




Traitement Numérique des Données

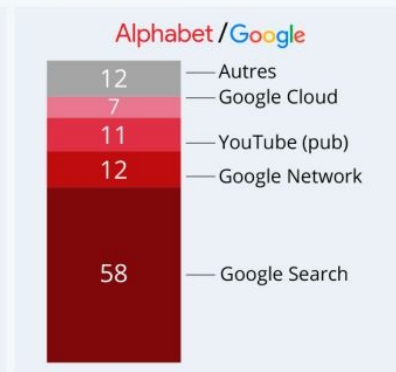
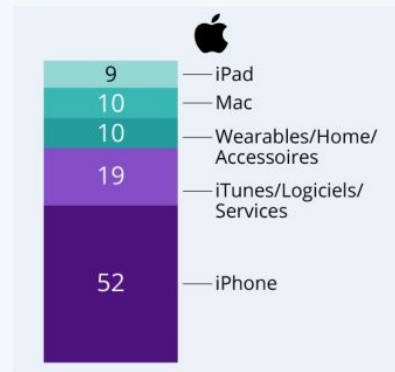
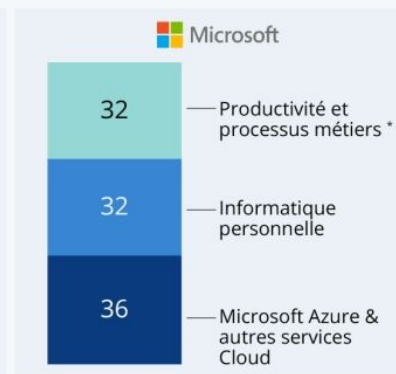


RAPPEL

1. Définition des données
 2. Base de données
 3. Modèle
 4. Processus
- 

Le cloud – big data

Part des segments d'activités et produits dans le chiffre d'affaires des entreprises en 2021, en %



* dont Office, LinkedIn et Dynamics

Sources : rapports des entreprises





CM part 2


1. Création d'une base de données
 - a. Shéma
 - b. modélisation
 2. SQL par l'exemple
 3. VB avec excel
- 

TABLE OF CONTENTS

01

Modélisation

02

Base de données

03

SQL

04

VB pour excel



Typage des données

SECURITE



EXACTITUDE



PERFORMANCE

Exemple d'un formulaire d'inscription

Congress Pass - Participant 1/1



Civilité *	Pays *
▼	France ▼
Prénom *	Département *
	▼
Nom *	Activité *
	▼
Société *	Centre d'intérêts
	▼
Email (pour recevoir votre badge) *	
Mobile	



01

Modélisation



“Un professeur enseigne une ou plusieurs matière.s pour une période donnée”

