

Support de cours SQL Server



Contexte général

Qu'est-ce qu'une base de données?



Qu'est-ce qu'une
base de données ?



▸ Une **base de données** (en anglais ***database***), permet de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une activité ;

De quoi est composée une base de données ?

- Données
- Langage permettant de gérer les données :
Standard Query Language (SQL)
- Fichier de log

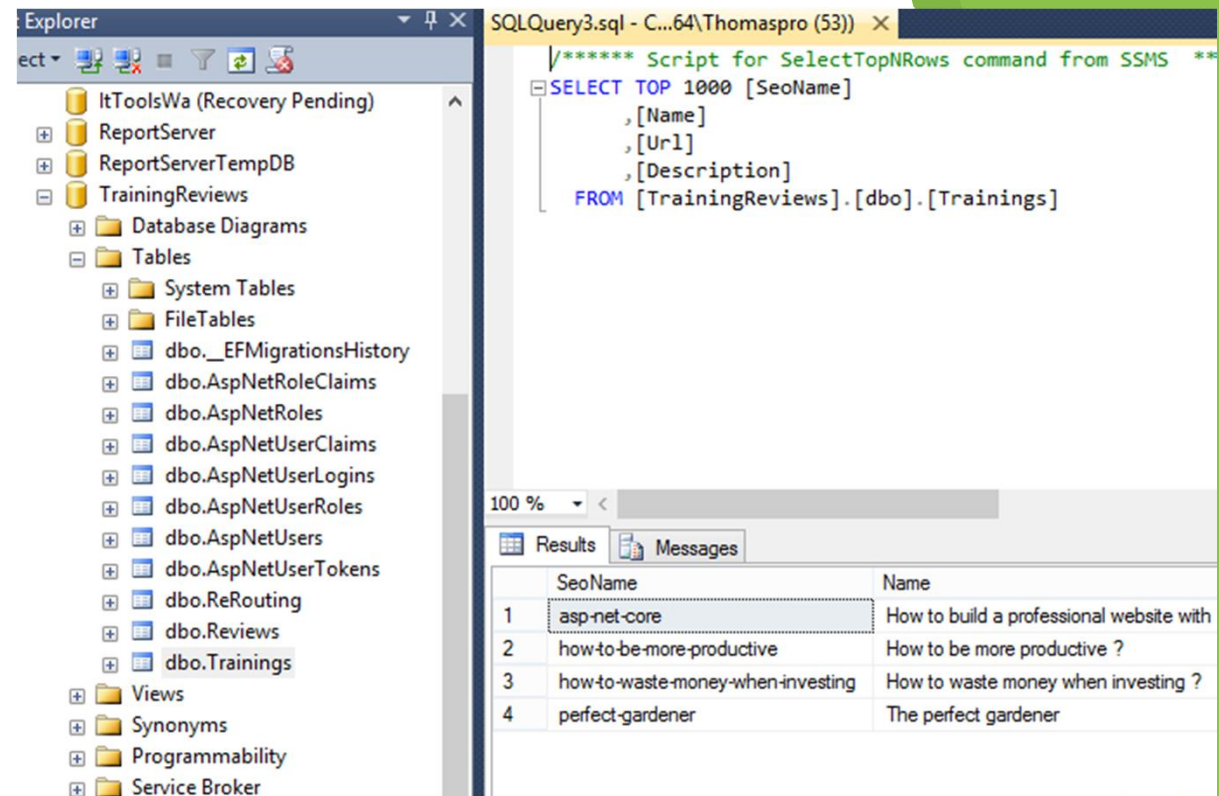
Pourquoi utiliser une base de données ?

- **Centraliser l'information**
- **Assurer l'intégrité des données**
- Accéder rapidement à l'information
- Recouper les données et traitements sur les données

*Exemple information disséminée contre
information centralisée – recherche dynamique*

