

# CRUD

## CREATE, READ, UPDATE, DELETE

Avant la lecture de ce document, je vous invite à télécharger les fichiers en relation avec ce document :

- Dictionnaire de Données - Notation de Films
  - Modèle Conceptuel de Données
  - Modèle Physique de Données
- Requêtes pour créer la structure de la base de données
- Requêtes pour insérer des données dans l'ensemble des tables de la base de données

Les requêtes sont à utiliser dans une base de données nommée "notationFilm" dans SQL SERVER

En espérant que ce document vous aidera !

Bon courage à toutes et à tous !

## TABLE DES MATIÈRES

INSERT INTO : permet d'insérer des données dans une ou des tables dans le but entre autres de tester leurs structures.

SELECT : permet de sélectionner une ou des données d'une ou plusieurs tables dans une base de données.

UPDATE : permet de modifier une ou des données d'une table dans une base de données.

DELETE : permet de supprimer une ou des données dans une table.

AS (Alias) : permet de donner un ou des noms temporaires à une ou des colonnes, à une ou des tables.

GROUP BY : permet de regrouper les lignes qui ont les mêmes valeurs dans des lignes récapitulatives.

ORDER BY : permet de classer le résultat d'une requête dans un ordre ascendant ou descendant.

HAVING : permet, à partir de fonctions d'agréations, de filtrer et extraire les seules données répondant à une condition spécifique.

BEGIN TRANSACTION : permet de tester une requête avant de l'exécuter.

INSERT INTO : permet d'insérer des données dans une ou des tables dans le but entre autres de tester leurs structures.

Avec le nom des colonnes et leurs valeurs associées																																
Requête	Exemple																															
<p><u>Ajout d'une valeur</u></p> <pre>INSERT INTO {nomTable} (     {nomColonne1},     {nomColonne2},     ... ) VALUES (     {valeurColonne1},     {valeurColonne2},     ... );</pre>	<p>On veut ajouter dans la table "Film" les entrées suivantes :</p> <ul style="list-style-type: none"><li>• "Matrix" sorti en 2001,</li><li>• "Parasite" sorti en 2015,</li><li>• "Les Sept Samourais" sorti en 1950</li><li>• "Julieta" sorti en 2014 .</li></ul> <p>Pour se faire, on écrit les requêtes suivantes :</p> <pre>INSERT INTO film (nom_film, annee_sortie_film) VALUES ('Matrix', 2001);  INSERT INTO film (nom_film, annee_sortie_film) VALUES ('Parasite', 2015);  INSERT INTO film (nom_film, annee_sortie_film) VALUES ('Les Sept Samourais', 1950);  INSERT INTO film (nom_film, annee_sortie_film) VALUES ('Julieta', 2014);</pre>	<p>On a pour résultat :</p> <table><tr><th>id_film</th><th>nom_film</th><th>annee_sortie_film</th></tr><tr><td>30</td><td>Somewhere</td><td>2010</td></tr><tr><td>31</td><td>THX 1138</td><td>1971</td></tr><tr><td>32</td><td>American Graffiti</td><td>1973</td></tr><tr><td>33</td><td>Star Wars : Episode IV - Un nouvel espoir</td><td>1977</td></tr><tr><td>34</td><td>Star Wars : Episode I - La menace fantôme</td><td>1999</td></tr><tr><td>35</td><td>Matrix</td><td>2001</td></tr><tr><td>36</td><td>Parasite</td><td>2015</td></tr><tr><td>37</td><td>Les Sept Samourais</td><td>1950</td></tr><tr><td>38</td><td>Julieta</td><td>2014</td></tr></table>	id_film	nom_film	annee_sortie_film	30	Somewhere	2010	31	THX 1138	1971	32	American Graffiti	1973	33	Star Wars : Episode IV - Un nouvel espoir	1977	34	Star Wars : Episode I - La menace fantôme	1999	35	Matrix	2001	36	Parasite	2015	37	Les Sept Samourais	1950	38	Julieta	2014
id_film	nom_film	annee_sortie_film																														
30	Somewhere	2010																														
31	THX 1138	1971																														
32	American Graffiti	1973																														
33	Star Wars : Episode IV - Un nouvel espoir	1977																														
34	Star Wars : Episode I - La menace fantôme	1999																														
35	Matrix	2001																														
36	Parasite	2015																														
37	Les Sept Samourais	1950																														
38	Julieta	2014																														
<p><u>Ajout de plusieurs valeurs</u></p> <pre>INSERT INTO {nomTable} (     {nomColonne1},     {nomColonne2},     ... ) VALUES (     {valeurColonne1},     {valeurColonne2},     ... ), (     {valeurColonne1},     {valeurColonne2},     ... );</pre>	<p>On veut ajouter trois nouvelles nationalités dans la table "Nationalité" : à savoir la Corée du Sud, le Japon et l'Espagne. Pour se faire, on écrit la requête suivante :</p> <pre>INSERT INTO nationalite (id_nationalite, nom_nationalite) VALUES ('KOR', 'Corée du Sud'), ('JPN', 'Japon'), ('ESP', 'Espagne');</pre>	<p>On a pour résultat :</p> <table><tr><th>id_nationalite</th><th>nom_nationalite</th></tr><tr><td>CHN</td><td>Chine</td></tr><tr><td>ESP</td><td>Espagne</td></tr><tr><td>FRA</td><td>France</td></tr><tr><td>GBR</td><td>Royaume-Uni</td></tr><tr><td>ITA</td><td>Italie</td></tr><tr><td>JPN</td><td>Japon</td></tr><tr><td>KOR</td><td>Corée du Sud</td></tr><tr><td>NZL</td><td>Nouvelle-Zélande</td></tr><tr><td>USA</td><td>Etats-Unis</td></tr></table>	id_nationalite	nom_nationalite	CHN	Chine	ESP	Espagne	FRA	France	GBR	Royaume-Uni	ITA	Italie	JPN	Japon	KOR	Corée du Sud	NZL	Nouvelle-Zélande	USA	Etats-Unis										
id_nationalite	nom_nationalite																															
CHN	Chine																															
ESP	Espagne																															
FRA	France																															
GBR	Royaume-Uni																															
ITA	Italie																															
JPN	Japon																															
KOR	Corée du Sud																															
NZL	Nouvelle-Zélande																															
USA	Etats-Unis																															