Modélisation

Professeur

Enseigne

Matière



Modélisation

Professeur

id

Nom

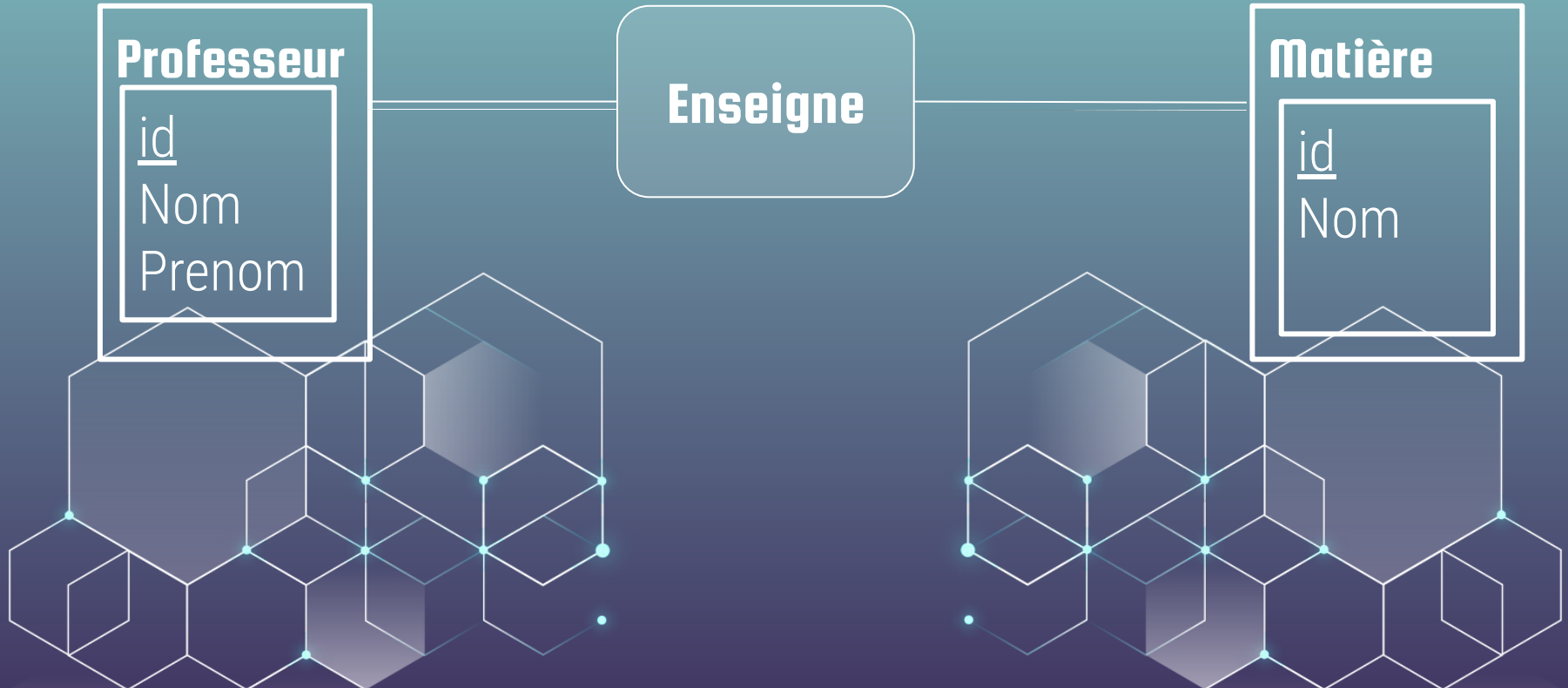
Prenom

Enseigne

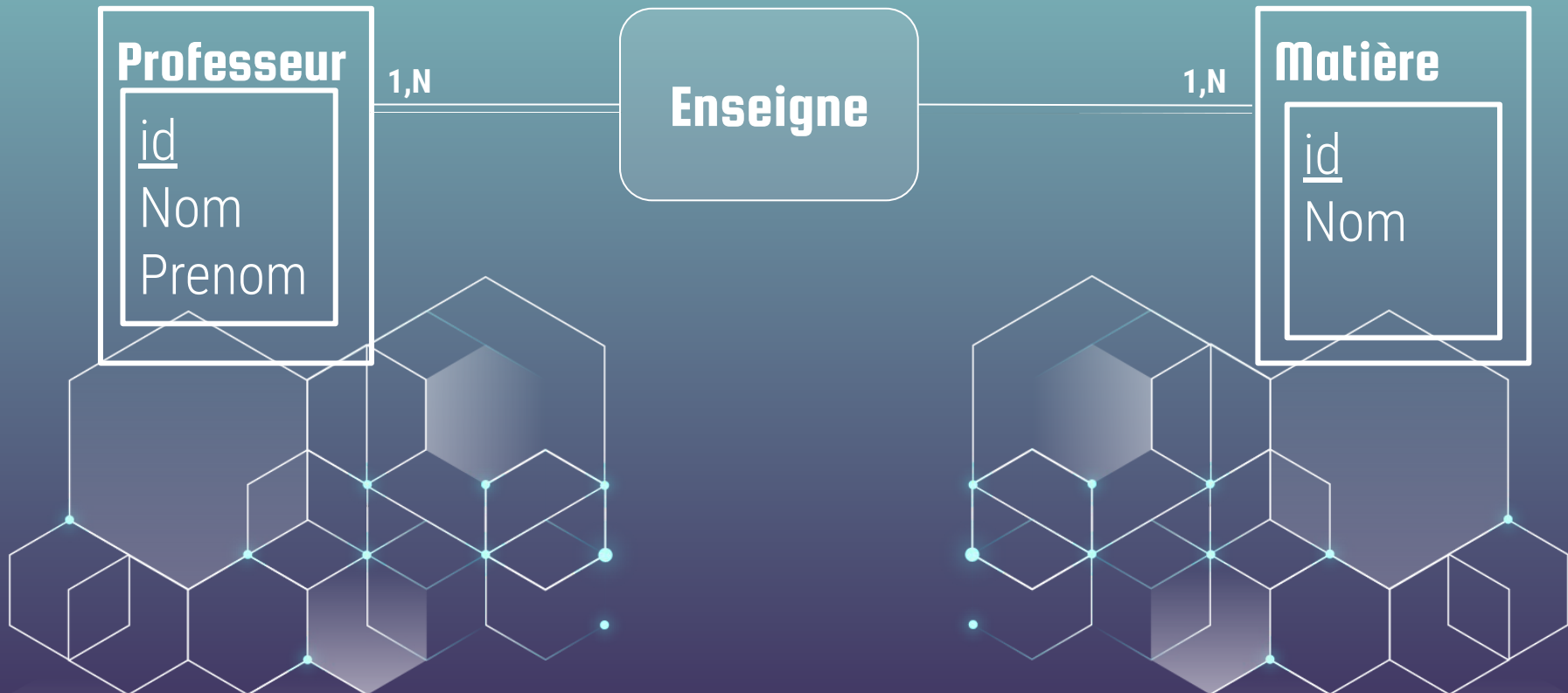
Matière



Modélisation



Modélisation



Rappel cardinalités

1. One to One (1:1) :

- Une instance d'une entité est associée à une et une seule instance de l'autre entité, et vice versa. Par exemple, un étudiant est associé à un seul numéro d'identification unique, et chaque numéro d'identification unique est associé à un seul étudiant.

2. One to Many (1:N) :

- Une instance d'une entité est associée à plusieurs instances de l'autre entité, mais chaque instance de l'autre entité est associée à une seule instance de la première entité. Par exemple, un professeur peut avoir plusieurs étudiants, mais chaque étudiant n'a qu'un seul professeur.