Information stockée dans des tables

- Tables
- Colonnes
- Lignes

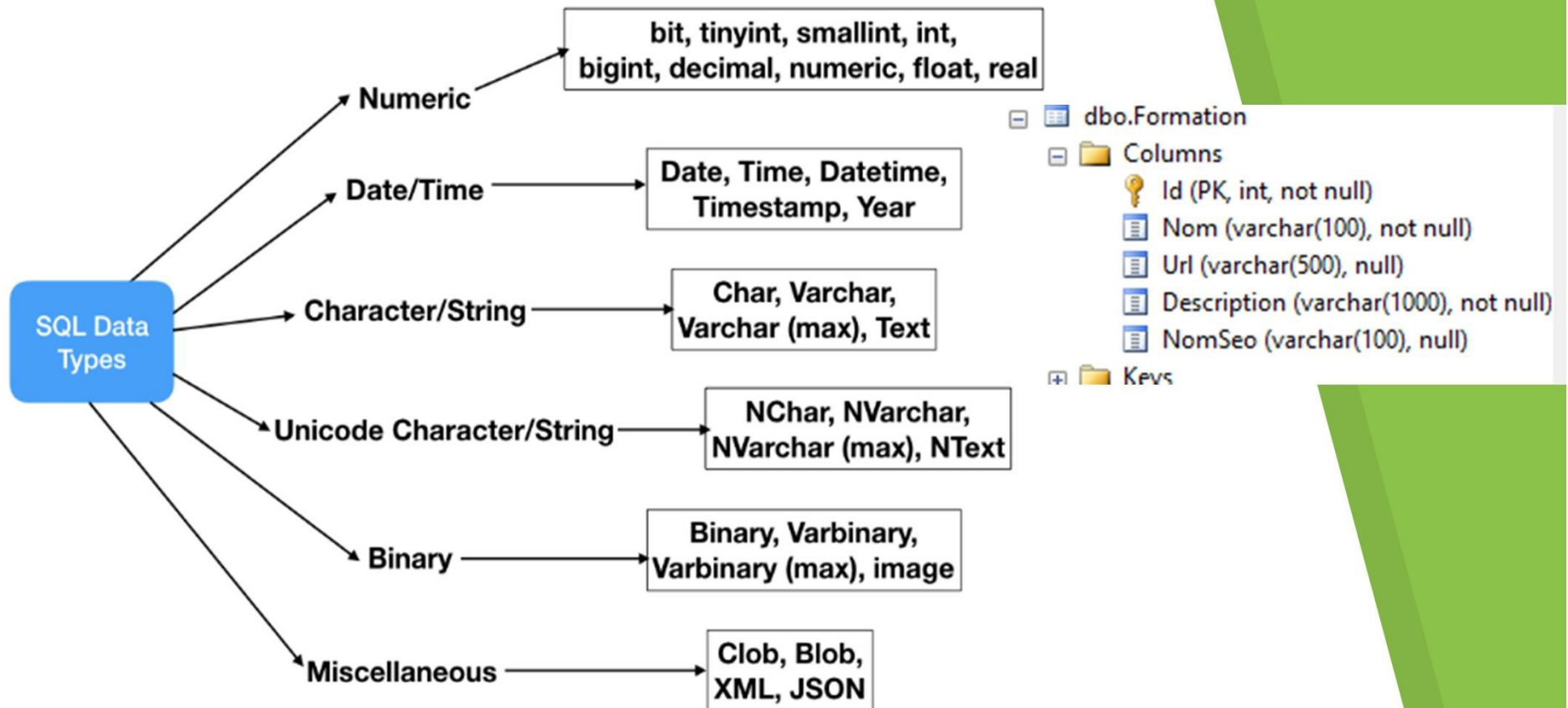


The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Explorer' pane shows the database structure, with 'dbo.Trainings' selected under the 'Tables' folder. On the right, the 'SQLQuery3.sql' window shows a query that selects the top 1000 rows from the 'Trainings' table, displaying columns 'SeoName', 'Name', 'Url', and 'Description'. Below the query, the 'Results' pane shows the first four rows of the query output.

```
SELECT TOP 1000 [SeoName]
, [Name]
, [Url]
, [Description]
FROM [TrainingReviews].[dbo].[Trainings]
```

	SeoName	Name
1	asp-net-core	How to build a professional website with
2	how-to-be-more-productive	How to be more productive ?
3	how-to-waste-money-when-investing	How to waste money when investing ?
4	perfect-gardener	The perfect gardener

Types de données



Contexte général

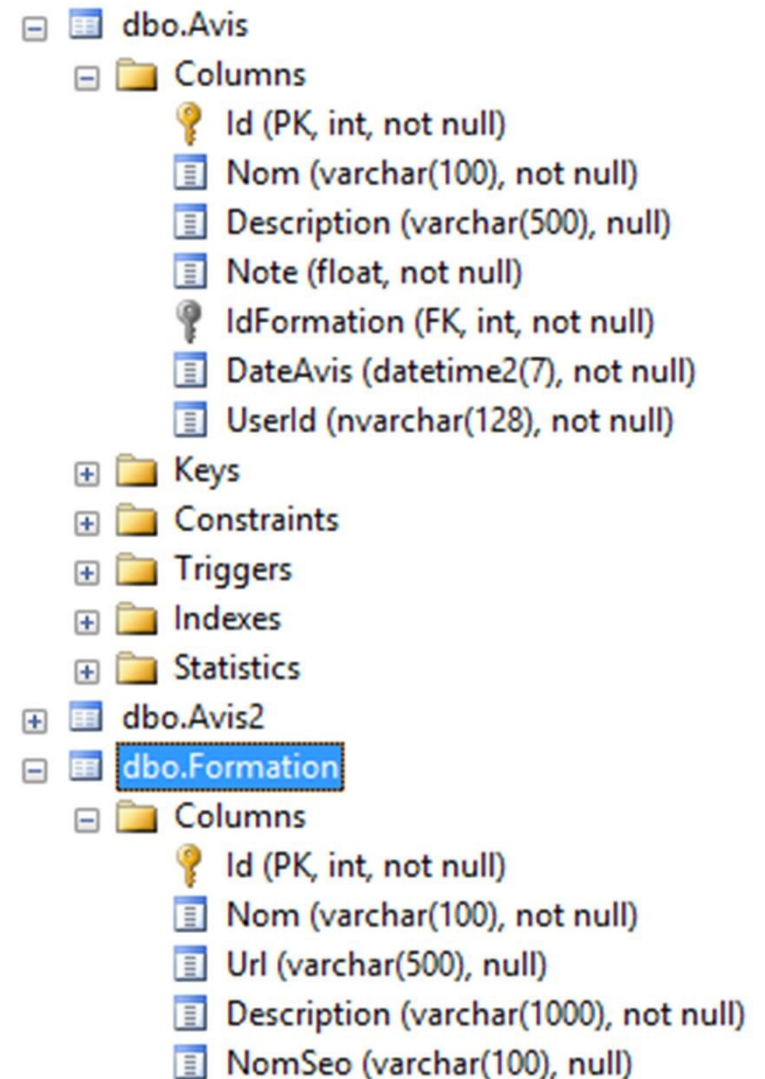
Base de données SQL



Qu'est-ce qu'une base de données relationnelle ?