Avec seulement les valeurs associées aux colonnes																												
Requête	Exemple																											
<div>Ajout d'une donnée</div> <div>INSERT INTO {nomTable} VALUES (     {valeurColonne1},     {valeurColonne2},     ... );</div>	<div>Dans un premier temps, on veut ajouter les réalisatrices Lana et Lilly Wachowski à la table "Realisateur". Pour se faire, on écrit les requêtes suivantes :</div> <div>INSERT INTO realisateur VALUES ('Wachowski', 'Lana', 1);  INSERT INTO realisateur VALUES ('Wachowski', 'Lilly', 1);</div>	<div>On a pour résultat :</div> <table><tr><th>id_realisateur</th><th>nom_realisateur</th><th>prenom_realisateur</th><th>sexe_realisateur</th></tr><tr><td>10</td><td>Foster</td><td>Jodie</td><td>1</td></tr><tr><td>11</td><td>Campion</td><td>Jane</td><td>1</td></tr><tr><td>12</td><td>Jenkins</td><td>Patty</td><td>1</td></tr><tr><td>13</td><td>Wachowski</td><td>Lana</td><td>1</td></tr><tr><td>14</td><td>Wachowski</td><td>Lilly</td><td>1</td></tr></table>			id_realisateur	nom_realisateur	prenom_realisateur	sexe_realisateur	10	Foster	Jodie	1	11	Campion	Jane	1	12	Jenkins	Patty	1	13	Wachowski	Lana	1	14	Wachowski	Lilly	1
	id_realisateur	nom_realisateur	prenom_realisateur	sexe_realisateur																								
	10	Foster	Jodie	1																								
11	Campion	Jane	1																									
12	Jenkins	Patty	1																									
13	Wachowski	Lana	1																									
14	Wachowski	Lilly	1																									
	<div>Dans un second temps, on veut ajouter la nationalité américaine (d'identifiant 'USA') à ces deux réalisatrices (ayant les identifiants 13 et 14) dans la table "Nationalite_Realisateur". Pour se faire, on écrit les requêtes suivantes :</div> <div>INSERT INTO nationalite_realisateur VALUES ('USA', 13);  INSERT INTO nationalite_realisateur VALUES ('USA', 14);</div>	<div>On a pour résultat :</div> <table><tr><th>nationalite_id</th><th>realisateur_id</th></tr><tr><td>USA</td><td>9</td></tr><tr><td>USA</td><td>10</td></tr><tr><td>USA</td><td>12</td></tr><tr><td>USA</td><td>13</td></tr><tr><td>USA</td><td>14</td></tr></table>			nationalite_id	realisateur_id	USA	9	USA	10	USA	12	USA	13	USA	14												
nationalite_id	realisateur_id																											
USA	9																											
USA	10																											
USA	12																											
USA	13																											
USA	14																											
	<div>Enfin, on veut ajouter le film "Matrix" (avec l'identifiant 35) à ces deux réalisatrices (ayant les identifiants 13 et 14) dans la table "Realisateur_Film". Pour se faire, on écrit les requêtes suivantes :</div> <div>INSERT INTO realisateur_film VALUES (13, 35);  INSERT INTO realisateur_film VALUES (14, 35);</div>	<div>On a pour résultat :</div> <table><tr><th>realisateur_id</th><th>film_id</th></tr><tr><td>10</td><td>6</td></tr><tr><td>11</td><td>3</td></tr><tr><td>11</td><td>4</td></tr><tr><td>12</td><td>1</td></tr><tr><td>12</td><td>2</td></tr><tr><td>13</td><td>35</td></tr><tr><td>14</td><td>35</td></tr></table>			realisateur_id	film_id	10	6	11	3	11	4	12	1	12	2	13	35	14	35								
realisateur_id	film_id																											
10	6																											
11	3																											
11	4																											
12	1																											
12	2																											
13	35																											
14	35																											

Dans un premier temps, on veut ajouter trois réalisateurs à la table “Realisateur” : Pedro Almodòvar, Akira Kurosawa et Bong Joon-ho. Pour se faire, on écrit la requête suivante :

```
INSERT INTO realisateur
VALUES
    ('Almodòvar', 'Pedro', 0),
    ('Kurosawa', 'Akira', 0),
    ('Joon-ho', 'Bong', 0);
```

On a pour résultat :

id_realisateur	nom_realisateur	prenom_realisateur	sexe_realisateur
12	Jenkins	Patty	1
13	Wachowski	Lana	1
14	Wachowski	Lilly	1
15	Almodòvar	Pedro	0
16	Kurosawa	Akira	0
17	Joon-ho	Bong	0

Ajout de plusieurs données

```
INSERT INTO {nomTable}
VALUES
(
    {valeurColonne1},
    {valeurColonne2},
    ...
),
(
    {valeurColonne1},
    {valeurColonne2},
    ...
);
```

Dans un second temps, dans la table “Nationalite\_Realisateur”, on veut ajouter les nationalités suivantes :

- espagnole (d’identifiant ‘ESP’) à Pedro Almodòvar (ayant l’identifiants 15),
- japonaise (d’identifiant ‘JPN’) à Akira Kurosawa (ayant l’identifiant 16),
- sud-coréenne (d’identifiant ‘KOR’) à Bong Joon-ho (ayant l’identifiant 17) .

Pour se faire, on écrit les requêtes suivantes :

```
INSERT INTO nationalite_realisateur
VALUES
    ('ESP', 15),
    ('JPN', 16),
    ('KOR', 17);
```

On a pour résultat :

nationalite_id	realisateur_id
ESP	15
FRA	4
FRA	5
FRA	9
GBR	6
GBR	8
ITA	7
JPN	16
KOR	17

Enfin, dans la table “Realisateur\_Film”, on veut ajouter les films suivants :

- “Parasite” (avec l’identifiant 36) au réalisateur Bong Joon-ho (ayant l’identifiant 17),
- “Les Sept Samourais” (avec l’identifiant 37) au réalisateur Akira Kurosawa (ayant l’identifiant 16)
- “Julieta” (avec l’identifiant 38) au réalisateur Pedro Almodòvar (ayant l’identifiant 15).

Pour se faire, on écrit les requêtes suivantes :

```
INSERT INTO realisateur_film
VALUES
    (17, 36),
    (16, 37),
    (15, 38);
```

On a pour résultat :

realisateur_id	film_id
13	35
14	35
15	38
16	37
17	36

/!\ POINTS IMPORTANTS /!\ :

- Lors de l’insertion des données, on ne prend pas en compte les colonnes avec la contrainte “IDENTITY” (Auto-incrément).
- Il faut remplir les tables d’entrée dans un premier temps et ensuite les autres. Pour rappel les tables d’entrée sont les tables ne recevant pas de FOREIGN KEY (clé étrangère).
- Lors de l’insertion des données, toutes les colonnes indiquées entre parenthèses doivent ABSOLUMENT recevoir une donnée.
- Les valeurs affectées doivent respectivement correspondre à la colonne en question.
- Dans la deuxième possibilité, il faut IMPÉRATIVEMENT entrer toutes les valeurs dans le même ordre que celui des colonnes sans oubli(s), sous peine de se trouver face à des conflits de clés.

SELECT : permet de sélectionner une ou des données d’une ou plusieurs tables dans une base de données.

Sélectionner des données d'une colonne d'une table.																						
Requête	Exemple																					
<p><u>Sélection des données</u></p> <p>SELECT {nomColonne} FROM {nomTable};</p>	<p>On veut sélectionner l'ensemble des genres de films dans la table "Genre". Pour se faire, on écrit la requête suivante :</p> <p>SELECT nom_genre FROM genre;</p>	<p>On a pour résultat :</p> <table><tr><th>nom_genre</th><th>nom_genre</th></tr><tr><td>Action</td><td>Western</td></tr><tr><td>Aventure</td><td>Science-fiction</td></tr><tr><td>Comédie</td><td>Romance</td></tr><tr><td>Comédie dramatique</td><td>Biopic</td></tr><tr><td>Drame</td><td>Thriller</td></tr><tr><td>Fantastique</td><td>Musical</td></tr><tr><td>Guerre</td><td>Comédie musicale</td></tr><tr><td>Policier</td><td>Historique</td></tr><tr><td>Epouvante-Horreur</td><td>Arts Martiaux</td></tr></table>	nom_genre	nom_genre	Action	Western	Aventure	Science-fiction	Comédie	Romance	Comédie dramatique	Biopic	Drame	Thriller	Fantastique	Musical	Guerre	Comédie musicale	Policier	Historique	Epouvante-Horreur	Arts Martiaux
nom_genre	nom_genre																					
Action	Western																					
Aventure	Science-fiction																					
Comédie	Romance																					
Comédie dramatique	Biopic																					
Drame	Thriller																					
Fantastique	Musical																					
Guerre	Comédie musicale																					
Policier	Historique																					
Epouvante-Horreur	Arts Martiaux																					