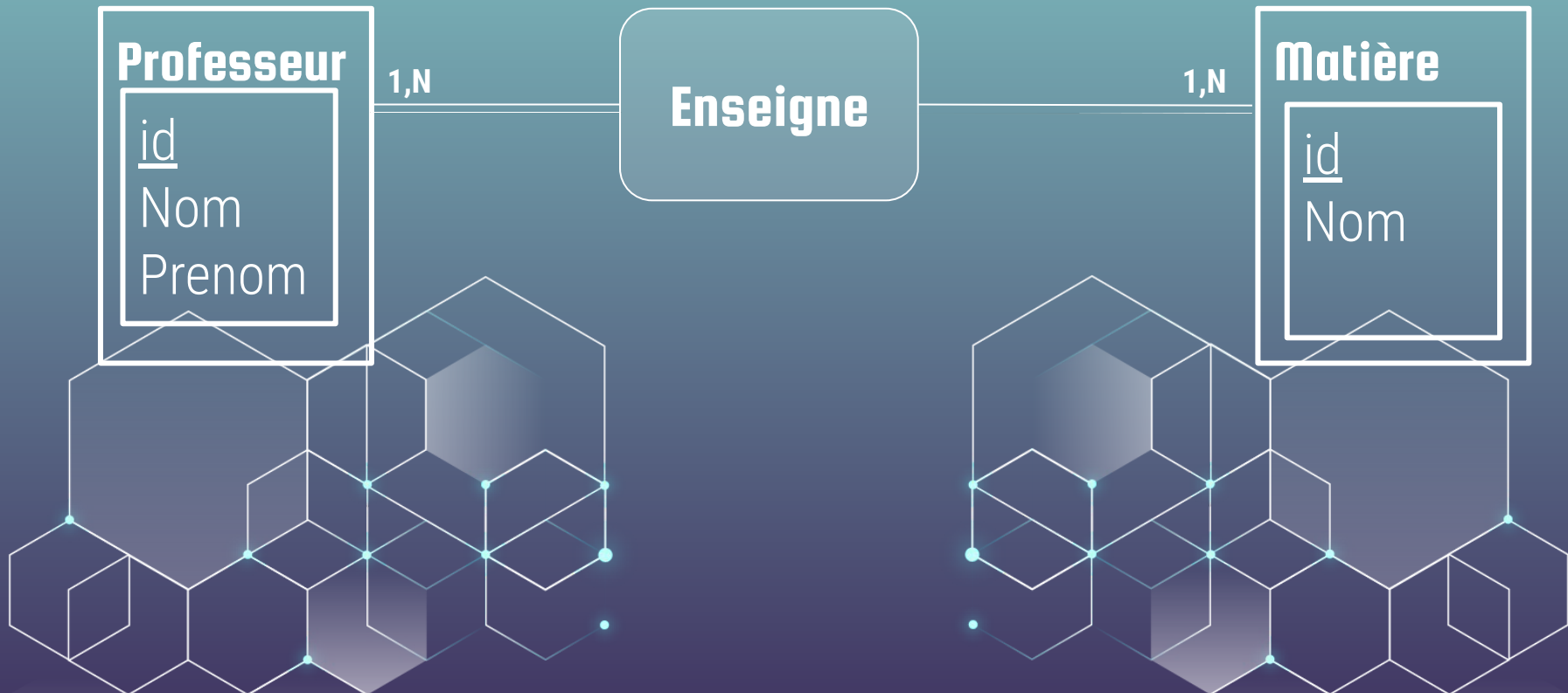
1. Many to One (N:1) :

- Plusieurs instances d'une entité sont associées à une seule instance de l'autre entité. Par exemple, plusieurs étudiants peuvent être associés à une seule université.

2. Many to Many (N:N) :

- Plusieurs instances d'une entité peuvent être associées à plusieurs instances de l'autre entité. Par exemple, de nombreux étudiants peuvent être inscrits à plusieurs cours, et chaque cours peut avoir plusieurs étudiants.