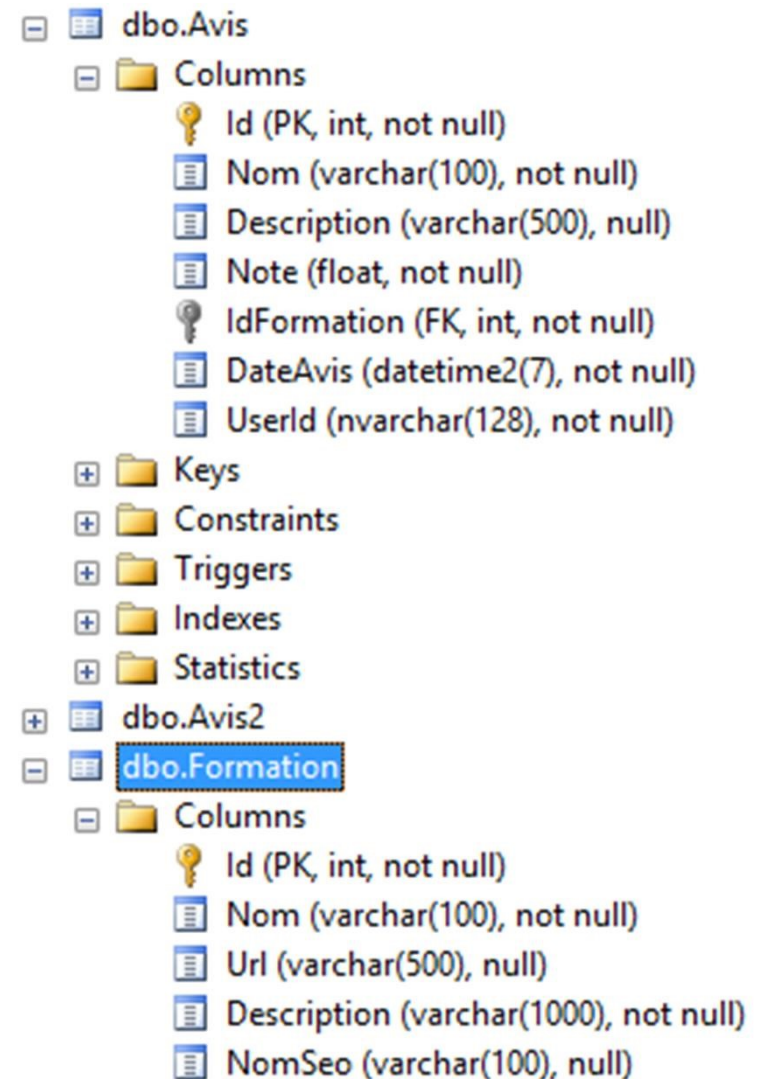
Qu'est-ce qu'une base de données relationnelle ?

► Une base de données structurée pour reconnaître les **relations** entre les **informations** stockées.



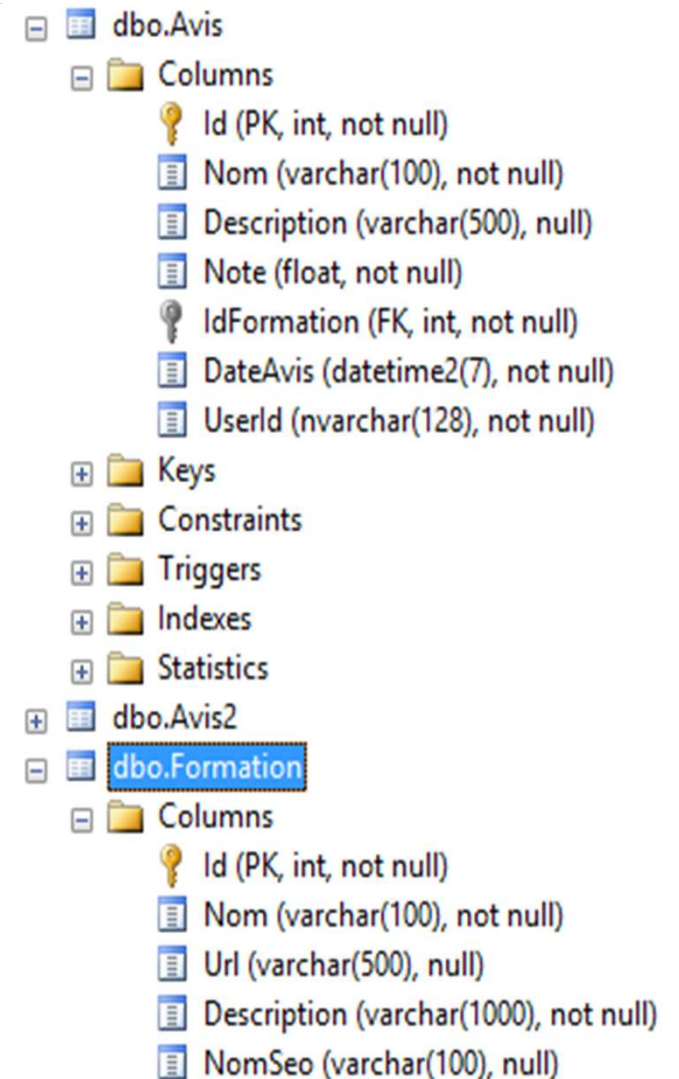
Primary Key

- La contrainte PRIMARY KEY identifie de manière unique chaque enregistrement d'une table.
- Les clés primaires doivent contenir des valeurs UNIQUE et ne peuvent pas contenir de valeurs NULL.
- Une table ne peut contenir qu'UNE SEULE clé primaire; et dans la table, cette clé primaire peut consister en une ou plusieurs colonnes (champs)



Foreign Key

- Une clé étrangère est une clé utilisée pour relier deux tables.
- Une clé étrangère est un champ (ou une collection de champs) dans une table qui fait référence à la clé primaire dans une autre table.



Index

► Les index sont utilisés pour extraire très rapidement les données de la base de données. Les utilisateurs ne peuvent pas voir les index, ils sont simplement utilisés pour accélérer les recherches / requêtes.

```
CREATE INDEX index_name  
ON table_name (column1, column2, ...);
```

Procédure stockée

```
CREATE PROCEDURE SelectAllCustomers @City nvarchar(30)
AS
SELECT * FROM Customers WHERE City = @City
GO;
```

► Une procédure stockée est du code SQL préparé que vous pouvez enregistrer afin de pouvoir le réutiliser encore et encore.

```
EXEC SelectAllCustomers City = "London";
```

► Donc, si vous avez une requête SQL que vous écrivez encore et encore, enregistrez-la en tant que procédure stockée, puis appelez-la pour l'exécuter.