Sélectionner des données de plusieurs colonnes d’une table.																								
Requête	Exemple																							
<p><u>Sélection de plusieurs données</u></p> <p>SELECT</p> <p>    {nomColonne1},</p> <p>    {nomColonne2},</p> <p>    ...</p> <p>FROM {nomTable};</p>	<p>On veut sélectionner l’ensemble des noms et des prénoms des utilisateurs de la table “Utilisateur”. Pour se faire, on écrit la requête suivante :</p> <p>SELECT</p> <p>    nom_utilisateur,</p> <p>    prenom_utilisateur</p> <p>FROM utilisateur;</p>	<p>On a pour résultat :</p> <table><tr><th>nom_utilisateur</th><th>prenom_utilisateur</th></tr><tr><td>Faure</td><td>Aurélie</td></tr><tr><td>Dumez</td><td>Carole</td></tr><tr><td>Lapraz</td><td>Danielle</td></tr><tr><td>Luciano</td><td>Florian</td></tr><tr><td>Lapraz</td><td>Isabelle</td></tr><tr><td>Pierre</td><td>Jean</td></tr><tr><td>Luciano</td><td>Marius</td></tr><tr><td>Pendragon</td><td>Arthur</td></tr><tr><td>Poupoule</td><td>Kadoc</td></tr><tr><td>Luciano</td><td>Romain</td></tr></table>	nom_utilisateur	prenom_utilisateur	Faure	Aurélie	Dumez	Carole	Lapraz	Danielle	Luciano	Florian	Lapraz	Isabelle	Pierre	Jean	Luciano	Marius	Pendragon	Arthur	Poupoule	Kadoc	Luciano	Romain
nom_utilisateur	prenom_utilisateur																							
Faure	Aurélie																							
Dumez	Carole																							
Lapraz	Danielle																							
Luciano	Florian																							
Lapraz	Isabelle																							
Pierre	Jean																							
Luciano	Marius																							
Pendragon	Arthur																							
Poupoule	Kadoc																							
Luciano	Romain																							

Sélectionner toutes les données de toutes les colonnes d'une table.																																															
Requête	Exemple																																														
<p><u>Sélection des données</u></p> <p>SELECT * FROM {nomTable};</p>	<p>On veut sélectionner l'ensemble des données de la table "Utilisateur". Pour se faire, on écrit la requête suivante :</p> <p>SELECT * FROM utilisateur;</p>	<p>On a pour résultat :</p> <table><tr><th>id_utilisateur</th><th>nom_utilisateur</th><th>prenom_utilisateur</th><th>mdp_utilisateur</th></tr><tr><td>Aurelie13</td><td>Faure</td><td>Aurélie</td><td>aurFlor_89</td></tr><tr><td>Carole58</td><td>Dumez</td><td>Carole</td><td>dumCarole_58</td></tr><tr><td>Daniel47</td><td>Lapraz</td><td>Danielle</td><td>danielleLPZ_47</td></tr><tr><td>FloSpel</td><td>Luciano</td><td>Florian</td><td>flopiLuc_91</td></tr><tr><td>Isabelle87</td><td>Lapraz</td><td>Isabelle</td><td>isaRetsuko_87</td></tr><tr><td>JeanP57</td><td>Pierre</td><td>Jean</td><td>jeanPierre_57</td></tr><tr><td>Marius58</td><td>Luciano</td><td>Marius</td><td>marLUC_58</td></tr><tr><td>MiOurs</td><td>Pendragon</td><td>Arthur</td><td>Excalibur_9999</td></tr><tr><td>MorduKadoc</td><td>Poupoule</td><td>Kadoc</td><td>KPP_mordu45</td></tr><tr><td>Romain06</td><td>Luciano</td><td>Romain</td><td>romainLuc_06</td></tr></table>		id_utilisateur	nom_utilisateur	prenom_utilisateur	mdp_utilisateur	Aurelie13	Faure	Aurélie	aurFlor_89	Carole58	Dumez	Carole	dumCarole_58	Daniel47	Lapraz	Danielle	danielleLPZ_47	FloSpel	Luciano	Florian	flopiLuc_91	Isabelle87	Lapraz	Isabelle	isaRetsuko_87	JeanP57	Pierre	Jean	jeanPierre_57	Marius58	Luciano	Marius	marLUC_58	MiOurs	Pendragon	Arthur	Excalibur_9999	MorduKadoc	Poupoule	Kadoc	KPP_mordu45	Romain06	Luciano	Romain	romainLuc_06
id_utilisateur	nom_utilisateur	prenom_utilisateur	mdp_utilisateur																																												
Aurelie13	Faure	Aurélie	aurFlor_89																																												
Carole58	Dumez	Carole	dumCarole_58																																												
Daniel47	Lapraz	Danielle	danielleLPZ_47																																												
FloSpel	Luciano	Florian	flopiLuc_91																																												
Isabelle87	Lapraz	Isabelle	isaRetsuko_87																																												
JeanP57	Pierre	Jean	jeanPierre_57																																												
Marius58	Luciano	Marius	marLUC_58																																												
MiOurs	Pendragon	Arthur	Excalibur_9999																																												
MorduKadoc	Poupoule	Kadoc	KPP_mordu45																																												
Romain06	Luciano	Romain	romainLuc_06																																												

Sélectionner une donnée d'une colonne d'une table avec la clause WHERE.				
Requête	Exemple			
<p><u>Sélection d'une donnée</u></p> <p>SELECT {nomColonne} FROM {nomTable} WHERE {nomColonne} = valeur;</p>	<p>On veut sélectionner le nom du film ayant l'identifiant n°15 dans la table "Film". Pour se faire, on écrit la requête suivante :</p> <p>SELECT nom_film FROM film WHERE id_film = 15;</p>	<p>On a pour résultat :</p> <table><tr><th>nom_film</th></tr><tr><td>Thelma et Louise</td></tr></table>	nom_film	Thelma et Louise
nom_film				
Thelma et Louise				

Sélectionner des données de plusieurs colonnes d'une table avec la clause WHERE.								
Requête	Exemple							
<p><u>Sélection des données</u></p> <pre>SELECT     {nomColonne1},     {nomColonne2},     ... FROM {nomTable} WHERE {nomColonne} = valeur;</pre>	<p>On veut sélectionner le nom, le prénom et le sexe du réalisateur ayant l'identifiant n°6 dans la table "Realisateur". Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     nom_realisateur,     prenom_realisateur,     sexe_realisateur FROM realisateur WHERE id_realisateur = 6;</pre>	<p>On a pour résultat :</p> <table> <tr> <th>nom_realisateur</th><th>prenom_realisateur</th><th>sexe_realisateur</th></tr> <tr> <td>Scott</td><td>Ridley</td><td>0</td></tr> </table>	nom_realisateur	prenom_realisateur	sexe_realisateur	Scott	Ridley	0
nom_realisateur	prenom_realisateur	sexe_realisateur						
Scott	Ridley	0						

Sélectionner une ou des données d'une table en appliquant des [fonctions](#).