Modélisation





Point sur le modèle

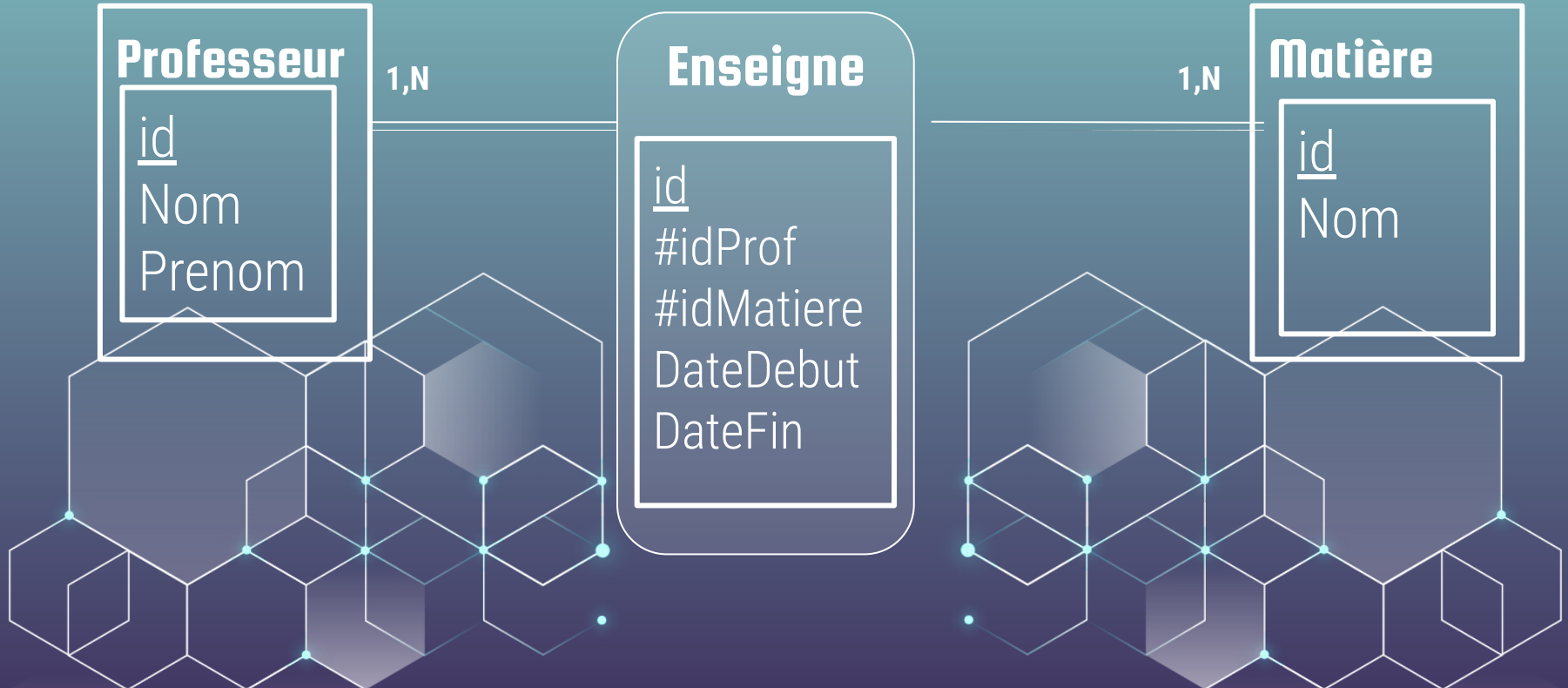
- 2 entités
- 1 association
- Des cardinalités

Un professeur enseigne une ou plusieurs matière.s

Une matière est enseignée par un ou plusieurs professeur.s



Modélisation





L'association enseigne

L'association enseigne va jouer le rôle de relation entre Professeur et Matière



Réflexion

Enseigne (id, #idProf, #idMatiere, DateDebut, DateFin)

Id : clé primaire
#idProf : clé étrangère
#idMatiere : clé étrangère

Variante :

Enseigne (#idProf, #idMatiere, DateDebut, DateFin)
Pas de clé primaire





02

database



Modèle conceptuel

→

Modèle physique

Les entités et les associations
deviennent des tableaux
“relationnels”



Concrètement cela donne quoi ?



Modèle Relationnel

v	cmr504	professeur
🔑		id : int(11)
📄		nom : varchar(60)
📄		prenom : varchar(60)

v	cmr504	enseigne
🔑		id : int(11)
#		professeur : int(11)
#		matiere : int(11)
📅		dateDebut : date
📅		dateFin : date

v	cmr504	matiere
🔑		id : int(11)
📄		nom : varchar(50)





03

SQL

SQL

Clause	Rôle	Exemple
SELECT	Permet d'exprimer et d'exécuter des requêtes ensemblistes sur la base de données	<pre>SELECT essai_comment FROM essai WHERE ref_essai = 8419 ;</pre>
INSERT INTO	Permet d'ajouter des lignes ou des tuples dans une table de la base de données	<pre>INSERT INTO essai ('ref_essai', 'essai_libelle') VALUES (12, 'ajout de ligne') ;</pre>
DELETE FROM	Supprime des lignes dans une table de la base de données	<pre>DELETE FROM essai WHERE essai_libelle CONTAINS '%lbc' ;</pre>
UPDATE	Met à jour des lignes existantes dans une table de la base de données	<pre>UPDATE essai SET essai_libelle = 'texte du jeu' WHERE ref_essai = 12 ;</pre>
<Window function> OVER	Permet d'effectuer une opération de fenêtrage. Disponible uniquement à partir de la norme SQL – ISO 2003.	<pre>SUM(montant) OVER (PARTITION BY region, categorie, ORDER BY date_vente DESC)</pre>