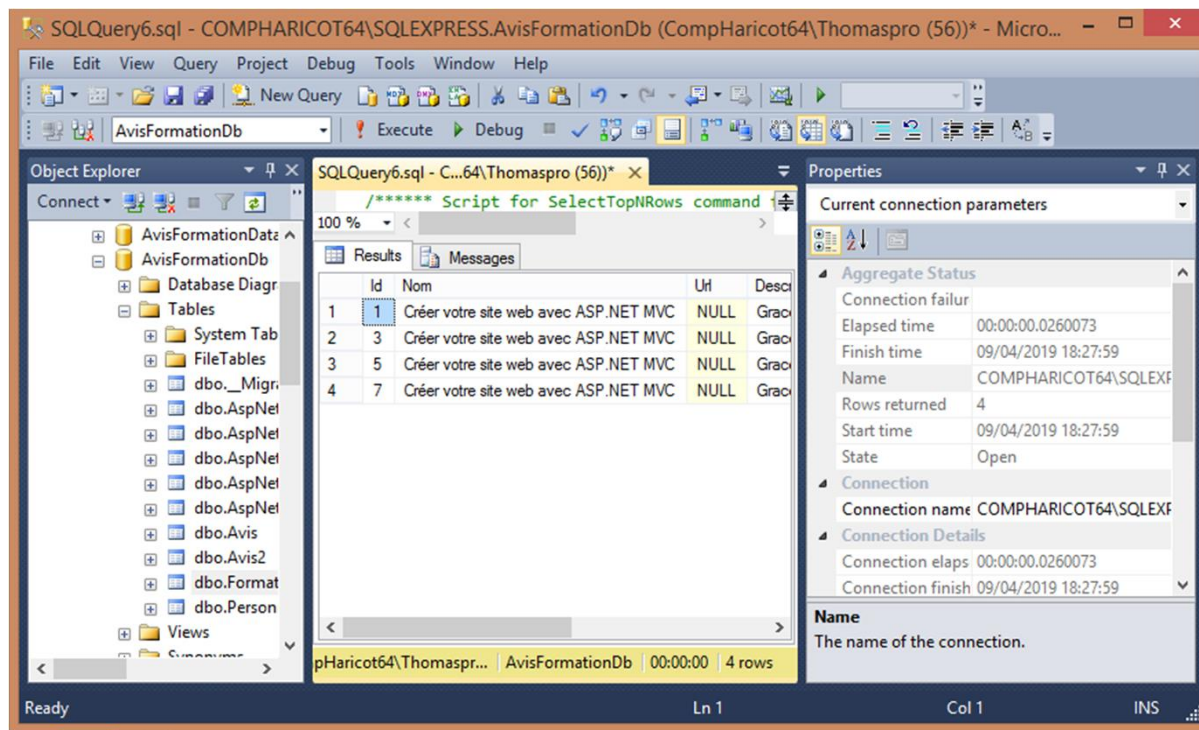
► Vous pouvez également transmettre des paramètres à une procédure stockée afin que celle-ci puisse agir en fonction de la ou

Qu'est-ce que SQL SERVER ?

Microsoft SQL Server est un Système de gestion de base de données (SGBD) relationnel et transactionnel.

Autres SGBD : Oracle, Sybase ...

SQL Server Management Studio





Apprendre les bases du SQL

Bases du langage SQL

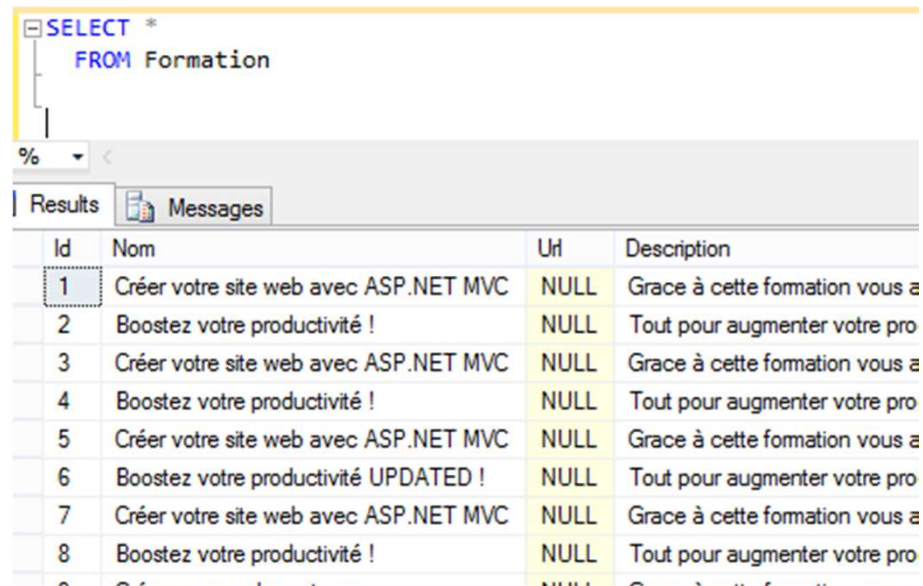
- **SELECT**
 - INSERT
 - UPDATE
 - DELETE
 - CREATE TABLE
 - DROP TABLE
-
- <https://www.w3schools.com/sql/>

SELECT

- Permet de lire des données en base

```
SELECT column1, column2, ...  
FROM table_name;
```

```
SELECT * FROM table_name;
```



The screenshot shows a SQL Server Enterprise Manager interface. At the top, a query window displays the SQL command: `SELECT * FROM Formation`. Below the query window, the 'Results' tab is active, showing a table with 4 columns: 'Id', 'Nom', 'Url', and 'Description'. The table contains 8 rows of data. The first row is selected, and the 'Id' column is highlighted. The 'Url' column for all rows contains the value 'NULL'.

Id	Nom	Url	Description
1	Créer votre site web avec ASP.NET MVC	NULL	Grace à cette formation vous a
2	Boostez votre productivité !	NULL	Tout pour augmenter votre pro
3	Créer votre site web avec ASP.NET MVC	NULL	Grace à cette formation vous a
4	Boostez votre productivité !	NULL	Tout pour augmenter votre pro
5	Créer votre site web avec ASP.NET MVC	NULL	Grace à cette formation vous a
6	Boostez votre productivité UPDATED !	NULL	Tout pour augmenter votre pro
7	Créer votre site web avec ASP.NET MVC	NULL	Grace à cette formation vous a
8	Boostez votre productivité !	NULL	Tout pour augmenter votre pro

SELECT DISTINCT

L'instruction SELECT DISTINCT est utilisée pour renvoyer uniquement des valeurs distinctes (différentes).

