

# Introduction aux Bases De Données

# Sommaire

Une Base De Donnée, qu'est-ce que c'est ?

Un peu d'histoire

Différents modèles de SGBD

Un peu d'algèbre relationnel

Représentation des relations par des tableaux

Premiers pas avec SQL

Travaux Pratiques

Quelques liens...

# Une Base De Donnée, qu'est-ce que c'est ?

## ***Wikipédia :***

Une base de données est un conteneur informatique permettant de stocker [...] l'intégralité des informations en rapport avec une activité.

## ***Larousse :***

Ensemble structuré de fichiers regroupant des informations ayant certains caractères en commun ; logiciel permettant de constituer et de gérer ces fichiers.

## ***Universalis :***

Une base de données rassemble sur un support informatique une grande quantité d'information sur un sujet donné. Cette information est représentée par une ou plusieurs collections dont les éléments doivent respecter une structure précise.

# Une Base De Donnée, qu'est-ce que c'est ?

## *Pour résumer :*

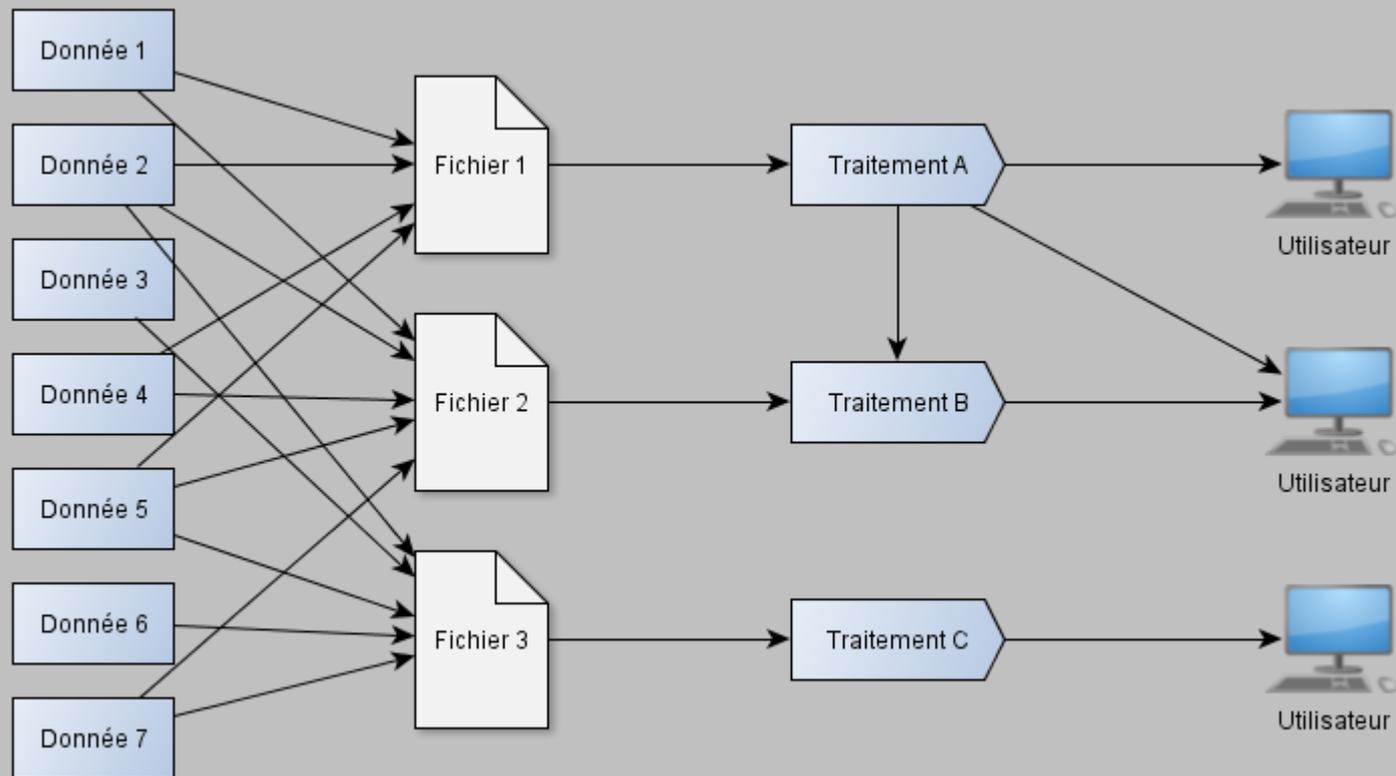
une base de données est la mémoire d'un système d'information. Elle est constituée d'un ensemble cohérent d'informations complexes.