Fonctions	Requête	Exemple																							
<p><b>LOWER</b></p> <p>mettre en minuscule une chaîne de caractères</p>	<pre>SELECT LOWER({nomColonne}) FROM {nomTable};</pre>	<p>Dans la table "Nationalite", on veut passer les identifiants de nationalité en minuscule. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT LOWER(id_nationalite) AS nationalite FROM nationalite;</pre>	<p>On a pour résultat :</p> <table><thead><tr><th>nationalite</th><th>nationalite</th></tr></thead><tbody><tr><td>chn</td><td>fra</td></tr><tr><td>esp</td><td>gbr</td></tr><tr><td>fra</td><td>ita</td></tr><tr><td>gbr</td><td>jpn</td></tr><tr><td>ita</td><td></td></tr></tbody></table>	nationalite	nationalite	chn	fra	esp	gbr	fra	ita	gbr	jpn	ita											
nationalite	nationalite																								
chn	fra																								
esp	gbr																								
fra	ita																								
gbr	jpn																								
ita																									
<p><b>UPPER</b></p> <p>mettre en majuscule une chaîne de caractères</p>	<pre>SELECT UPPER({nomColonne}) FROM {nomTable};</pre>	<p>Dans la table "Realisateur", on veut passer les noms des réalisateurs en majuscule. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT UPPER(nom_realisateur) AS realisateur FROM realisateur;</pre>	<p>On a pour résultat :</p> <table><thead><tr><th>realisateur</th><th>realisateur</th></tr></thead><tbody><tr><td>LUCAS</td><td>FOSTER</td></tr><tr><td>COPPOLA</td><td>CAMPION</td></tr><tr><td>ZHAO</td><td>JENKINS</td></tr><tr><td>LE BESCO</td><td>WACHOWSKI</td></tr><tr><td>GANS</td><td>WACHOWSKI</td></tr><tr><td>SCOTT</td><td>ALMODÓVAR</td></tr><tr><td>BENIGNI</td><td>KUROSAWA</td></tr><tr><td>GARLAND</td><td>JOON-HO</td></tr><tr><td>CHAZELLE</td><td></td></tr></tbody></table>	realisateur	realisateur	LUCAS	FOSTER	COPPOLA	CAMPION	ZHAO	JENKINS	LE BESCO	WACHOWSKI	GANS	WACHOWSKI	SCOTT	ALMODÓVAR	BENIGNI	KUROSAWA	GARLAND	JOON-HO	CHAZELLE			
realisateur	realisateur																								
LUCAS	FOSTER																								
COPPOLA	CAMPION																								
ZHAO	JENKINS																								
LE BESCO	WACHOWSKI																								
GANS	WACHOWSKI																								
SCOTT	ALMODÓVAR																								
BENIGNI	KUROSAWA																								
GARLAND	JOON-HO																								
CHAZELLE																									
<p><b>TOP</b></p> <p>afficher les x premiers résultats d'une sélection</p>	<p><b>Sans l'utilisation de la clause WHERE</b></p> <pre>SELECT TOP {nombreChoisi} ({nomColonne}) FROM {nomTable}</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer les 10 premières notes. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT TOP 10 (note_film) FROM utilisateur_film;</pre>	<p>On a pour résultat :</p> <table><thead><tr><th></th><th>note_film</th></tr></thead><tbody><tr><td>1</td><td>4</td></tr><tr><td>2</td><td>9</td></tr><tr><td>3</td><td>10</td></tr><tr><td>4</td><td>6</td></tr><tr><td>5</td><td>10</td></tr><tr><td>6</td><td>4</td></tr><tr><td>7</td><td>4</td></tr><tr><td>8</td><td>3</td></tr><tr><td>9</td><td>6</td></tr><tr><td>10</td><td>9</td></tr></tbody></table>		note_film	1	4	2	9	3	10	4	6	5	10	6	4	7	4	8	3	9	6	10	9
		note_film																							
1	4																								
2	9																								
3	10																								
4	6																								
5	10																								
6	4																								
7	4																								
8	3																								
9	6																								
10	9																								
	<p><b>Avec l'utilisation de la clause WHERE</b></p> <pre>SELECT TOP {nombreChoisi} ({nomColonne}) FROM {nomTable} WHERE {nomColonne} = valeur;</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer les 5 premières notes ainsi que l'identifiant des films notés par l'utilisateur ayant l'identifiant "MorduKadoc". Pour se faire, on écrit la requête suivante :</p> <pre>SELECT TOP 5 (note_film), film_id FROM utilisateur_film WHERE utilisateur_id = 'MorduKadoc';</pre>	<p>On a pour résultat :</p> <table><thead><tr><th></th><th>note_film</th><th>film_id</th></tr></thead><tbody><tr><td>1</td><td>9</td><td>1</td></tr><tr><td>2</td><td>9</td><td>6</td></tr><tr><td>3</td><td>10</td><td>9</td></tr><tr><td>4</td><td>5</td><td>14</td></tr><tr><td>5</td><td>8</td><td>18</td></tr></tbody></table>		note_film	film_id	1	9	1	2	9	6	3	10	9	4	5	14	5	8	18				
	note_film	film_id																							
1	9	1																							
2	9	6																							
3	10	9																							
4	5	14																							
5	8	18																							



<p><b>SUM</b></p> <p>afficher la somme des valeurs d'une sélection</p>	<p><u>Sans l'utilisation de la clause WHERE</u></p> <pre>SELECT     SUM ({nomColonne}) FROM {nomTable};</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer la somme de toutes les notes postées. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     SUM (note_film) AS total_notes FROM utilisateur_film;</pre>	<p>On a pour résultat :</p> 
	<p><u>Avec l'utilisation de la clause WHERE</u></p> <pre>SELECT     SUM ({nomColonne}) FROM {nomTable} WHERE {nomColonne} = valeur;</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer la somme de toutes les notes postées pour le film ayant l'identifiant n°34. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     SUM (note_film) AS total_notes_SW1 FROM utilisateur_film WHERE film_id = 34;</pre>	<p>On a pour résultat :</p> 
<p><b>AVG</b></p> <p>afficher la moyenne des valeurs d'une sélection</p>	<p><u>Sans l'utilisation de la clause WHERE</u></p> <pre>SELECT     AVG ({nomColonne}) FROM {nomTable};</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer la moyenne de toutes les notes postées. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     AVG (note_film) AS moyenne_notes FROM utilisateur_film;</pre>	<p>On a pour résultat :</p> 
	<p><u>Avec l'utilisation de la clause WHERE</u></p> <pre>SELECT     AVG ({nomColonne}) FROM {nomTable} WHERE {nomColonne} = valeur;</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer la moyenne de toutes les notes postées pour le film ayant l'identifiant n°27. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     AVG (note_film) AS moyenne_notes_ETN FROM utilisateur_film WHERE film_id = 27;</pre>	<p>On a pour résultat :</p> 
<p><b>COUNT</b></p> <p>retourne le nombre de lignes qui correspond à un critère spécifique</p>	<p><u>Sans l'utilisation de la clause WHERE</u></p> <pre>SELECT     COUNT ({nomColonne}) FROM {nomTable};</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer le nombre total de notes postées tous films confondus. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     COUNT (note_film) AS nombre_notes FROM utilisateur_film;</pre>	<p>On a pour résultat :</p> 
	<p><u>Avec l'utilisation de la clause WHERE</u></p> <pre>SELECT     COUNT ({nomColonne}) FROM {nomTable} WHERE {nomColonne} = valeur;</pre>	<p>Dans la table "Utilisateur_Film", on veut récupérer le nombre total de notes postées pour le film ayant l'identifiant n°12. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     COUNT (note_film) AS nombre_notes_VB FROM utilisateur_film WHERE film_id = 12;</pre>	<p>On a pour résultat :</p> 

/!\ POINTS IMPORTANTS /!\ :

- La liste des **fonctions** présentées ci-dessus est non exhaustive : il en existe plusieurs autres.
- Les **fonctions** présentées ci-dessus peuvent également se cumuler au sein d'une même requête.