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

```
SELECT distinct nom, nomseo  
FROM Formation
```

%

Results Messages

nom	nomseo
Boostez votre mémoire !	formation-memoire
Boostez votre productivité !	formation-productivite
Boostez votre productivité UPDATED !	formation-productivite
Créer votre propre drone !	drone
Créer votre site web avec ASP.NET MVC	asp-net-mvc
Créez un superbe potager	potager
Triplez votre productivité !	formation-productivite-2

Clause WHERE

- Filtre les résultats à partir d'une condition spécifiée

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

```
SELECT *  
FROM Formation WHERE NomSeo = 'formation-productivite'
```

100 %

Results Messages

	Id	Nom	Url	Description	NomSeo
1	2	Boostez votre productivité !	NULL	Tout pour augmenter votre productivité. Devenez...	formation-productivite
2	4	Boostez votre productivité !	NULL	Tout pour augmenter votre productivité. Devenez...	formation-productivite
3	6	Boostez votre productivité UPDATED !	NULL	Tout pour augmenter votre productivité. Devenez...	formation-productivite
4	8	Boostez votre productivité !	NULL	Tout pour augmenter votre productivité. Devenez...	formation-productivite

AND, OR

- Avec WHERE on peut combiner AND ou OR pour complxifier notre conditio nde filtrage

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 AND condition3 ...;
```

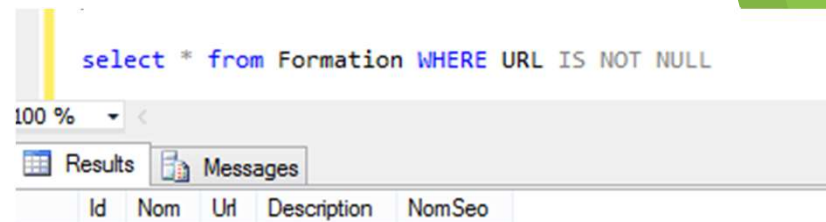
```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR condition3 ...;
```

Valeurs NULL

- NULL = aucune donnée dans la colonne

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```



The screenshot shows a SQL query execution interface. The query entered is `select * from Formation WHERE URL IS NOT NULL`. The interface includes a zoom level of 100%, tabs for 'Results' and 'Messages', and a table of results with columns: Id, Nom, Url, Description, and NomSeo.

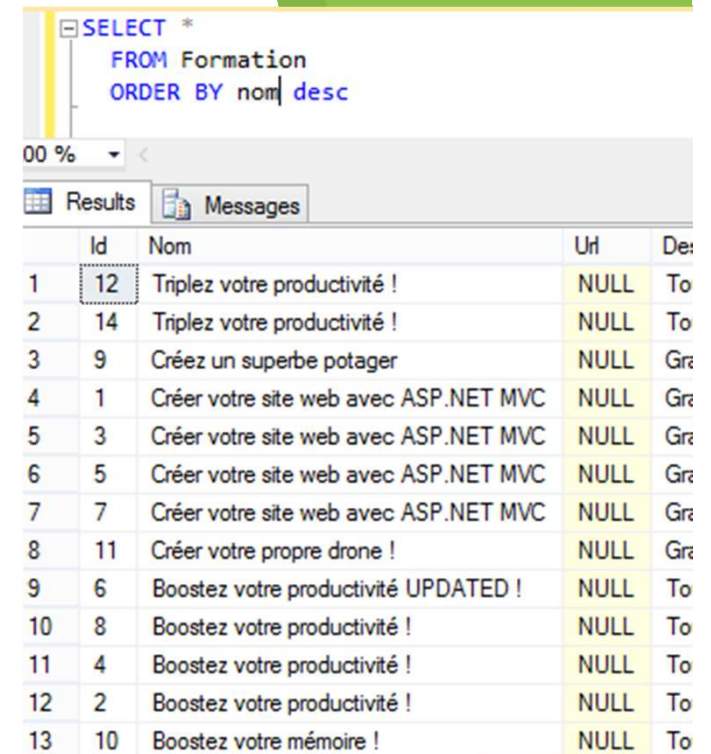
Id	Nom	Url	Description	NomSeo
----	-----	-----	-------------	--------

ORDER BY

Le mot clé ORDER BY permet de trier le jeu de résultats par ordre croissant ou décroissant.

ORDER BY Syntax

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL statement:

```
SELECT *  
FROM Formation  
ORDER BY nom desc
```

The results window displays a table with 5 columns: Id, Nom, Url, and De. The table contains 13 rows of data, sorted by the 'nom' column in descending order. The 'Id' column is highlighted in the first row.

	Id	Nom	Url	De
1	12	Triplez votre productivité !	NULL	To
2	14	Triplez votre productivité !	NULL	To
3	9	Créez un superbe potager	NULL	Gr
4	1	Créer votre site web avec ASP.NET MVC	NULL	Gr
5	3	Créer votre site web avec ASP.NET MVC	NULL	Gr
6	5	Créer votre site web avec ASP.NET MVC	NULL	Gr
7	7	Créer votre site web avec ASP.NET MVC	NULL	Gr
8	11	Créer votre propre drone !	NULL	Gr
9	6	Boostez votre productivité UPDATED !	NULL	To
10	8	Boostez votre productivité !	NULL	To
11	4	Boostez votre productivité !	NULL	To
12	2	Boostez votre productivité !	NULL	To
13	10	Boostez votre mémoire !	NULL	To

Sous requetes SELECT

- Il est tout à fait possible d'inclure des sous requetes :

```
-- Find the name of the company that placed order 10290.  
  