Les informations élémentaires (données) sont regroupées en fonction du sens qu'on leur donne.

# Un peu d'histoire

*Jusqu'aux années 60 :*

les données étaient stockées de façon « classique » dans des fichiers.



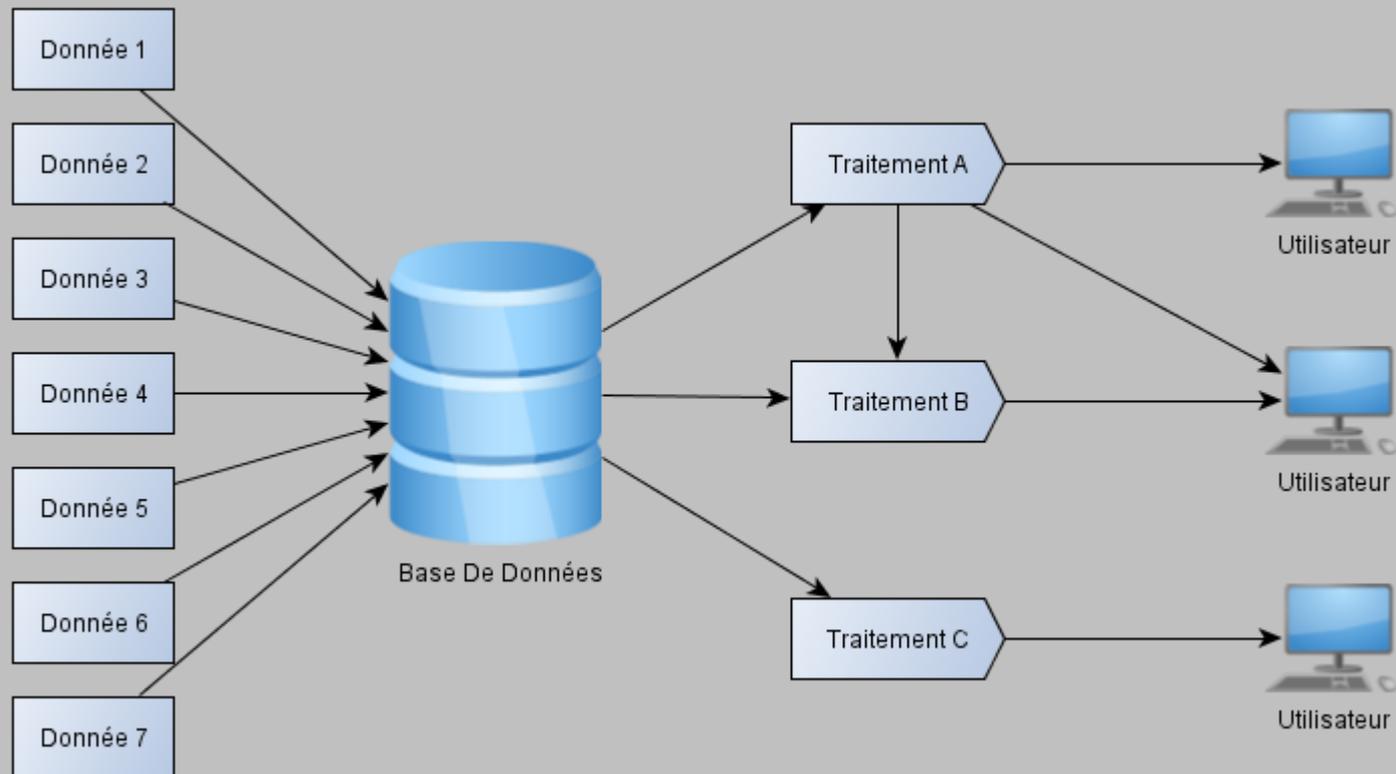
# Un peu d'histoire

## *Principaux inconvénients de cette structure :*

- ✓ le format de fichier dépend en général du traitement ;
- ✓ De fait, la saisie est à refaire pour chaque fichier ;
- ✓ même chose pour les mises à jour...
- ✓ contrôle des erreurs au moment du traitement : plus de risques d'erreurs et délais plus longs pour répercuter les corrections ;
- ✓ grande redondance des données : l'intégrité n'est pas garantie et les ressources nécessaire pour le stockage sont importantes.