SELECT * FROM `professeur`

id	nom	prenom
1	einstein	albert
2	poincarré	henry
3	descartes	rené
4	ABRAGAM	Anatole
5	Amboise	Jacques
6	BÉRAULT	Lucie
7	Preynat	Jean-Jacques
8	BLANC	Vincent
9	BLONDEL	Françoise





```
SELECT nom FROM `professeur`
```




nom
einstein
poincaré
descartes
ABRAGAM
Amboise
BÉRAULT
Preynat
BLANC
BLONDEL



```
sql
```

```
SELECT matiere.nom  
FROM matiere  
LEFT JOIN enseigne ON matiere.id = enseigne.matiere  
WHERE enseigne.matiere IS NULL;
```

Explication : Cette requête utilise une jointure gauche pour trouver toutes les matières qui n'ont pas été enseignées (c'est-à-dire qu'il n'y a pas d'entrée correspondante dans la table enseigne).



sql

```
SELECT matiere.nom  
FROM matiere  
JOIN enseigne ON matiere.id = enseigne.matiere  
JOIN professeur ON enseigne.proesseur = professeur.id  
WHERE professeur.nom = 'ABRAGAM';
```

1. Liste des matières enseignées par un professeur donné :

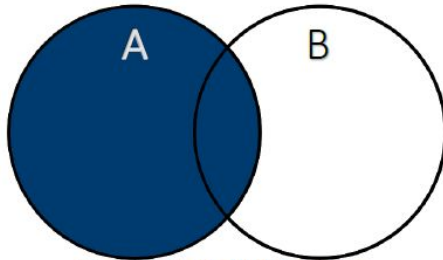
Explication : Cette requête liste toutes les matières enseignées par le professeur nommé ABRAGAM. Elle utilise des jointures entre les tables enseigne, matiere et professeur basées sur les identifiants correspondants.

sql

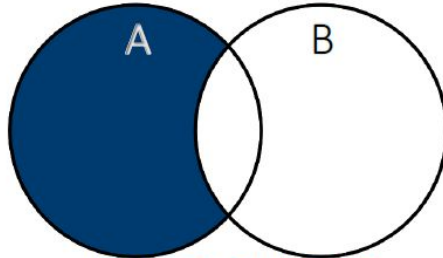
```
SELECT professeur.nom, professeur.prenom  
FROM professeur  
JOIN enseigne ON professeur.id = enseigne.professeur  
JOIN matiere ON enseigne.matiere = matiere.id  
WHERE matiere.nom = 'Mathématiques';
```

1. Liste des professeurs qui enseignent "Mathématiques" :

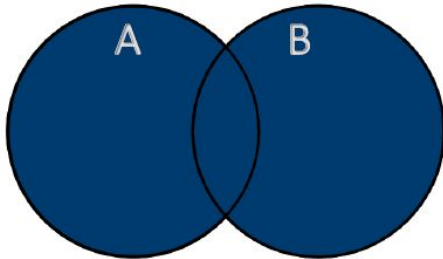
Explication : Cette requête liste tous les professeurs qui enseignent les Mathématiques, en utilisant des jointures entre les tables enseigne, matiere et professeur.