### Utilisation de plusieurs **fonctions** au sein d'une même requête

#### Exemple

Dans la table "Utilisateur\_Film", on veut récupérer à la fois le nombre, la somme et la moyenne de toutes les notes du film ayant l'identifiant n°7. Pour se faire, on écrit la requête suivante :

```
SELECT
    COUNT(note_film) AS nombre_notes_whiplash,
    SUM(note_film) AS somme_notes_whiplash,
    AVG(note_film) AS moyenne_notes_whiplash
FROM utilisateur_film
WHERE film_id = 7;
```

On a pour résultat :

nombre_notes_whiplash	somme_notes_whiplash	moyenne_notes_whiplash
6	38	6

UPDATE : permet de modifier une ou des données d’une table dans une base de données.

Modifier une donnée dans une colonne d’une table.																										
Requête	Exemple																									
<p><u>Modification d’une donnée</u></p> <p>UPDATE {nomTable} SET {nomColonne} = valeur WHERE {nomColonne} = valeur;</p>	<p>On veut modifier dans la table “Film” les années de sortie des films suivants :</p> <ul style="list-style-type: none"><li>• 1999 pour “Matrix”,</li><li>• 2019 pour “Parasite”,</li><li>• 1954 pour “Les Sept Samourais”,</li><li>• 2016 pour “Julieta”.</li></ul> <p>Pour se faire, on écrit les requêtes suivantes :</p> <p>UPDATE film SET annee_sortie_film = 1999 WHERE nom_film = 'Matrix';</p> <p>UPDATE film SET annee_sortie_film = 2019 WHERE nom_film = 'Parasite';</p> <p>UPDATE film SET annee_sortie_film = 1954 WHERE nom_film = 'Les Sept Samourais';</p> <p>UPDATE film SET annee_sortie_film = 2016 WHERE nom_film = 'Julieta';</p>	<p>On a pour résultat :</p> <table><tr><th>id_film</th><th>nom_film</th><th>annee_sortie_film</th></tr><tr><td>32</td><td>American Graffiti</td><td>1973</td></tr><tr><td>33</td><td>Star Wars : Episode IV - Un nouvel espoir</td><td>1977</td></tr><tr><td>34</td><td>Star Wars : Episode I - La menace fantôme</td><td>1999</td></tr><tr><td>35</td><td>Matrix</td><td>1999</td></tr><tr><td>36</td><td>Parasite</td><td>2019</td></tr><tr><td>37</td><td>Les Sept Samourais</td><td>1954</td></tr><tr><td>38</td><td>Julieta</td><td>2016</td></tr></table>	id_film	nom_film	annee_sortie_film	32	American Graffiti	1973	33	Star Wars : Episode IV - Un nouvel espoir	1977	34	Star Wars : Episode I - La menace fantôme	1999	35	Matrix	1999	36	Parasite	2019	37	Les Sept Samourais	1954	38	Julieta	2016
id_film	nom_film	annee_sortie_film																								
32	American Graffiti	1973																								
33	Star Wars : Episode IV - Un nouvel espoir	1977																								
34	Star Wars : Episode I - La menace fantôme	1999																								
35	Matrix	1999																								
36	Parasite	2019																								
37	Les Sept Samourais	1954																								
38	Julieta	2016																								

Modifier des données dans plusieurs colonnes d’une table.																															
Requête	Exemple																														
<p><u>Modification de plusieurs données</u></p> <p>UPDATE {nomTable}</p> <p>SET</p> <p>        {nomColonne1} = valeur1,</p> <p>        {nomColonne2} = valeur2,</p> <p>        ...</p> <p>WHERE {nomColonne} = valeur;</p>	<p>Sans la table “Utilisateur”, on veut modifier le nom et le prénom de l'utilisateur ayant l'identifiant “JeanP57” et les remplacer par “Paul” pour le nom et “Joann” pour le prénom. Pour se faire, on écrit la requête suivante :</p> <p>UPDATE utilisateur</p> <p>SET</p> <p>nom_utilisateur = 'Paul',</p> <p>prenom_utilisateur = 'Joann'</p> <p>WHERE id_utilisateur = 'JeanP57';</p>	<p>On a pour résultat :</p> <table><tr><td>id_utilisateur</td><td>nom_utilisateur</td><td>prenom_utilisateur</td><td>mdp_utilisateur</td></tr><tr><td>Isabelle87</td><td>Lapraz</td><td>Isabelle</td><td>isaRetsuko_87</td></tr><tr><td>JeanP57</td><td>Paul</td><td>Joann</td><td>jeanPierre_57</td></tr><tr><td>Marius58</td><td>Luciano</td><td>Marius</td><td>marLUC_58</td></tr><tr><td>MiOurs</td><td>Pendragon</td><td>Arthur</td><td>Excalibur_9999</td></tr><tr><td>MorduKadoc</td><td>Poupoule</td><td>Kadoc</td><td>KPP_mordu45</td></tr><tr><td>Romain06</td><td>Luciano</td><td>Romain</td><td>romainLuc_06</td></tr></table>		id_utilisateur	nom_utilisateur	prenom_utilisateur	mdp_utilisateur	Isabelle87	Lapraz	Isabelle	isaRetsuko_87	JeanP57	Paul	Joann	jeanPierre_57	Marius58	Luciano	Marius	marLUC_58	MiOurs	Pendragon	Arthur	Excalibur_9999	MorduKadoc	Poupoule	Kadoc	KPP_mordu45	Romain06	Luciano	Romain	romainLuc_06
id_utilisateur	nom_utilisateur	prenom_utilisateur	mdp_utilisateur																												
Isabelle87	Lapraz	Isabelle	isaRetsuko_87																												
JeanP57	Paul	Joann	jeanPierre_57																												
Marius58	Luciano	Marius	marLUC_58																												
MiOurs	Pendragon	Arthur	Excalibur_9999																												
MorduKadoc	Poupoule	Kadoc	KPP_mordu45																												
Romain06	Luciano	Romain	romainLuc_06																												

/!\ POINTS IMPORTANTS /!\ :

→ Faites attention quand vous modifiez des données avec l’instruction UPDATE. Si vous oubliez la clause WHERE, toutes les données de votre table seront modifiées !

DELETE : permet de supprimer une ou des données dans une table.

Supprimer les données d’une table en précisant la ou les données à supprimer														
Requête	Exemple													
<pre>DELETE {nomTable} WHERE {nomColonne} = valeur;</pre>	<p>Dans la table “Genre”, on veut supprimer le genre de film “Arts Martiaux” possédant l’identifiant n°18. Pour se faire, on écrit la requête suivante :</p> <pre>DELETE genre WHERE id_genre = 18;</pre>	<p>On a pour résultat :</p> <table><tr><th>id_genre</th><th>nom_genre</th></tr><tr><td>13</td><td>Biopic</td></tr><tr><td>14</td><td>Thriller</td></tr><tr><td>15</td><td>Musical</td></tr><tr><td>16</td><td>Comédie musicale</td></tr><tr><td>17</td><td>Historique</td></tr></table>	id_genre	nom_genre	13	Biopic	14	Thriller	15	Musical	16	Comédie musicale	17	Historique
id_genre	nom_genre													
13	Biopic													
14	Thriller													
15	Musical													
16	Comédie musicale													
17	Historique													

/ ! \ POINTS IMPORTANTS / ! \ :  
→ Dans le paramètre {nomColonne} de la clause WHERE, on indique **TOUJOURS la référence unique**.

