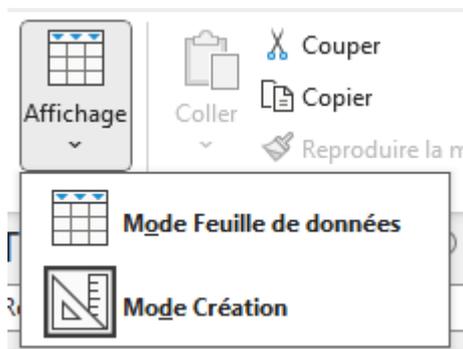


0 INTRODUCTION

➔ Téléchargez et ouvrez dans Microsoft Access la base de données fournie (music.accdb).

Dans l'onglet de Navigation sur la gauche de votre écran, vous avez **une liste de toutes les tables présentes** dans la base de données. Vous pouvez cliquer deux fois sur une table pour l'ouvrir et inspecter ses données et son schéma. Vous basculez entre la visualisation des données et la visualisation du schéma en cliquant sur « Affichage » dans l'onglet « Accueil ».

- Mode Feuille de données : affichage des données
- Mode Création : affichage du schéma



Tables	
ALBUM	
ALBUM_TYPE	
ARTISTE	
CONTINENT	
GENRE	
LABEL	
LIEU	
LIEU_CATEGORIE	
LIEU_TYPE	
PAYS	
PISTE	
SHOW	
SOCIETE	
SOUSGENRE	
TOURNEE	
VILLE	

Attention à ne pas modifier les données !

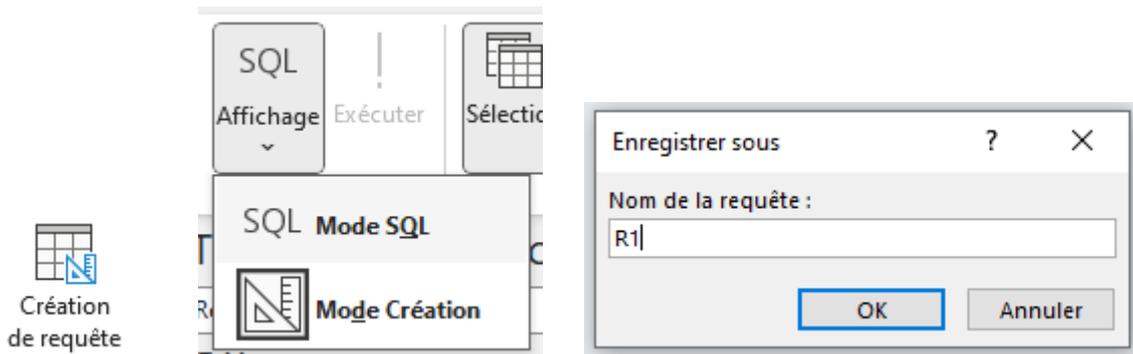
La base de données fournie contient des données sur plusieurs artistes, leurs albums et leurs tournées. Voici le **modèle relationnel** de la base :

ARTISTE (ArtisteID, NomArtiste, TypeArtiste, DebutCarriere, FinCarriere)
ALBUM (AlbumID, Titre, #ArtisteID, DateSortie, DureeMinutes, DureeSecondes, Tracks, #SousGenreID, #LabelID, #TypeID, Top200MeilleurePos, Top200FinAnnee, Ventes, WikiURL, Pochette)
TOURNEE (TourneeID, #ArtisteID, NomTournee, DateDebut, DateFin, Etapes, Shows, Spectateurs, Recettes, #AlbumID)
ALBUM_TYPE (TypeID, Type)
SOUSGENRE (SousGenreID, NomSousGenre, #GenreID)
GENRE (GenreID, NomGenre)
PISTE (PisteID, #AlbumID, NumPiste, NomPiste, DureeMinutes, DureeSecondes, DateSortieSingle, Top100MeilleurePos, Top100FinAnnee)
LABEL (LabelID, NomLabel, #SocieteID, LogoURL)
SOCIETE (SocieteID, NomSociete, LogoURL)
SHOW (ShowID, #TourneeID, Etape, DateShow, #LieuID, BilletsVendus, BilletsDisponibles, Recettes, Annule, MotifAnnulation)
LIEU (LieuID, NomLieu, #LieuTypeID, Capacite, ConstructionDebut, DatePremierShow,

DateDernierShow, Demolition, CoutConstruction, Adresse, #VilleID, Latitude, Longitude, AutresNoms)
 VILLE (VilleID, NomVille, #PaysID)
 PAYS (PaysID, NomPays, #ContinentID, DrapeauURL)
 CONTINENT (ContinentID, NomContinent)
 LIEU_TYPE (LieuTypeID, NomType, #LieuCategorieID)
 LIEU_CATEGORIE (LieuCategorieID, NomCategorie)

Pour entrer une requête SQL, cliquez sur « Création de requête » dans l'onglet « Créer ». Ensuite, cliquez sur « Affichage » et sélectionnez « Mode SQL ».