SELECT CompanyName  
FROM Customers  
WHERE CustomerID = (SELECT CustomerID  
                     FROM Orders  
                     WHERE OrderID = 10290);
```


Installation NorthWind

- Créer une base de données NorthWindDb
- Executer le script NorthWindScript.sql sur cette base

EXERCICE base NorthWind

1. Sélectionner tous les enregistrements de la table *Customers*
2. Sélectionner toutes les City de la table *Customers*
3. Sélectionner les différentes valeurs de Country de la table *Customers*
4. Sélectionner tous les enregistrements pour lesquels la colonne City a la valeur "Berlin".
5. Sélectionnez tous les enregistrements pour lesquels la colonne CustomerID a la valeur 32.

Exercice SELECT Suite

1. Sélectionnez tous les enregistrements pour lesquels la colonne City a la valeur 'Berlin' et la colonne PostalCode a la valeur 12209.
2. Sélectionnez tous les enregistrements pour lesquels la colonne City a la valeur 'Berlin', ainsi que ceux pour lesquels la colonne City a la valeur 'London'.
3. Sélectionnez tous les enregistrements de la table *Customers*, triez le résultat inversé par ordre alphabétique par la colonne City.
4. Sélectionnez tous les enregistrements de la table *Customers*, triez le résultat par ordre alphabétique, d'abord par la colonne Country, puis par la colonne City.

Bases du langage SQL

- SELECT
- **INSERT**
- UPDATE
- DELETE
- CREATE TABLE
- DROP TABLE

INSERT

- Sert à insérer des données en bases

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO dbo.Formation (Nom, Url, Description, NomSeo)  
VALUES ('ma formation', NULL, 'Grace à cette formation ...', 'ma-formation')  
GO
```

|

EXERCICE INSERT & NULL

- ▶ Insérez un nouvel enregistrement dans la table *Customers*.
- ▶ Sélectionnez tous les enregistrements des *Customers* où la colonne PostalCode est vide.
- ▶ Sélectionnez tous les enregistrements des *Customers* où la colonne PostalCode n'est pas vide

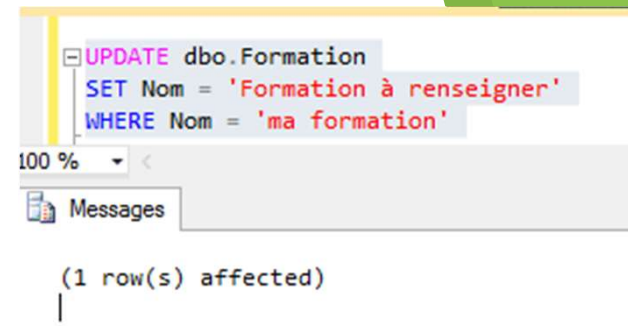
Bases du langage SQL

- SELECT
- INSERT
- **UPDATE**
- DELETE
- CREATE TABLE
- DROP TABLE

UPDATE

L'instruction UPDATE est utilisée pour modifier les enregistrements existants dans une table.

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```



The screenshot shows a SQL query execution window. The query is: `UPDATE dbo.Formation SET Nom = 'Formation à renseigner' WHERE Nom = 'ma formation'`. Below the query, the execution status is displayed: `(1 row(s) affected)`. The window has a 'Messages' tab selected, and the zoom level is set to 100%.

```
UPDATE dbo.Formation  
SET Nom = 'Formation à renseigner'  
WHERE Nom = 'ma formation'
```

100 % <

Messages

(1 row(s) affected)
|

Exercice Update

- Définissez la valeur des colonnes City sur "Oslo", mais uniquement sur celles où la colonne Country a la valeur "Norway".
- Mettez à jour la valeur City et la valeur Country pour le CustomerId=32 en 1 seule requête

Bases du langage SQL

- SELECT
- INSERT
- UPDATE
- **DELETE**
- CREATE TABLE
- DROP TABLE

DELETE

- L'instruction DELETE est utilisée pour supprimer des enregistrements existants dans une table.

```
DELETE FROM table_name WHERE condition;
```

```
DELETE FROM dbo.Formation  
WHERE Id = 15
```

100 % <

Messages

(1 row(s) affected)

Exercice Delete

- Supprimez tous les enregistrements de la table Customers où la valeur Country est 'Norway'.

CREATE TABLE

L'instruction CREATE TABLE permet de créer une nouvelle table dans une base de données.

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

```
CREATE TABLE dbo.Formation2  
(  
    Id          INT IDENTITY NOT NULL,  
    Nom         VARCHAR (100) NOT NULL,  
    Url         VARCHAR (500),  
    Description VARCHAR (1000) NOT NULL,  
    NomSeo      VARCHAR (100),  
    PRIMARY KEY (Id)  
);
```

DROP TABLE

- Supprime une table

```
DROP TABLE table_name;
```

```
DROP TABLE dbo.Formation2
```

Exercice Create /Drop table

- Écrivez l'instruction SQL correcte pour créer une nouvelle table appelée *Persons* avec les colonnes suivantes :
- Supprimez cette meme table

```
PersonID int,  
LastName varchar(255),  
FirstName varchar(255),  
Address varchar(255),  
City varchar(255)
```

SQL : fonctions principales

- **SELECT TOP**
- MIN and MAX
- COUNT, AVG, SUM
- LIKE
- IN

SELECT TOP

- La clause SELECT TOP est utilisée pour spécifier le nombre d'enregistrements à renvoyer.
- La clause SELECT TOP est utile sur les grandes tables contenant des milliers d'enregistrements. Le renvoi d'un grand nombre d'enregistrements peut avoir une incidence sur les performances.

```
SELECT TOP number | percent column_name(s)  
FROM table_name  
WHERE condition;
```

```
SELECT TOP 1000 [Id]  
    , [Nom]  
    , [Url]  
    , [Description]  
    , [NomSeo]  
FROM Formation
```

SQL : fonctions principales

- SELECT TOP
- **MIN and MAX**
- COUNT, AVG, SUM
- LIKE
- IN

MIN et MAX

- La fonction MIN () renvoie la plus petite valeur de la colonne sélectionnée.
- La fonction MAX () renvoie la plus grande valeur de la colonne sélectionnée.

