

# Operations

Faire **vivre** la base de données : **créer**,  
**lire**, **remplacer**, **supprimer** des  
informations



Compétence demandée :  
**Comprendre les opérations de  
groupe et de jointure**

Pour vivre de l'information (base de données, REST API, SOAP), il faut des opérations basiques : les opérations  
CRUD



FRANCAIS	ANGLAIS
Créer	
Lire	
Remplacer	
Supprimer	

FRANCAIS	ANGLAIS
Créer	CREATE
Lire	READ
Remplacer	UPDATE
Supprimer	DELETE

FRANCAIS	ANGLAIS
Créer	CREATE
Lire	READ
Remplacer	UPDATE
Supprimer	DELETE

FRANCAIS	ANGLAIS
Créer	CREATE
Lire	READ
Remplacer	UPDATE
Supprimer	DELETE

Opérations **CRUD**



# Opérations **CRUD** pour les lignes

FRANCAIS	ANGLAIS	SQL
Créer	<b>C</b> REATE	<b>I</b> NSERT
Lire	<b>R</b> EAD	<b>S</b> ELECT
Remplacer	<b>U</b> PPDATE	<b>U</b> PPDATE
Supprimer	<b>D</b> ELETE	<b>D</b> ELETE



1. OPERATION DE GROUPE
2. OPERATION DE JOINTURE

# OPERATION DE GROUPE

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0
7	2020-05-23	20.0

GROUPE #1  
date : 2020-05-21

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0

GROUPE #2  
date : 2020-05-22

id	date	montant
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0

GROUPE #3  
date : 2020-05-23

id	date	montant
7	2020-05-23	20.0

GROUPE #1  
date : 2020-05-21

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0

GROUPE #2  
date : 2020-05-22

id	date	montant
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0

GROUPE #3  
date : 2020-05-23

id	date	montant
7	2020-05-23	20.0

GROUPE #1  
date : 2020-05-21  
date, SUM(montant)  
2020-05-21, 180.0

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0

GROUPE #2  
date : 2020-05-22

id	date	montant
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0

GROUPE #3  
date : 2020-05-23

id	date	montant
7	2020-05-23	20.0



GROUPE #1  
date : 2020-05-21  
date, SUM(montant)  
2020-05-21, 180.0

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0

GROUPE #2  
date : 2020-05-22  
date, SUM(montant)  
2020-05-22, 50.0

id	date	montant
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0

GROUPE #3  
date : 2020-05-23

id	date	montant
7	2020-05-23	20.0

GROUPE #1  
date : 2020-05-21  
date, SUM(montant)  
2020-05-21, 180.0

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0

GROUPE #2  
date : 2020-05-22  
date, SUM(montant)  
2020-05-22, 50.0

id	date	montant
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0

GROUPE #3  
date : 2020-05-23  
date, SUM(montant)  
2020-05-23, 20.0

id	date	montant
7	2020-05-23	20.0

date	SUM(montant)
2020-05-21	180.0
2020-05-22	50.0
2020-05-23	20.0

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0
7	2020-05-23	20.0

date	SUM(montant)
2020-05-21	180.0
2020-05-22	50.0
2020-05-23	20.0

id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0
7	2020-05-23	20.0

date	SUM(montant)
2020-05-21	180.0
2020-05-22	50.0
2020-05-23	20.0

Nous venons de **GROUPER** les lignes  
**PAR** date en faisant **SUM** du montant



id	date	montant
1	2020-05-21	30.0
2	2020-05-21	60.0
3	2020-05-21	90.0
4	2020-05-22	10.0
5	2020-05-22	10.0
6	2020-05-22	30.0
7	2020-05-23	20.0

*fusion ???*  
*après*

date	SUM(montant)
2020-05-21	180.0
2020-05-22	50.0
2020-05-23	20.0

Nous venons de **GROUPE** les lignes  
**PAR** date en faisant **SUM** du montant

**GROUPE** permet de créer une table avec 1 ligne par groupe.

La table est **divisé en groupes** (**par** une colonne, par ex: date) dont les lignes ont été **rassemblées** (par une **fonction d'agrégat**).





# OPERATION DE JOINTURE

proprietaires

id	nom	prenom	age	poids
1		Michael		
2				
3				
4		Sophie		
5				
6				
7				
8				

restaurants

id	nom	adresse	note	proprietaire_id
1	Raman			1
2				
3	Le ciel			4
4	Raimond			4
5	Bistro Toc			4
6				
7	Affine			1
8	Colin			1

proprietaires

id	nom	prenom	age	poids
1		Michael		
2				
3				
4		Sophie		
5				
6				
7				
8				

restaurants

id	nom	adresse	note	proprietaire_id
1	Raman			1
2				
3	Le ciel			4
4	Raimond			4
5	Bistro Toc			4
6				
7	Affine			1
8	Colin			1

TABLE

JOINTURE



TABLE

TABLE

Nous allons de FAIRE LA JOINTURE (LES COLLER)

proprietaires

restaurants

Nous allons de FAIRE LA JOINTURE (LES COLLER)

proprietaires

restaurants

id	nom	prenom	age	poids	id	nom	adresse	note	proprietaire_id
1		Michael			1	Raman			1
					7	Affine			1
					8	Colin			1

Nous allons de FAIRE LA JOINTURE (LES COLLER)

proprietaires

restaurants

id	nom	prenom	age	poids	id	nom	adresse	note	proprietaire_id
1		Michael			1	Raman			1
					7	Affine			1
					8	Colin			1

id	nom	prenom	age	poids	id	nom	adresse	note	proprietaire_id
4		Sophie			3	Le ciel			4
					4	Raimond			4
					5	Bistro Toc			4



Nous allons de FAIRE LA JOINTURE (LES COLLER)

proprietaires

restaurants

id	nom	prenom	age	poids	id	nom	adresse	note	proprietaire_id
1		Michael			1	Raman			1
1		Michael			7	Affine			1
1		Michael			8	Colin			1

id	nom	prenom	age	poids	id	nom	adresse	note	proprietaire_id
4		Sophie			3	Le ciel			4
4		Sophie			4	Raimond			4
4		Sophie			5	Bistro Toc			4

Nous allons de FAIRE LA JOINTURE (LES COLLER)

proprietaires

restaurants

id	nom	prenom	age	poids	id	nom	adresse	note	proprietaire_id
1		Michael			1	Raman			1
1		Michael			7	Affine			1
1		Michael			8	Colin			1
4		Sophie			3	Le ciel			4
4		Sophie			4	Raimond			4
4		Sophie			5	Bistro Toc			4

Nous allons de FAIRE LA JOINTURE (LES COLLER)

proprietaires

restaurants

p.id	p.nom	p.prenom	p.age	p.poids	r.id	r.nom	r.adresse	r.note	r.proprietaire_id
1		Michael			1	Raman			1
1		Michael			7	Affine			1
1		Michael			8	Colin			1
4		Sophie			3	Le ciel			4
4		Sophie			4	Raimond			4
4		Sophie			5	Bistro Toc			4

TABLE

JOINTURE



TABLE

TABLE

**LA JOINTURE** permet de coller toutes les informations de plusieurs tables ayant une relation entre-elles.

Elle colle les lignes en utilisant la FK.