Dans la fenêtre qui apparaît, nommez la nouvelle requête. Exemple de nom : R1.



Dans l'éditeur de requête, supprimez le « SELECT » qui s'affiche automatiquement et écrivez votre code SQL. Il convient de commencer par la clause FROM, puis compléter la clause SELECT au-dessus.

Lancez votre requête en cliquant sur « Exécuter ».

Exécuter

1 LISTE D'ARTISTES

Ecrivez une requête SQL, nommée R1, pour afficher les colonnes de la table ARTISTE indiquées dans l'image ci-dessous :

NomArtiste	TypeArtiste	DebutCarriere
AC/DC	Band	1973
Bad Bunny	Solo Artist	2013
Ron Jovi	Band	1983

```
SELECT A.NomArtiste, A.TypeArtiste, A.DebutCarriere
FROM ARTISTE A
```

Modifiez votre requête pour renommer les colonnes comme dans l'image ci-dessous : (Utilisez le mot clé AS)

Nom d'artiste	Type d'artiste	Début de carrière
AC/DC	Band	1973
Bad Bunny	Solo Artist	2013

```
SELECT A.NomArtiste AS [Nom d'artiste], A.TypeArtiste AS [Type d'artiste],
A.DebutCarriere AS [Début de carrière]
FROM ARTISTE A
```

Ajoutez une clause ORDER BY pour trier la liste par ordre décroissant d'année de début de carrière :

Nom d'artiste	Type d'artiste	Début de carrière
Bad Bunny	Solo Artist	2013
Harry Styles	Solo Artist	2010
One Direction	Band	2010
The Weeknd	Solo Artist	2009

```
SELECT A.NomArtiste AS [Nom d'artiste], A.TypeArtiste AS [Type d'artiste],
       A.DebutCarriere AS [Début de carrière]
FROM ARTISTE A
ORDER BY A.DebutCarriere DESC
```

Complétez la clause ORDER BY pour faire en sorte que, si plusieurs artistes ont le même début de carrière, alors ils apparaissent en ordre alphabétique (A-Z) :

Nom d'artiste	Type d'artiste	Début de carrière
Bad Bunny	Solo Artist	2013
Harry Styles	Solo Artist	2010
One Direction	Band	2010
The Weeknd	Solo Artist	2009
Bruno Mars	Solo Artist	2004
Ed Sheeran	Solo Artist	2004
Taylor Swift	Solo Artist	2004

```
SELECT A.NomArtiste AS [Nom d'artiste], A.TypeArtiste AS [Type d'artiste],
       A.DebutCarriere AS [Début de carrière]
FROM ARTISTE A
ORDER BY A.DebutCarriere DESC, A.NomArtiste
```

Enregistrez votre requête sous le nom R1.

2 LISTES D'ALBUMS AVEC RESTRICTIONS DE LIGNES

Dans toutes les questions ci-dessous, affichez les colonnes Titre, Top200MeilleurePos, Top200FinAnnee, et Ventes de la table ALBUM et triez le résultat par ordre alphabétique de Titre.

- a) Ecrivez une nouvelle requête SQL, nommée R2a, pour afficher les Albums qui ont atteint la position 1 dans le Top200 (attribut Top200MeilleurePos).

Titre	Top200MeilleurePos	Top200FinAnnee	Ventes
...Baby One More Time	1	2	14
=	1	15	1
1989	1	1	6,25
1989 (Taylor's Version)	1		

```
SELECT A.Titre, A.Top200MeilleurePos, A.Top200FinAnnee, A.Ventes
FROM ALBUM A
WHERE A.Top200MeilleurePos = 1
ORDER BY A.Titre
```

- b) Ecrivez une nouvelle requête SQL, nommée R2b, pour afficher les Albums qui ont fini l'année en position 1 dans le Top200 (attribut Top200FinAnnee).

```
SELECT A.Titre, A.Top200MeilleurePos, A.Top200FinAnnee, A.Ventes
FROM ALBUM A
WHERE A.Top200FinAnnee = 1
ORDER BY A.Titre
```