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

SQL : fonctions principales

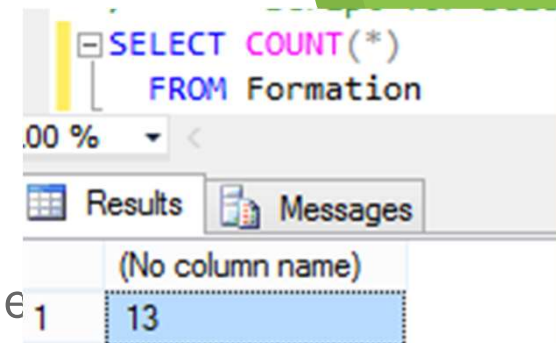
- SELECT TOP
- MIN and MAX
- **COUNT, AVG, SUM**
- LIKE
- IN

COUNT, AVG, SUM

- La fonction COUNT() renvoie le nombre de lignes correspondant à un critère spécifié.
- La fonction AVG() renvoie la valeur moyenne d'une colonne numérique.
- La fonction SUM() renvoie la somme totale d'une colonne numérique.

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```



The screenshot shows a SQL query editor with the query: `SELECT COUNT(*) FROM Formation`. Below the query, there are tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a single row with the value 13. The column header is '(No column name)'.

(No column name)
13

SQL : fonctions principales

- SELECT TOP
- MIN and MAX
- COUNT, AVG, SUM
- **LIKE**

WHERE + LIKE

- ▶ L'opérateur LIKE est utilisé dans une clause WHERE pour **rechercher un pattern spécifié dans une colonne**.
- ▶ Deux caractères génériques sont souvent utilisés avec l'opérateur LIKE:
 - % - Le signe de **pourcentage** représente zéro, un ou plusieurs caractères.
 - _ - Le trait de soulignement représente un seul caractère

LIKE Operator

Description

WHERE CustomerName LIKE 'a%'

Finds any values that start with "a"

WHERE CustomerName LIKE '%a'

Finds any values that end with "a"

WHERE CustomerName LIKE '%or%'

Finds any values that have "or" in any position

WHERE CustomerName LIKE '_r%'

Finds any values that have "r" in the second position

WHERE CustomerName LIKE 'a_%_ %'

Finds any values that start with "a" and are at least 3 characters in length

WHERE ContactName LIKE 'a%o'

Finds any values that start with "a" and ends with "o"

```
SELECT *  
FROM Formation where nom like '%MVC%'
```

100 %

Results Messages

	Id	Nom	Url	Descript
1	1	Cr��rer votre site web avec ASP.NET MVC	NULL	Grace ��
2	3	Cr��rer votre site web avec ASP.NET MVC	NULL	Grace ��
3	5	Cr��rer votre site web avec ASP.NET MVC	NULL	Grace ��
4	7	Cr��rer votre site web avec ASP.NET MVC	NULL	Grace ��

Exercice LIKE

1. Sélectionnez tous les enregistrements pour lesquels la valeur de la colonne City **commence** par la lettre "a". (table *Customers*)
2. Sélectionnez tous les enregistrements pour lesquels la valeur de la colonne City **fini**t par la lettre "a". (table *Customers*)
3. Sélectionnez tous les enregistrements pour lesquels la valeur de la colonne City **contient** la lettre "a". (table *Customers*)
4. Sélectionnez tous les enregistrements pour lesquels la valeur de la colonne City **commence par** la lettre « a » et finit par « b ».
5. Sélectionnez tous les enregistrements pour lesquels la valeur de la colonne City **NE commence PAS** par la lettre "a".
6. Sélectionnez tous les enregistrements où la première lettre de la City est un "a", un "c" ou un "s".

JOINTURES

- ▶ INNER JOIN
 - ▶ LEFT JOIN
 - ▶ RIGHT JOIN
- ▶ Une clause JOIN est utilisée pour combiner les lignes de deux ou plusieurs tables, en fonction d'une colonne liée entre elles.

OrderID	CustomerID	OrderDate
10308	2	1996-09-18
10309	37	1996-09-19
10310	77	1996-09-20

CustomerID	CustomerName	ContactName	Country
1	Alfreds Futterkiste	Maria Anders	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mexico

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

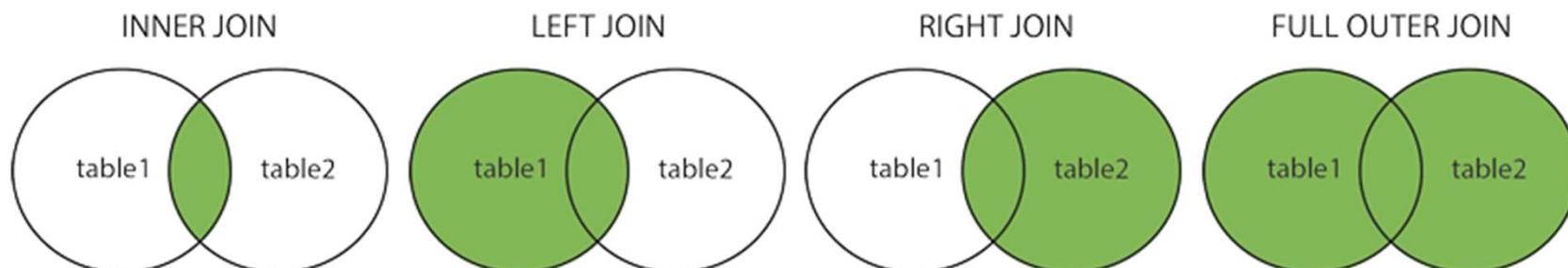


OrderID	CustomerName	OrderDate
10308	Ana Trujillo Emparedados y helados	9/18/1996
10365	Antonio Moreno Taquería	11/27/1996
10383	Around the Horn	12/16/1996
10355	Around the Horn	11/15/1996
10278	Berglunds snabbköp	8/12/1996