# Un peu d'histoire

*A partir de la fin des années 60 :*  
apparition des premiers SGBD  
(Systèmes de Gestion de Bases de données)



# Un peu d'histoire

## *Principaux avantages de cette structure :*

✓une seule saisie ;

✓les mises à jours sont immédiatement répercutées ;

✓la validité des données est assurée par le SGBD : moins de risque d'erre

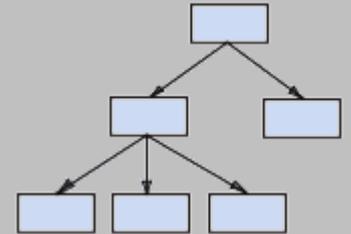
✓unicité de l'information ;

✓rationalisation des ressources pour le stockage.

# Différents modèles de Systèmes de Gestion de Bases de Données

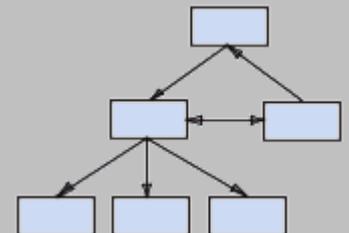
## ***le modèle hiérarchique :***

les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD. L'accès à une donnée se fait par des chemins pré-établis.



## ***le modèle réseau :***

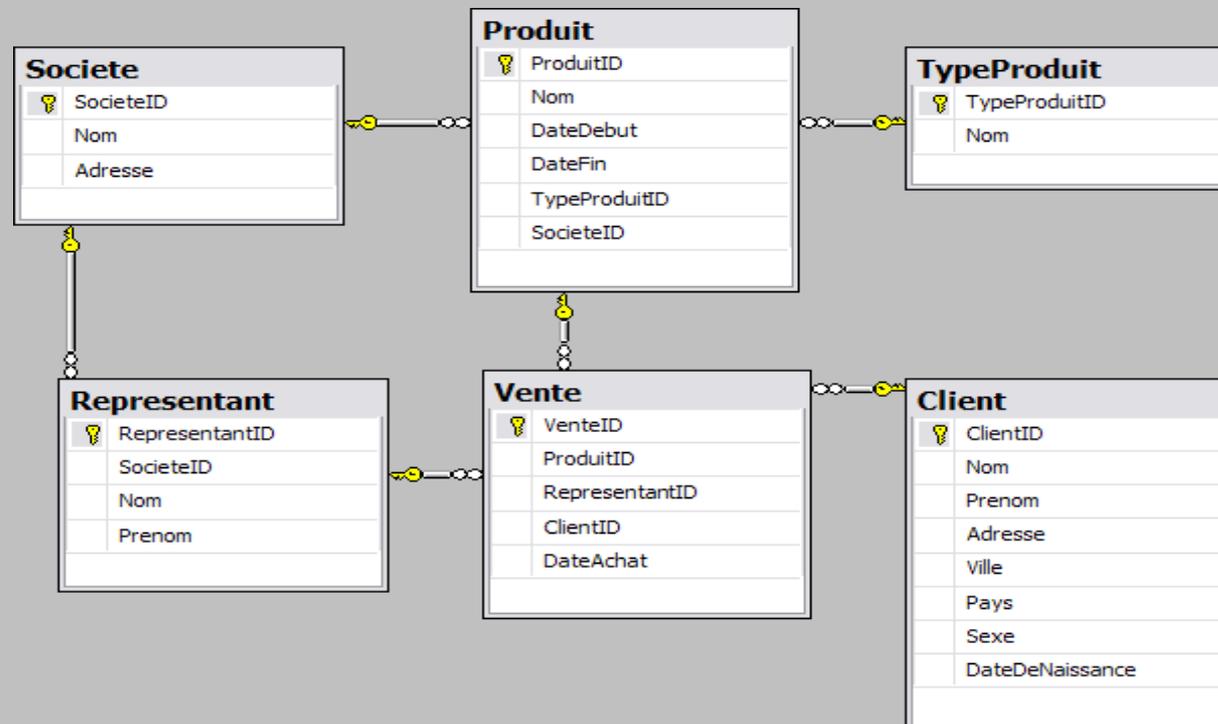
comme le modèle hiérarchique, ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant. Là aussi, l'accès à une donnée se fait par des chemins pré-établis.



# Différents modèles de Systèmes de Gestion de Bases de Données

## *le modèle relationnel :*

les données sont **décomposées et organisées** dans des matrices appelées relations ou tables. Le contenu de la base de données peut ainsi être synthétisé par des opérations d'algèbre relationnelle telles que l'intersection, la jointure et le produit cartésien.