Supprimer l'ensemble des données d'une table sans précision		
Requête	Exemple	
<code>DELETE {nomTable}</code>	On veut supprimer l'ensemble des données provenant de la table "Genre". Pour se faire, on écrit la requête suivante :  <code>DELETE genre ;</code>	Comme résultat, l'ensemble des données de la table "Genre" est supprimé.

/ ! \ POINTS IMPORTANTS / ! \ :  
→ Cette requête efface de manière **DEFINITIVE toutes les données** se trouvant dans une table !  
→ Cependant, il faut noter que **la structure de la table persiste** après l'utilisation de la commande DELETE.

AS (Alias) : permet de donner un ou des noms temporaires à une ou des colonnes, à une ou des tables.

Donner un nom temporaire à une colonne.													
Requête	Exemple												
<p><a href="#">Donner nom temporaire à une colonne</a></p> <p>SELECT {nomColonne} AS {nomAlias}</p> <p>FROM {nomTable}</p>	<p>Dans la table "Utilisateur", on veut récupérer l'ensemble des mots de passe sous le nom de colonne "mot_de_passe". Pour se faire, on écrit la requête suivante :</p> <p>SELECT mdp_utilisateur AS mot_de_passe</p> <p>FROM utilisateur;</p>	<p>On a pour résultat :</p> <table><tr><td>mot_de_passe</td></tr><tr><td>aurFlor_89</td></tr><tr><td>dumCarole_58</td></tr><tr><td>danielleLPZ_47</td></tr><tr><td>flopiLuc_91</td></tr><tr><td>isaRetsuko_87</td></tr><tr><td>jeanPierre_57</td></tr><tr><td>marLUC_58</td></tr><tr><td>Excalibur_9999</td></tr><tr><td>KPP_mordu45</td></tr><tr><td>romainLuc_06</td></tr></table>	mot_de_passe	aurFlor_89	dumCarole_58	danielleLPZ_47	flopiLuc_91	isaRetsuko_87	jeanPierre_57	marLUC_58	Excalibur_9999	KPP_mordu45	romainLuc_06
mot_de_passe													
aurFlor_89													
dumCarole_58													
danielleLPZ_47													
flopiLuc_91													
isaRetsuko_87													
jeanPierre_57													
marLUC_58													
Excalibur_9999													
KPP_mordu45													
romainLuc_06													

Donner des noms temporaires à plusieurs colonnes.																																			
Requête	Exemple																																		
<p><a href="#">Donner nom temporaire à plusieurs colonnes</a></p> <pre>SELECT     {nomColonne1} AS {nomAlias1},     {nomColonne2} AS {nomAlias2}     ... FROM {nomTable};</pre>	<p>Dans la table "Utilisateur", on veut récupérer l'ensemble des identifiants, noms et prénoms sous les noms de colonnes respectifs "identifiants", "noms" et "prenoms". Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     id_utilisateur AS identifiants,     nom_utilisateur AS noms,     prenom_utilisateur AS prenoms FROM utilisateur;</pre>	<p>On a pour résultat :</p> <table><tr><th>identifiants</th><th>noms</th><th>prenoms</th></tr><tr><td>Aurelie13</td><td>Faure</td><td>Aurélie</td></tr><tr><td>Carole58</td><td>Dumez</td><td>Carole</td></tr><tr><td>Daniel47</td><td>Lapraz</td><td>Danielle</td></tr><tr><td>FloSpel</td><td>Luciano</td><td>Florian</td></tr><tr><td>Isabelle87</td><td>Lapraz</td><td>Isabelle</td></tr><tr><td>JeanP57</td><td>Paul</td><td>Joann</td></tr><tr><td>Marius58</td><td>Luciano</td><td>Marius</td></tr><tr><td>MiOurs</td><td>Pendragon</td><td>Arthur</td></tr><tr><td>MorduKadoc</td><td>Poupoule</td><td>Kadoc</td></tr><tr><td>Romain06</td><td>Luciano</td><td>Romain</td></tr></table>	identifiants	noms	prenoms	Aurelie13	Faure	Aurélie	Carole58	Dumez	Carole	Daniel47	Lapraz	Danielle	FloSpel	Luciano	Florian	Isabelle87	Lapraz	Isabelle	JeanP57	Paul	Joann	Marius58	Luciano	Marius	MiOurs	Pendragon	Arthur	MorduKadoc	Poupoule	Kadoc	Romain06	Luciano	Romain
identifiants	noms	prenoms																																	
Aurelie13	Faure	Aurélie																																	
Carole58	Dumez	Carole																																	
Daniel47	Lapraz	Danielle																																	
FloSpel	Luciano	Florian																																	
Isabelle87	Lapraz	Isabelle																																	
JeanP57	Paul	Joann																																	
Marius58	Luciano	Marius																																	
MiOurs	Pendragon	Arthur																																	
MorduKadoc	Poupoule	Kadoc																																	
Romain06	Luciano	Romain																																	

Donner un nom temporaire à une table.													
Requête	Exemple												
<p><a href="#">Donner nom temporaire à une table</a></p> <pre>SELECT {nomColonne} FROM {nomTable} AS {nomAlias};</pre>	<p>Grâce à la table “Utilisateur” que l’on renomme “Users”, on veut afficher tous les noms des utilisateurs enregistrés dans une colonne nommée “registred_users”. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT users.nom_utilisateur AS registred_users FROM utilisateur AS users</pre>	<p>On a pour résultat :</p> <table><tr><th>registred_users</th></tr><tr><td>Faure</td></tr><tr><td>Dumez</td></tr><tr><td>Lapraz</td></tr><tr><td>Luciano</td></tr><tr><td>Lapraz</td></tr><tr><td>Paul</td></tr><tr><td>Luciano</td></tr><tr><td>Pendragon</td></tr><tr><td>Poupoule</td></tr><tr><td>Luciano</td></tr></table>	registred_users	Faure	Dumez	Lapraz	Luciano	Lapraz	Paul	Luciano	Pendragon	Poupoule	Luciano
registred_users													
Faure													
Dumez													
Lapraz													
Luciano													
Lapraz													
Paul													
Luciano													
Pendragon													
Poupoule													
Luciano													

Donner des noms temporaires à plusieurs tables.											
Requête	Exemple										
<p><a href="#">Donner noms temporaires à plusieurs tables</a></p> <pre>SELECT     {nomAlias1}.{nomColonne1},     {nomAlias2}.{nomColonne2},     ... FROM     {nomTable1} AS {nomAlias1},     {nomTable2} AS {nomAlias2},     ... WHERE     {nomAlias1}.{nomColonneChoisie1} = valeur1,     {nomAlias2}.{nomColonneChoisie2} = valeur2;</pre>	<p>Grâce aux tables "Genre", "Film" et "Genre_Film" que l'on renomme respectivement "g", "f" et "gf" par l'utilisation des alias, on veut récupérer le nom du film, son année de sortie ainsi que les différents genres cinématographiques qui le caractérisent lorsque l'identifiant du film est le n°7. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     f.nom_film,     f.annee_sortie_film,     g.nom_genre FROM genre AS g, genre_film AS gf, film AS f WHERE     f.id_film = 7     AND g.id_genre = gf.genre_id     AND f.id_film = gf.film_id;</pre>	<p>On a pour résultat :</p> <table border="1"> <thead> <tr> <th>nom_film</th><th>annee_sortie_film</th><th>nom_genre</th></tr> </thead> <tbody> <tr> <td>Whiplash</td><td>2014</td><td>Drame</td></tr> <tr> <td>Whiplash</td><td>2014</td><td>Musical</td></tr> </tbody> </table>	nom_film	annee_sortie_film	nom_genre	Whiplash	2014	Drame	Whiplash	2014	Musical
nom_film	annee_sortie_film	nom_genre									
Whiplash	2014	Drame									
Whiplash	2014	Musical									

/!\ POINTS IMPORTANTS /!\ :

- Les **Alias** sont utiles quand il y a plus d'une table impliquée dans une requête, que des fonctions sont utilisées dans la requête, que les noms de colonnes sont trop grands ou illisibles (non explicites) ou encore quand deux colonnes ou plus sont combinées ensembles.
- Un **Alias** existe seulement le temps d'une requête.