- c) Ecrivez une nouvelle requête SQL, nommée R2c, pour afficher les Albums qui sont entrés dans le Top10. *Vous devrez tester si la meilleure position est ≤ 10 .*

```
SELECT A.Titre, A.Top200MeilleurePos, A.Top200FinAnnee, A.Ventes
FROM ALBUM A
WHERE A.Top200MeilleurePos <= 10
ORDER BY A.Titre
```

- d) Ecrivez une nouvelle requête SQL, nommée R2d, pour afficher les Albums qui ont vendu au moins 10 millions de copies. *La colonne Ventes donne le nombre de copies vendues en millions.*

```
SELECT A.Titre, A.Top200MeilleurePos, A.Top200FinAnnee, A.Ventes
FROM ALBUM A
WHERE A.Ventes >= 10
ORDER BY A.Titre
```

- e) Ecrivez une nouvelle requête SQL, nommée R2e, pour afficher les Albums qui ont atteint la position 1 dans le Top200 **et** qui ont vendu au moins 10 millions de copies.

```
SELECT A.Titre, A.Top200MeilleurePos, A.Top200FinAnnee, A.Ventes
FROM ALBUM A
WHERE A.Top200MeilleurePos = 1 AND A.Ventes >= 10
ORDER BY A.Titre
```

- f) Ecrivez une nouvelle requête SQL, nommée R2f, pour afficher les albums qui ont vendu au moins 2 millions de copies, mais qui n'ont pas atteint le Top10.

```
SELECT A.Titre, A.Top200MeilleurePos, A.Top200FinAnnee, A.Ventes
FROM ALBUM A
WHERE A.Top200MeilleurePos > 10 AND A.Ventes >= 2
ORDER BY A.Titre
```

3 JOINTURES

- a) Ecrivez une nouvelle requête SQL, nommée R3a, pour afficher le Titre, les Ventes et le nom de Sous-genre de tous les albums ayant vendu au moins 0,5 million de copies. *Vous devrez faire une jointure entre les tables ALBUM et SOUSGENRE.*

Titre	Ventes	NomSousGenre
Black Ice	2	Hard rock
Lost Highway	1	Country rock
Have A Nice Day	1	Pop rock
Circus	3	Pop
Born in the U.S.A.	17	Rock and roll
Magic	1	Rock
The Risina	2.1	Rock

```
SELECT A.Titre, A.Ventes, SG.NomSousGenre
FROM ALBUM A INNER JOIN SOUSGENRE SG ON A.SousGenreID=SG.SousGenreID
WHERE A.Ventes>=0.5
```

- b) Ecrivez une nouvelle requête SQL, nommée R3b, pour afficher le Titre, les Ventes et le Nom de Sous-genre de tous les albums ayant vendu au moins 0,5 million de copies et dont le Genre est 'Metal'. Triez les résultats par ordre décroissant de Ventes.

Vous devrez faire une jointure entre les tables ALBUM, SOUSGENRE et GENRE (jointure à 3 tables).

Titre	Ventes	NomSousGenre
Metallica	17,3	Heavy metal
...And Justice for All	9,7	Thrash metal
Master of Puppets	7,98	Thrash metal
Ride the Lightning	6,95	Thrash metal
Load	5,4	Heavy metal

```
SELECT A.Titre, A.Ventes, SG.NomSousGenre
FROM (ALBUM A INNER JOIN SOUSGENRE SG ON A.SousGenreID=SG.SousGenreID)
INNER JOIN GENRE G ON SG.GenreID=G.GenreID
WHERE A.Ventes>=0.5 And G.NomGenre='Metal'
ORDER BY A.Ventes DESC
```

4 COLONNES CALCULEES

Ecrivez une nouvelle requête SQL, nommée R4, pour afficher les colonnes DateShow, BilletsDisponibles, BilletsVendus, Recettes de la table SHOW.

```
SELECT S.DateShow, S.BilletsDisponibles, S.BilletsVendus, S.Recettes
FROM SHOW S
```

Dans la même requête, ajoutez une colonne calculée, nommée 'Billets Invendus', qui soustrait les billets vendus aux billets disponibles. *Utilisez le mot clé AS pour déclarer le nom de la colonne.*

```
SELECT S.DateShow, S.BilletsDisponibles, S.BilletsVendus,
S.BilletsDisponibles-S.BilletsVendus AS 'Billets Invendus', S.Recettes
FROM SHOW S
```

Dans la même requête, ajoutez une clause WHERE pour afficher uniquement les Shows avec au moins 10000 billets invendus. *Vous ne pourrez pas utiliser le nom de votre colonne calculée dans la clause WHERE, vous devrez réécrire l'expression mathématique.*

```
SELECT S.DateShow, S.BilletsDisponibles, S.BilletsVendus,
S.BilletsDisponibles-S.BilletsVendus AS 'Billets Invendus', S.Recettes
FROM SHOW S
WHERE S.BilletsDisponibles-S.BilletsVendus >= 10000
```