LEFT INCLUSIVE

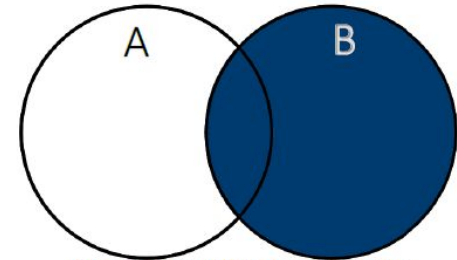


LEFT EXCLUSIVE

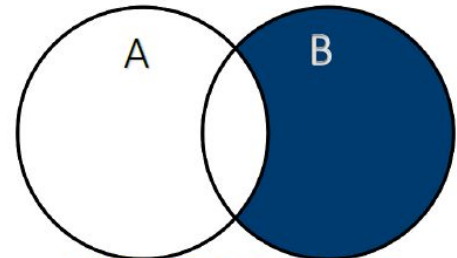


FULL OUTER INCLUSIVE

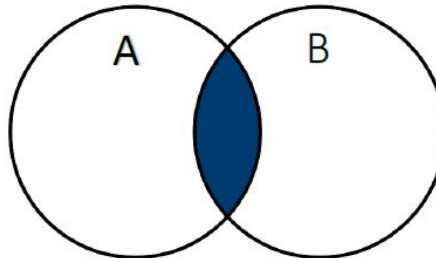
SQL JOINS	
LEFT INCLUSIVE SELECT [Select List] FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key	RIGHT INCLUSIVE SELECT [Select List] FROM TableA A RIGHT OUTER JOIN TableB B ON A.Key= B.Key
LEFT EXCLUSIVE SELECT [Select List] FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key WHERE B.Key IS NULL	RIGHT EXCLUSIVE SELECT [Select List] FROM TableA A LEFT OUTER JOIN TableB B ON A.Key= B.Key WHERE A.Key IS NULL
FULL OUTER INCLUSIVE SELECT [Select List] FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key	FULL OUTER EXCLUSIVE SELECT [Select List] FROM TableA A FULL OUTER JOIN TableB B ON A.Key = B.Key WHERE A.Key IS NULL OR B.Key IS NULL
INNER JOIN SELECT [Select List] FROM TableA A INNER JOIN TableB B ON A.Key = B.Key	



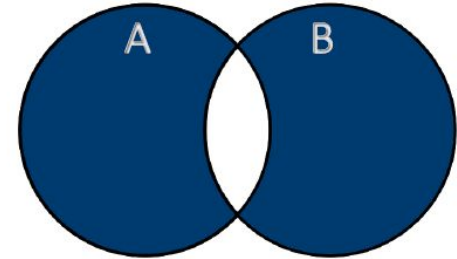
RIGHT INCLUSIVE



RIGHT EXCLUSIVE



INNER JOIN



FULL OUTER EXCLUSIVE



04

Visual basic

Définition

Définition


Visual Basic (VB) est un environnement de programmation développé par Microsoft qui permet aux développeurs de créer des applications pour Windows.





Interface utilisateur


Visual Basic propose une interface utilisateur graphique (GUI) conviviale qui permet aux développeurs de créer des interfaces utilisateur en glisser-déposer. Cela rend la création d'interfaces utilisateur assez facile, même pour les débutants.





Evenementiel


VB est axé sur l'événement, ce qui signifie qu'il réagit aux actions de l'utilisateur, comme les clics de souris ou les touches du clavier. Les programmeurs peuvent écrire du code pour répondre à ces événements et créer des applications interactives.





Accessibilité

VB est conçu pour être accessible aux débutants tout en offrant suffisamment de puissance aux développeurs expérimentés. Il fournit de nombreuses fonctionnalités et outils intégrés pour faciliter le développement d'applications.





Interopérabilité

Les applications VB peuvent interagir avec des composants et des bibliothèques créés dans d'autres langages et environnements, grâce à la compatibilité COM (Component Object Model) et .NET.



Microsoft Visual Basic pour Applications - Excelformation.fr - Formation VBA.xlsm - [Module8 (Code)]

Fichier Edition Affichage Insertion Format Débogage Exécution Outils Compléments Fenêtre ?

Li 15, Col 1

Projet - VBAProject

- EXCELFORMATION_barreOutils (ExcelFr
- EXCELFORMATION_exporterVersPower
- VBAProject (Excelformation.fr - Forma
 - Microsoft Excel Objets
 - Feuil1 (01-Introduction)
 - Feuil10 (07-Les fonctions)
 - Feuil11 (08-Lancer macro)
 - Feuil12 (09-VBE)
 - Feuil2 (02-Activer le menu Développe
 - Feuil3 (03-Enregistreur de macro)
 - Feuil4 (04-Analyse d'une macro)
 - Feuil5 (05-Les variables)
 - Feuil6 (05.1-Les variables)
 - Feuil7 (06-Les procédures)
 - Feuil8 (05.2-La portée)
 - Feuil9 (05.3-Manipuler)