# Différents modèles de Systèmes de Gestion de Bases de Données

## *le modèle relationnel (suite).*

Le modèle relationnel est l'un des plus répandus.

Les bases de données relationnelles utilisent en général un langage d'interrogation et de mise à jour appelé SQL :

Structured Query Language (langage de requête structuré)

C'est un langage déclaratif : on décrit ce qu'on veut obtenir, la machine fait le nécessaire.

# Un peu d'algèbre relationnel

L'algèbre relationnelle est une théorie mathématique proche de la théorie des ensembles qui définit des opérations qui peuvent être effectuées sur des relations (des matrices contenant un ensemble de nuplets).

Une définition qui nous suffit :

**Un ensemble est une collection d'éléments**

Un ensemble peut être :

.vide  $\emptyset$

.Infini Exemple :

.fini Exemple :  $A = \{1, 3, 8, 12, 20\}$

**Tous nos ensembles seront finis !**

**Cardinal d'un ensemble fini : nombre de ses éléments**

# Un peu d'algèbre relationnel

## ***Opérations de base sur les ensembles***

Union de deux ensembles :

Intersection de deux ensembles :

Produit cartésien de deux ensembles :

Les éléments de  $A.B$  sont des couples.

Les éléments d'un produit cartésien de  $n$  ensembles sont de  $n$ -uplets.

# Un peu d'algèbre relationnel

## **Relation :**

sous-ensemble (fini) du produit cartésien de plusieurs domaines.

## **Par extension :**

un domaine est une relation ;

Le produit cartésien de plusieurs relations est une relation.

# Un peu d'algèbre relationnel

## **Exemple :**

$A = \{\text{vert, jaune, rouge}\}$

$B = \{\text{allumé, éteint}\}$

$A.B = \{(\text{vert, allumé}), (\text{vert, éteint}), (\text{jaune, allumé}), (\text{jaune, éteint}),$   
 $(\text{rouge, allumé}), (\text{rouge, éteint})\}$

$$|A.B| = |A| \times |B|$$

## **Domaine :**

Un domaine est un ensemble dont les éléments sont des « valeurs ».

A et B sont des domaines

# Représentation des relations par des tableaux

On peut représenter les relations sous forme de tableaux :

Table A

A
Vert
Jaune
Rouge

Table B

B
Allumé
Éteint

Table C

A	B
Vert	Allumé
Vert	Éteint
Rouge	Allumé

Les valeurs d'une « colonne » appartiennent à un même domaine, mais on parlera plutôt de de « champ » à la place de colonne.

En entête de colonne, on trouve le nom du champ.

Chaque ligne correspond à un n-uplet, mais on parlera plutôt « d'enregistrements »

# Représentation des relations par des tableaux

**Noms des champs d'un produit de deux relations :**

R1			R2		R1.R		
Ville	Dept	Dept	Région	Ville	R1.Dept	R2.Dept	Région
Orléans	45	28	Centre	Orléans	45	28	Centre
Chartres	28	45	Centre	Orléans	45	45	Centre
				Chartres	28	28	Centre
				Chartres	28	45	Centre

Dans R1 et R2, deux champs portent le même nom...

Pour les distinguer dans R1.R2, on utilise des « noms qualifiés ».