Dans la même requête, triez par ordre décroissant de billets invendus. *Vous ne pourrez pas utiliser le nom de votre colonne calculée dans la clause ORDER BY.*

DateShow	BilletsDispc	BilletsVendi	'Billets Invendus'	Recettes
31/07/2013	77000	25807	51193	1453804
12/11/1997	50000	14491	35509	1282160
19/05/1997	55000	23709	31291	1201035
10/11/1997	50000	18751	31249	893865
06/08/1994	50000	19893	30107	

```
SELECT S.DateShow, S.BilletsDisponibles, S.BilletsVendus,
       S.BilletsDisponibles-S.BilletsVendus AS 'Billets Invendus', S.Recettes
FROM SHOW S
WHERE S.BilletsDisponibles-S.BilletsVendus >= 10000
ORDER BY S.BilletsDisponibles-S.BilletsVendus DESC
```

Dans la même requête, ajoutez une colonne nommée 'Prix moyen billet' pour calculer le prix moyen des billets vendus. *Vous devrez diviser les recettes par le nombre de billets vendus.*

```
SELECT S.DateShow, S.BilletsDisponibles, S.BilletsVendus,
       S.BilletsDisponibles-S.BilletsVendus AS 'Billets Invendus', S.Recettes,
       S.Recettes/S.BilletsVendus AS 'Prix moyen billet'
FROM SHOW S
WHERE S.BilletsDisponibles-S.BilletsVendus >= 10000
ORDER BY S.BilletsDisponibles-S.BilletsVendus DESC
```

Dans la même requête, ajoutez une colonne nommée 'Perte estimée' pour calculer la perte à cause des billets invendus. Vous devrez multiplier le nombre de billets invendus par le prix moyen, mais vous ne pourrez pas utiliser les alias des colonnes pour le calcul. Il faudra écrire l'expression mathématique complète, en faisant attention aux parenthèses.

DateShow	BilletsDisponibles	BilletsVendus	'Billets Invendus'	Recettes	'Prix moyen billet'	'Perte estimée'
31/07/2013	77000	25807	51193	1453804	56,3337079087069	2883891,50897043
12/11/1997	50000	14491	35509	1282160	88,479746049272	3141827,3024636
19/05/1997	55000	23709	31291	1201035	50,6573453119069	1585118,99215488
10/11/1997	50000	18751	31249	893865	47,6702575862621	1489647,8793131
06/08/1994	50000	19893	30107			
28/05/2009	73997	45000	28997	5100750	113,35	3286809,95
08/11/1997	50000	24807	25193	1282160	51.6854113758213	1302110.56879107

```
SELECT S.DateShow, S.BilletsDisponibles, S.BilletsVendus,
       S.BilletsDisponibles-S.BilletsVendus AS 'Billets Invendus', S.Recettes,
       S.Recettes/S.BilletsVendus AS 'Prix moyen billet', (S.BilletsDisponibles-
       S.BilletsVendus)*(S.Recettes/S.BilletsVendus) AS 'Perte estimée'
FROM SHOW S
WHERE S.BilletsDisponibles-S.BilletsVendus >= 10000
ORDER BY S.BilletsDisponibles-S.BilletsVendus DESC
```

5 FONCTIONS D'AGREGATION AVEC GROUPES

Ecrivez une nouvelle requête SQL, nommée R5, qui fait une jointure entre ALBUM et ARTISTE en affichant le NomArtiste.

NomArtiste
AC/DC

Le Nom d'artiste sera affiché plusieurs fois : c'est normal.

```
SELECT ART.NomArtiste
FROM ALBUM ALB INNER JOIN ARTISTE ART ON ALB.ArtisteID=ART.ArtisteID
```

Dans la même requête, ajoutez une clause GROUP BY pour regrouper par NomArtiste.

```
SELECT ART.NomArtiste
FROM ALBUM ALB INNER JOIN ARTISTE ART ON ALB.ArtisteID=ART.ArtisteID
GROUP BY ART.NomArtiste
```

Ensuite, ajoutez des colonnes calculées dans la clause SELECT pour calculer, pour chaque artiste :

- Le nombre d'Albums
- La somme des Ventes
- La moyenne des Ventes

Triez les résultats par ordre décroissant de nombre d'Albums :