Propriétés - Module8

Module8 Module

Alphabétique Par catégorie

(Name) Module8

(Général)

utiliserUneFonction

```
Sub maMacroDeTest(nom, message)
    MsgBox "Bonjour " & nom & " " & message
End Sub

Sub appelerLaMacroDeTest()
    Call maMacroDeTest(message:="Comment vas-tu ?", nom:="excelformation.fr")
End Sub

Function utiliserUneFonction()
    appelerLaMacroDeTest
End Function
```



Démonstration

vba

```
Sub MultiplierEtAfficher()  
    Dim valeur As Double  
    valeur = Sheets("Feuil1").Range("A7").Value * 4  
    Sheets("Feuil1").Range("B9").Value = valeur  
End Sub
```


situation



modèle



BD

Excel (csv, ..)


Vie réelle
Problématique
Optimisation

Recueil des données
Logiciel ERP/PGI
Extraction



API

L'API, qui signifie "Interface de Programmation d'Application", est un ensemble de règles et de spécifications qu'une application ou un logiciel peut suivre pour accéder et utiliser les fonctions et les services d'une autre application ou d'un autre logiciel. L'API permet donc à différents logiciels de communiquer entre eux.



End User with
Browser



Request



Response



API



Server Back-end
System



Customer

Make the
Order



Delivery of
order



Waiter

Take the
Order

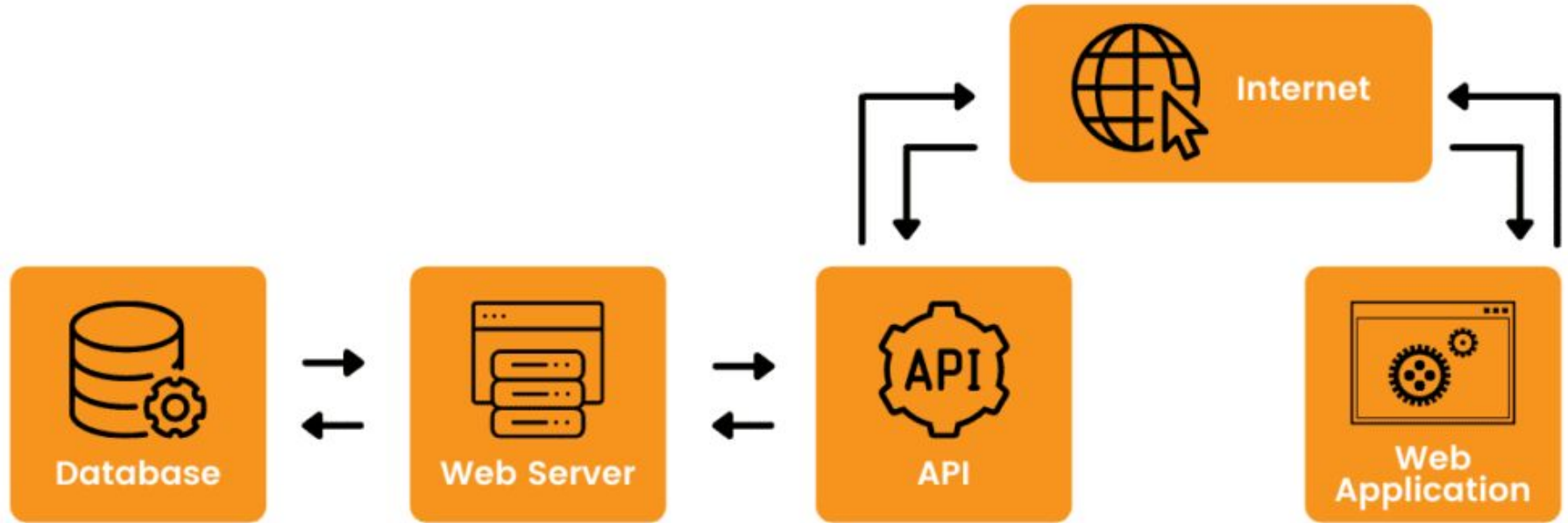


Bringing
from Kitchen



Chef

What is an **API**?





Merci de votre attention