GROUP BY : permet de regrouper les lignes qui ont les mêmes valeurs dans des lignes récapitulatives.

Requête	Exemple																							
<pre>SELECT     fonction({nomColonne}),     {nomColonne}, FROM {nomTable} GROUP BY {nomColonneChoisie};</pre>	<p>Grâce à la table "Utilisateur_Film", on veut lister le regroupement par utilisateur du nombre de notes postées. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     COUNT(note_film) AS nombre_note_utilisateur, utilisateur_id FROM utilisateur_film GROUP BY utilisateur_id;</pre>	<p>On a pour résultat :</p> <table><tr><th>nombre note utilisateur</th><th>utilisateur id</th></tr><tr><td>7</td><td>Aurelie13</td></tr><tr><td>7</td><td>Carole58</td></tr><tr><td>17</td><td>Daniel47</td></tr><tr><td>17</td><td>FloSpel</td></tr><tr><td>7</td><td>Isabelle87</td></tr><tr><td>9</td><td>JeanP57</td></tr><tr><td>14</td><td>Marius58</td></tr><tr><td>7</td><td>MiOurs</td></tr><tr><td>9</td><td>MorduKadoc</td></tr><tr><td>10</td><td>Romain06</td></tr></table>	nombre note utilisateur	utilisateur id	7	Aurelie13	7	Carole58	17	Daniel47	17	FloSpel	7	Isabelle87	9	JeanP57	14	Marius58	7	MiOurs	9	MorduKadoc	10	Romain06
nombre note utilisateur	utilisateur id																							
7	Aurelie13																							
7	Carole58																							
17	Daniel47																							
17	FloSpel																							
7	Isabelle87																							
9	JeanP57																							
14	Marius58																							
7	MiOurs																							
9	MorduKadoc																							
10	Romain06																							

/!\ POINTS IMPORTANTS /!\ :  
→ Le mot-clé **GROUP BY** est souvent utilisé avec les fonctions d'agrégation (**COUNT**, **MAX**, **MIN**, **SUM**, **AVG**) pour grouper les résultats en une ou plusieurs colonnes.

ORDER BY : permet de classer le résultat d'une requête dans un ordre ascendant ou descendant.

Requête	Exemple																																														
<pre>SELECT     {nomColonne1},     {nomColonne2},     ... FROM {nomTable} GROUP BY     {nomColonneChoisie1},     {nomColonneChoisie2},     ... ORDER BY     {nomColonneChoisie1},     {nomColonneChoisie2},     ... ASC   DESC;</pre>	Grâce à la table "Utilisateur_Film", on veut lister le regroupement par utilisateur du nombre de notes postées, dans un ordre ascendant (1) puis dans un ordre descendant (2). Pour se faire, on écrit les requêtes suivantes :		On a pour résultats :																																												
			Ordre Ascendant (ASC)	Ordre Descendant (DESC)																																											
	<pre>(1) : Ordre Ascendant SELECT     COUNT(note_film) AS     nombre_note_utilisateur,     utilisateur_id FROM     utilisateur_film GROUP BY     utilisateur_id ORDER BY     nombre_note_utilisateur ASC,     utilisateur_id ASC;</pre>	<pre>(2) : Ordre Descendant SELECT     COUNT(note_film) AS     nombre_note_utilisateur,     utilisateur_id FROM     utilisateur_film GROUP BY     utilisateur_id ORDER BY     nombre_note_utilisateur DESC,     utilisateur_id DESC;</pre>	<table><tr><th>nombre_note_utilisateur</th><th>utilisateur_id</th></tr><tr><td>7</td><td>Aurelie13</td></tr><tr><td>7</td><td>Carole58</td></tr><tr><td>7</td><td>Isabelle87</td></tr><tr><td>7</td><td>MiOurs</td></tr><tr><td>9</td><td>JeanP57</td></tr><tr><td>9</td><td>MorduKadoc</td></tr><tr><td>10</td><td>Romain06</td></tr><tr><td>14</td><td>Marius58</td></tr><tr><td>17</td><td>Daniel47</td></tr><tr><td>17</td><td>FloSpel</td></tr></table>	nombre_note_utilisateur	utilisateur_id	7	Aurelie13	7	Carole58	7	Isabelle87	7	MiOurs	9	JeanP57	9	MorduKadoc	10	Romain06	14	Marius58	17	Daniel47	17	FloSpel	<table><tr><th>nombre_note_utilisateur</th><th>utilisateur_id</th></tr><tr><td>17</td><td>FloSpel</td></tr><tr><td>17</td><td>Daniel47</td></tr><tr><td>14</td><td>Marius58</td></tr><tr><td>10</td><td>Romain06</td></tr><tr><td>9</td><td>MorduKadoc</td></tr><tr><td>9</td><td>JeanP57</td></tr><tr><td>7</td><td>MiOurs</td></tr><tr><td>7</td><td>Isabelle87</td></tr><tr><td>7</td><td>Carole58</td></tr><tr><td>7</td><td>Aurelie13</td></tr></table>	nombre_note_utilisateur	utilisateur_id	17	FloSpel	17	Daniel47	14	Marius58	10	Romain06	9	MorduKadoc	9	JeanP57	7	MiOurs	7	Isabelle87	7	Carole58	7
nombre_note_utilisateur	utilisateur_id																																														
7	Aurelie13																																														
7	Carole58																																														
7	Isabelle87																																														
7	MiOurs																																														
9	JeanP57																																														
9	MorduKadoc																																														
10	Romain06																																														
14	Marius58																																														
17	Daniel47																																														
17	FloSpel																																														
nombre_note_utilisateur	utilisateur_id																																														
17	FloSpel																																														
17	Daniel47																																														
14	Marius58																																														
10	Romain06																																														
9	MorduKadoc																																														
9	JeanP57																																														
7	MiOurs																																														
7	Isabelle87																																														
7	Carole58																																														
7	Aurelie13																																														

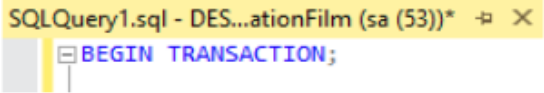
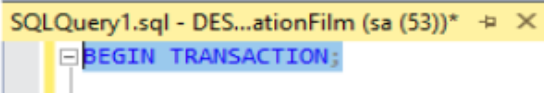
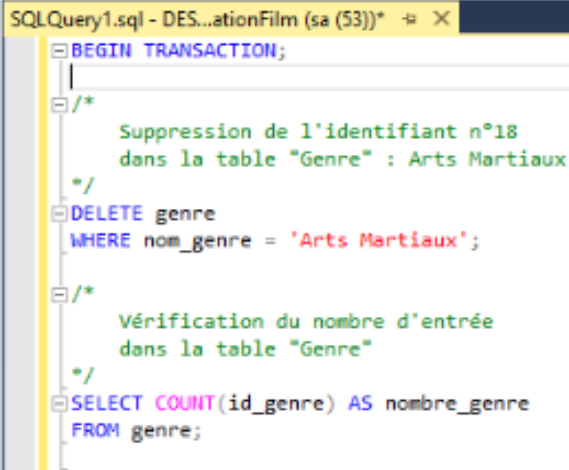

/!\ POINTS IMPORTANTS /!\ :  
→ Le mot-clé **ORDER BY** classe les résultats **par ordre ascendant** (du plus petit au plus grand) par défaut.