NomArtiste	'Nb Albums'	'Total Ventes'	'Moyenne Ventes'
The Rolling St	55	55,1399999856949	1,41384615347936
Elton John	43	78	2,36363636363636
David Bowie	35	11.7800000086427	0.736250000540167

```
SELECT ART.NomArtiste, COUNT(ALB.AlbumID) AS 'Nb Albums', SUM(ALB.Ventes) AS
'Total Ventes', AVG(ALB.Ventes) AS 'Moyenne Ventes'
FROM ALBUM ALB INNER JOIN ARTISTE ART ON ALB.ArtisteID=ART.ArtisteID
GROUP BY ART.NomArtiste
ORDER BY COUNT(ALB.AlbumId) DESC
```

Dans la même requête, ajoutez une clause WHERE pour prendre en compte uniquement les Albums qui ont atteint la position 1 dans le Top200.

NomArtiste	'Nb Albums'	'Total Ventes'	'Moyenne Ventes'
Taylor Swift	13	33,6040000915527	3,7337777879503
Bruce Springsteen	12	39,027999818325	3,54799998348409
Kenny Chesney	11	18,5	1,85
Madonna	9	28,9039998352528	3,21155553725031
The Rolling Stones	9	22	2,44444444444444
Elton John	8	45	5.625

```
SELECT ART.NomArtiste, COUNT(ALB.AlbumID) AS 'Nb Albums', SUM(ALB.Ventes) AS
'Total Ventes', AVG(ALB.Ventes) AS 'Moyenne Ventes'
FROM ALBUM ALB INNER JOIN ARTISTE ART ON ALB.ArtisteID=ART.ArtisteID
WHERE ALB.Top200MeilleurePos = 1
GROUP BY ART.NomArtiste
ORDER BY COUNT(ALB.AlbumId) DESC
```

Finalement, ajoutez une clause HAVING pour afficher uniquement les artistes avec moyenne de Ventes supérieure ou égale à 10 millions.

NomArtiste	'Nb Albums'	'Total Ventes'	'Moyenne Ventes'
Garth Brooks	8	70,5	10,0714285714286
Eagles	6	92	15,3333333333333
Michael Jackson	5	59,4000000953674	11,8800000190735
Guns ' Roses	2	25	12,5

```
SELECT ART.NomArtiste, COUNT(ALB.AlbumID) AS 'Nb Albums', SUM(ALB.Ventes) AS
'Total Ventes', AVG(ALB.Ventes) AS 'Moyenne Ventes'
FROM ALBUM ALB INNER JOIN ARTISTE ART ON ALB.ArtisteID=ART.ArtisteID
WHERE ALB.Top200MeilleurePos = 1
GROUP BY ART.NomArtiste
```

```
HAVING AVG(ALB.Ventes) > 10
ORDER BY COUNT(ALB.AlbumId) DESC
```

6 LISTES D'ALBUMS (SELECT TOP)

- a) Ecrivez une nouvelle requête SQL, nommée R6a, pour afficher les colonnes Titre, DateSortie, Ventes, et WikiURL de la table ALBUM. Triez les résultats par ordre décroissant des Ventes.

```
SELECT A.Titre, A.DateSortie, A.Ventes, A.WikiURL
FROM ALBUM A
ORDER BY A.Ventes DESC
```

Modifiez la requête pour afficher uniquement les 5 albums les plus vendus (utilisez la syntaxe SELECT TOP 5 à la place de SELECT). Vous devez obtenir le résultat ci-dessous :

Titre	DateSortie	Ventes	WikiURL
Their Greatest Hits (1971â€"1975)	17/02/1976	38	https://en.wikipedi
Thriller	29/11/1982	34	https://en.wikipedi
Hotel California	08/12/1976	26	https://en.wikipedi
Back in Black	25/07/1980	25	https://en.wikipedi
No Fences	27/08/1990	18	https://en.wikipedi
Appetite for Destruction	21/07/1987	18	https://en.wikipedi

Observez que 6 albums sont affichés, car deux albums sont classés ex æquo en 5^e position.

```
SELECT TOP 5 A.Titre, A.DateSortie, A.Ventes, A.WikiURL
FROM ALBUM A
ORDER BY A.Ventes DESC
```

- b) Ecrivez une nouvelle requête SQL, nommée R6b, pour afficher les 3 albums les plus récents (date de sortie la plus récente), en incluant les ex æquo en 3^e position. Affichez les mêmes colonnes que dans la requête R6a.

```
SELECT TOP 3 A.Titre, A.DateSortie, A.Ventes, A.WikiURL
FROM ALBUM A
ORDER BY A.DateSortie DESC
```