Types de jointures

- **(INNER)** JOIN: Retourne les enregistrements ayant des valeurs correspondantes dans les deux tables
- **LEFT** (OUTER) JOIN: Renvoie tous les enregistrements de la table de gauche et les enregistrements correspondants de la table de droite
- **RIGHT** (OUTER) JOIN: Renvoie tous les enregistrements de la table de droite et les enregistrements correspondants de la table de gauche.
- **FULL** (OUTER) JOIN: Renvoie tous les enregistrements lorsqu'il y a une correspondance dans la table gauche ou droite



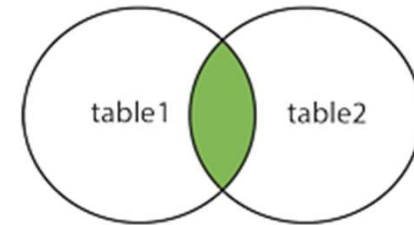
INNER JOIN

► Le mot clé INNER JOIN sélectionne les enregistrements dont les valeurs correspondent dans les deux tables.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDate
FROM Orders
INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;
```

INNER JOIN



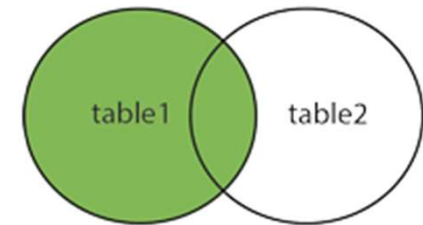
LEFT JOIN

► Le mot clé LEFT JOIN renvoie tous les enregistrements de la table de gauche (table1) et les enregistrements correspondants de la table de droite (table2). Le résultat est NULL du côté droit, s'il n'y a pas de correspondance.

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;
```

LEFT JOIN

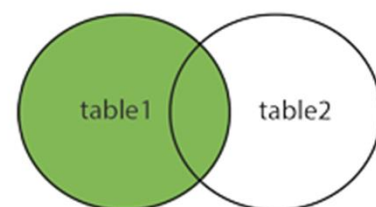


```

SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.CustomerName;

```

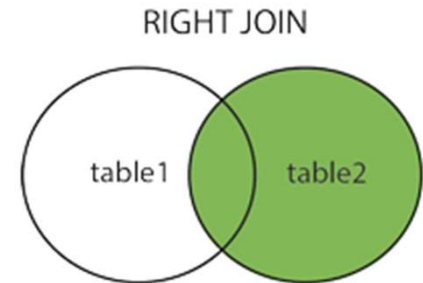
LEFT JOIN



CustomerName	OrderID
Alfreds Futterkiste	<i>null</i>
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
Around the Horn	10355
Around the Horn	10383
B's Beverages	10289
Berglunds snabbköp	10278
Berglunds snabbköp	10280
Berglunds snabbköp	10384
Blauer See Delikatessen	<i>null</i>

RIGHT JOIN

► Le mot clé RIGHT JOIN renvoie tous les enregistrements de la table de gauche (table1) et les enregistrements correspondants de la table de droite (table2). Le résultat est NULL du côté gauche, s'il n'y a pas de correspondance.

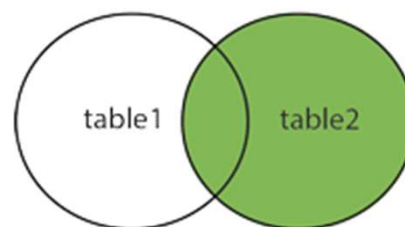


```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
FROM Orders
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
ORDER BY Orders.OrderID;
```

```
SELECT Orders.OrderID, Employees.LastName, Employees.FirstName
FROM Orders
RIGHT JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
ORDER BY Orders.OrderID;
```

RIGHT JOIN



OrderID	LastName	FirstName
	West	Adam
10248	Buchanan	Steven
10249	Suyama	Michael
10250	Peacock	Margaret
10251	Leverling	Janet
10252	Peacock	Margaret
10253	Leverling	Janet
10254	Buchanan	Steven

FULL OUTER JOIN

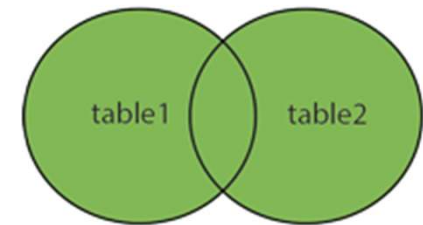
```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

► Le mot clé FULL OUTER JOIN renvoie tous les enregistrements lorsqu'il existe une correspondance dans les enregistrements de la table gauche (table1) ou droite (table2).

► Remarque: FULL OUTER JOIN peut potentiellement renvoyer de très grands ensembles de résultats!

► Astuce: FULL OUTER JOIN et FULL JOIN sont identiques.

FULL OUTER JOIN



```
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID
ORDER BY Customers.CustomerName;
```

CustomerName	OrderID
Alfreds Futterkiste	
Ana Trujillo Emparedados y helados	10308
Antonio Moreno Taquería	10365
	10382
	10351

Exercice JOIN

- Sélectionnez tous les noms de *Products* et leurs noms de *Categories*
- Sélectionnez tous les noms de *Products*, prix unitaires et la région du *Suppliers* qui n'a pas de fournisseurs des USA.
- Liste les *Employees* et le nom de leur responsable, sans inclure les *Employees* qui n'ont personne à qui faire rapport.

GROUP BY

► L'instruction GROUP BY est souvent utilisée avec des fonctions d'agrégation (COUNT, MAX, MIN, SUM, AVG) pour regrouper l'ensemble de résultats en une ou plusieurs colonnes.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

COUNT(CustomerID)	Country
3	Argentina
2	Austria
2	Belgium
9	Brazil
3	Canada

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

GROUP BY & HAVING

► La clause HAVING a été ajoutée à SQL car le mot clé WHERE n'a pas pu être utilisé avec des fonctions d'agrégat.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5;
```

Exercice Group by

- Indiquez le nombre de *Customers* dans chaque Country.
- Indiquez le nombre de *Customers* dans chaque Country, classés par Country avec le plus grand nombre de *Customers* en premier.

Merci de votre attention !