpro DESC ;
```

Partie 1 - chapitre 2 Avec la base de données n°1 :

Afficher toutes les informations sur les clients dans l'ordre alphabétique des villes.

```
SELECT * FROM Clients  
ORDER By villecli ;
```

Afficher le nom des films dans l'ordre alphabétique inverse :

```
SELECT nomfilm FROM Films  
ORDER BY nomfilm DESC ;
```

Afficher les locations de la plus ancienne à la plus récente :

```
SELECT * FROM Locations  
ORDER BY datedebut;
```



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 13 / 20

4 – Exercice sur colibri : Partie 1

Afficher les informations de la table Clients dans l'ordre alphabétique des noms puis pour les doublons dans l'ordre inverse des prénoms :

```
SELECT * FROM Clients
ORDER BY nomcli ASC, prenomcli DESC ;
```

Partie 1 - chapitre 3 Avec la base de données n°1 :

Afficher pour chaque film loué, le nom du film et la date de location. Le résultat devra être trié dans l'ordre alphabétique des films

```
SELECT Films.nomfilm, Locations.datedebut FROM Films
JOIN Locations ON Films.codeFilm = Locations.codefilm
ORDER BY Films.nomfilm ;
```

User Afficher sans doublon le nom des villes dans lesquelles des clients ont loué des films. Le résultat sera trié dans l'ordre alphabétique des villes

```
SELECT DISTINCT Clients.villecli
FROM Clients
JOIN Locations ON Clients.codecli = Locations.codecli
ORDER BY Clients.villecli;
```

Afficher les numéros et les villes des clients en indiquant pour ceux qui ont loué des films, le numéro du film emprunté. Le résultat sera affiché dans l'ordre alphabétique des villes.

```
SELECT DISTINCT Clients.codecli, Clients.villecli, Locations.codefilm
FROM Clients
LEFT JOIN Locations ON Clients.codecli = Locations.codecli
ORDER BY Clients.villecli ;
```



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 14 / 20

4 – Exercice sur colibri : Partie 1

Pour chaque client qui a loué un film, le nom du client, le nom du film loué, et la date de début de la location. Le résultat sera affiché dans l'ordre alphabétique des noms des clients

```
SELECT Clients.nomcli, Films.nomfilm, Locations.datedebut  
FROM Clients  
JOIN Locations ON Clients.codecli = Locations.codecli  
JOIN Films ON Locations.codefilm = Films.codefilm  
ORDER BY Clients.nomcli ;
```

Le nom de tous les clients et pour ceux qui ont loué un film, le nom du film loué et la durée de location. Le résultat sera affiché dans l'ordre alphabétique des noms des clients.

```
SELECT Clients.nomcli, Films.nomfilm, Locations.duree  
FROM Clients  
JOIN Locations ON Clients.codecli = Locations.codecli  
JOIN Films ON Locations.codefilm = Films.codefilm  
ORDER BY Clients.nomcli ;
```




SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 15 / 20

4 – Exercice sur colibri : Partie 1

Partie 1 - chapitre 4 Avec la base de données n°1 :

Afficher le nom du client n°1 dans une colonne qui portera comme titre "Nom du client n°1" :

```
SELECT nomcli AS 'Nom du client n °1 '  
FROM Clients WHERE codecli = 1 ;
```

Afficher le nom des clients qui commencent par un 'c' classé dans l'ordre alphabétique :

```
SELECT nomcli  
FROM Clients  
WHERE nomcli LIKE 'C%'  
ORDER BY nomcli ;
```

Afficher les codes des films, leurs noms et la date de début de location des films loués entre le 09/04/2013 et le 11/04/2013. Le résultat sera trié dans l'ordre inverse des dates de début de location :

```
SELECT flims.codefilm, films.nomfilm, locations.datedebut  
FROM films  
INNER JOIN Locations ON films.codefilm = locations.codefilm  
WHERE locations.datedebut BETWEEN '2013-04-09' AND '2013-04-11'  
ORDER BY locations.datedebut DESC ;
```

Afficher le nom des clients qui commencent par un 'c' ou un 'd' dans l'ordre alphabétique :

```
SELECT Clients.nomcli  
FROM Clients  
WHERE nomcli LIKE 'C%' OR nomcli LIKE 'D%'  
ORDER BY nomcli ;
```



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 16 / 20

4 – Exercice sur colibri : Partie 1

Afficher le nom et la ville des clients qui habitent Forbach ou Saint-Avold dans l'ordre alphabétique des noms des clients.

```
SELECT Clients.nomcli, Clients.villecli FROM Clients
WHERE Clients.villecli LIKE 'Forbach' OR Clients.villecli LIKE 'Saint-Avold'
ORDER BY Clients.nomcli ;
```

Afficher les noms et codes postaux des clients qui habitent en Moselle (département 57) dans l'ordre alphabétique des noms des clients.

```
SELECT nomcli, cplci
FROM clients
WHERE cplci LIKE '57%'
ORDER BY nomcli ASC ;
```

Partie 1 - chapitre 5 Avec la base de données n°1 :

Le nombre de films empruntés par le client n°1 dans une colonne qui portera comme titre "Nombre de films"

```
SELECT COUNT(*) AS « Nombre de films »
FROM Locations
WHERE codecli = 1 ;
```

Le nombre de films empruntés par le client n°1 dans une colonne qui portera comme titre "Nombre de films"

```
SELECT SUM(duree) AS « Nombre de jours de location client 1 »
FROM Locations
WHERE codecli = 1 ;
```



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 17 / 20

4 – Exercice sur colibri : Partie 1

La date de location la plus ancienne dans une colonne qui portera le titre "Date la plus ancienne".