HAVING : permet, à partir de fonctions d'agrégations, de filtrer et extraire les seules données répondant à une condition spécifique.

Requête	Exemple											
<pre>SELECT     fonction({nomColonne1}),     {nomColonne2},     ... FROM {nomTable} WHERE {condition} GROUP BY     {nomColonneChoisie1},     {nomColonneChoisie2},     ... HAVING {condition} ORDER BY     {nomColonneChoisie1},     {nomColonneChoisie2},     ... ASC   DESC;</pre>	<p>Grâce à la table “Utilisateur_Film”, on veut lister le regroupement par utilisateur du nombre de notes postées supérieur à 9, classé de la valeur la plus basse à la valeur la plus haute. Pour se faire, on écrit la requête suivante :</p> <pre>SELECT     COUNT(note_film) AS nombre_note_utilisateur, utilisateur_id FROM     utilisateur_film GROUP BY     utilisateur_id HAVING     COUNT(note_film) &gt; 9 ORDER BY     nombre_note_utilisateur ASC,     utilisateur_id ASC;</pre>	<p>On a pour résultats :</p> <table><tr><th>nombre_note_utilisateur</th><th>utilisateur_id</th></tr><tr><td>10</td><td>Romain06</td></tr><tr><td>14</td><td>Marius58</td></tr><tr><td>17</td><td>Daniel47</td></tr><tr><td>17</td><td>FloSpel</td></tr></table>	nombre_note_utilisateur	utilisateur_id	10	Romain06	14	Marius58	17	Daniel47	17	FloSpel
nombre_note_utilisateur	utilisateur_id											
10	Romain06											
14	Marius58											
17	Daniel47											
17	FloSpel											



BEGIN TRANSACTION : permet de tester une requête avant de l'exécuter.

Etapas	Exemples
<p>I. Ecrire l'instruction "BEGIN TRANSACTION"</p>	
<p>II. Surligner "BEGIN TRANSACTION" et exécuter</p>	
<p>III. Ecrire la (les) requête(s) à tester</p> <p>On cherche à supprimer dans la table "Genre" le genre cinématographique ayant le nom "Arts Martiaux".</p> <p>Ensuite on cherche à vérifier que la suppression de ce genre est effective en comptant le nombre d'entrées dans la table "Genre", par le biais de la fonction COUNT().</p>	
<p>IV. Surligner la (les) requête(s) et exécuter</p> <p>Ici, on voit que le nombre de genres est de 17. L'entrée avec l'identifiant n°18 portant le nom "Arts Martiaux" est donc bien supprimée.</p>	

Deux situations possibles																									
Si le résultat recherché lors de l'exécution de la requête <b>est celui attendu</b>		Si le résultat recherché lors de l'exécution de la requête <b>n'est pas celui attendu</b>																							
Etapes	Exemples	Etapes	Exemples																						
I. Ecrire l'instruction "COMMIT"	<div>SQLQuery1.sql - DES...ationFilm (sa (54))*</div> <pre>BEGIN TRANSACTION;  /*   Suppression de l'identifiant n°18   dans la table "Genre" : Arts Martiaux */ DELETE genre WHERE nom_genre = 'Arts Martiaux';  /*   Vérification du nombre d'entrée   dans la table "Genre" */ SELECT COUNT(id_genre) AS nombre_genre FROM genre;  COMMIT;</pre>	I. Ecrire l'instruction "ROLLBACK"	<div>SQLQuery1.sql - non connecté*</div> <pre>BEGIN TRANSACTION;  /*   Suppression de l'identifiant n°18   dans la table "Genre" : Arts Martiaux */ DELETE genre WHERE nom_genre = 'Arts Martiaux';  /*   Vérification du nombre d'entrée   dans la table "Genre" */ SELECT COUNT(id_genre) AS nombre_genre FROM genre;  ROLLBACK;</pre>																						
II. Surligner "COMMIT" et exécuter	<div>SQLQuery1.sql - DES...ationFilm (sa (54))*</div> <pre>BEGIN TRANSACTION;  /*   Suppression de l'identifiant n°18   dans la table "Genre" : Arts Martiaux */ DELETE genre WHERE nom_genre = 'Arts Martiaux';  /*   Vérification du nombre d'entrée   dans la table "Genre" */ SELECT COUNT(id_genre) AS nombre_genre FROM genre;  COMMIT;</pre>	II. Surligner "ROLLBACK" et exécuter	<div>SQLQuery1.sql - non connecté*</div> <pre>BEGIN TRANSACTION;  /*   Suppression de l'identifiant n°18   dans la table "Genre" : Arts Martiaux */ DELETE genre WHERE nom_genre = 'Arts Martiaux';  /*   Vérification du nombre d'entrée   dans la table "Genre" */ SELECT COUNT(id_genre) AS nombre_genre FROM genre;  ROLLBACK;</pre>																						
III. Les modifications engendrées par la requête sont alors appliquées  Dans la table "Genre", l'identifiant n°18 de nom "Arts Martiaux" ne se trouve plus dans la table.	<table><thead><tr><th>id_genre</th><th>nom_genre</th></tr></thead><tbody><tr><td>14</td><td>Thriller</td></tr><tr><td>15</td><td>Musical</td></tr><tr><td>16</td><td>Comédie musicale</td></tr><tr><td>17</td><td>Historique</td></tr></tbody></table>	id_genre	nom_genre	14	Thriller	15	Musical	16	Comédie musicale	17	Historique	III. Les modifications engendrées par la requête ne sont pas appliquées  Dans la table "Genre", l'identifiant n°18 de nom "Arts Martiaux" se trouve toujours dans la table.	<table><thead><tr><th>id_genre</th><th>nom_genre</th></tr></thead><tbody><tr><td>14</td><td>Thriller</td></tr><tr><td>15</td><td>Musical</td></tr><tr><td>16</td><td>Comédie musicale</td></tr><tr><td>17</td><td>Historique</td></tr><tr><td>18</td><td>Arts Martiaux</td></tr></tbody></table>	id_genre	nom_genre	14	Thriller	15	Musical	16	Comédie musicale	17	Historique	18	Arts Martiaux
id_genre	nom_genre																								
14	Thriller																								
15	Musical																								
16	Comédie musicale																								
17	Historique																								
id_genre	nom_genre																								
14	Thriller																								
15	Musical																								
16	Comédie musicale																								
17	Historique																								
18	Arts Martiaux																								

!! POINTS IMPORTANTS !! :

→ Ne pas oublier d'écrire "COMMIT" OU "ROLLBACK" en fin de test avec "BEGIN TRANSACTION". Cet oubli entraîne un blocage des colonnes et des tables.