```
SELECT MIN(datedebut) AS «Date la plus ancienne »  
FROM Locations  
WHERE codecli ;
```

Le chiffre d'affaires TTC réalisé avec le film n°2 en sachant que chaque jour de location est facturé 1.97 € HT et que le taux de TVA est de 20%. Le résultat sera arrondi au centime d'euro le plus proche et la colonne aura pour titre "TTC".

```
SELECT ROUND(SUM(duree * 1.97 * 1.20), 2) AS « Moyenne »  
FROM Locations  
WHERE codecli = 1 ;
```

La moyenne du CA TTC du client n°1 en sachant que chaque jour de location est facturé 1.97 € HT et que le taux de TVA est de 20%. Le résultat sera arrondi au centime d'euro le plus proche et la colonne aura pour titre "Moyenne".

```
SELECT ROUND(AVG(duree * 1.97 * 1.20), 2) AS « Moyenne »  
FROM Locations  
WHERE codecli = 1 ;
```



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 18 / 20

4 – Exercice sur colibri : Partie 1

Partie 1 - chapitre 6 Avec la base de données n°1 :

Le nombre de clients dans chaque ville dans une colonne qui s'appelle "Nombre de clients par ville"

```
SELECT COUNT(*) AS « Nombre de clients par ville »  
FROM Clients  
WHERE villecli ;
```

Le nom de chaque film avec la durée de location la plus longue. Le résultat sera affiché dans l'ordre alphabétique des noms des films.

```
SELECT Films.nomfilm, MAX(Locations.duree) AS « MAX(duree) »  
FROM Films JOIN Locations ON Films.codefilm = Locations.codefilm  
GROUP BY Films.nomfilm  
ORDER BY Films.nomfilm ;
```

Le nom de chaque film avec la durée de location la plus longue. Le résultat sera affiché dans l'ordre alphabétique des noms des films.

```
SELECT Films.nomfilm, ROUND(AVG(Locations.duree), 1) AS "Moyenne"  
FROM Films  
JOIN Locations ON Films.codefilm = Locations.codefilm  
GROUP BY Films.nomfilm  
ORDER BY Films.nomfilm ;
```



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 19 / 20

4 – Exercice sur colibri : Partie 1

Partie 1 - chapitre 7 Avec la base de données n°1 :

Les noms des clients dont le total des jours de location est supérieur à 4. Le résultat sera classé dans l'ordre alphabétique des noms des clients.

```
SELECT nomcli, SUM(locations.duree) AS « durée total » FROM Clients
JOIN Clients
JOIN Locations ON Clients.codecli = Location.codecli
GROUP BY nomcli HAVING SUM(Location.duree) > 4
ORDER BY nomcli ;
```

Afficher les codes les noms des films loués moins de 3 fois, ainsi que le nombre de location dans une colonne nommée "Nombre locations". Le résultat sera classé dans l'ordre croissant des codes de film.

```
SELECT Films.codefilm, Films.nomfil, ROUND(AVG(Locations.duree) , 1) AS « AVG(duree) »
FROM Films
JOIN Locations ON Films.codefilm = Locations.codefilm
GROUP BY Films.codefilm, Film.nomfilm
HAVING COUNT(Locations.codefilm) < 3
ORDER BY Films.codefilm ;
```

Afficher les codes les noms des films loués moins de 3 fois, ainsi que le nombre de location dans une colonne nommée "Nombre locations". Le résultat sera classé dans l'ordre croissant des codes de film

```
SELECT Films.codefilm, Films.nomfil, ROUND(AVG(Locations.duree) , 1) AS « AVG(duree) »
FROM Films
JOIN Locations ON Films.codefilm = Locations.codefilm
GROUP BY Films.codefilm, Film.nomfilm
HAVING AVG(Locations.codefilm) < 2
ORDER BY « Durée moyenne de location » DESC;
```



SQL

Exercice colibri

03 / 01 / 2023

Version : 1

Page : 20 / 20

5 – Exercice sur colibri : Partie 2

Partie 2 - chapitre 1 Avec la base de données n°1 :

Saisissez la requête qui permettra d'insérer le film n°12 "The Raid"

```
INSERT INTO Films(codefilm, nomfilm)
VALUES(12, « The Raid ») ;
```

Saisissez la requête qui permettra d'insérer le film n°12 "The Raid" et le films n°13 "Le loup de Wall Street".

```
INSERT INTO Films(codefilm, nomfilm)
VALUES(12, « The Raid ») , (13, « Le loup de Wall Street ») ;
```

Saisissez la requête qui permettra d'insérer le client n°124 qui s'appelle "Jean" "Talu" (les autres informations sur ce client ne sont pas connues).

```
INSERT INTO Clients(codecli,prenomcli,nomcli)
VALUES(124, 'Jean', 'Talu') ;
```

Supprimez toutes les lignes de la table clients.

```
DELETE FROM Locations ;
DELETE FROM Clients ;
```

Éditée par	Tom COELHO	
Révisée par :	Tom COELHO	
Suivie par :	Tom COELHO	
Validée par :	Tom COELHO	
Date : 07 / 01 / 2023	 (Saint Paul	Version : 1