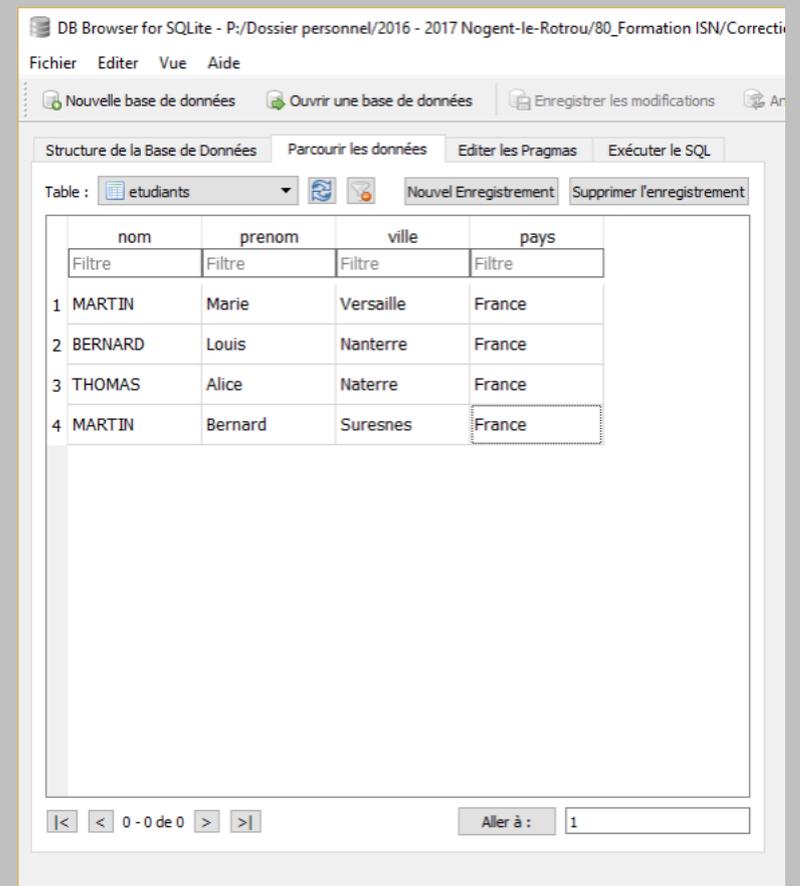
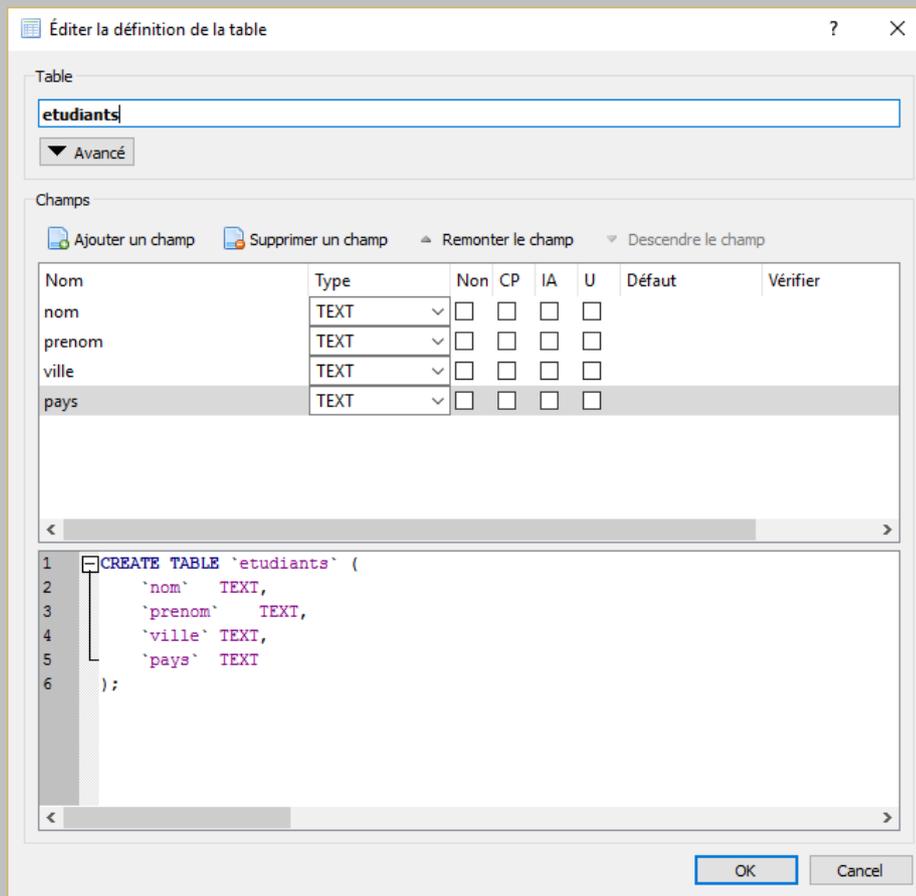
# Premiers pas avec SQL

## Création d'une base de données avec DB Browser for SQLite

Nom de la base de données : universite\_1

Une seule table : etudiants

4 enregistrements



# Premiers pas avec SQL

## **Projection :**

C'est une opération sur les relations.

Elle permet à partir d'une relation existante d'en obtenir une nouvelle en ne conservant qu'une partie de ses **champs**.

nom	prenom	ville	pays
MARTIN	Marie	Versailles	France
BERNARD	Louis	Nanterre	France
THOMAS	Alice	Nanterre	France
MARTIN	Bernard	Suresnes	France

nom	prénom
MARTIN	Marie
BERNARD	Louis
THOMAS	Alice
MARTIN	Bernard

En SQL, on parle d'une sélection :

**SELECT nom,prenom FROM etudiants ;**

# Premiers pas avec SQL

## **Restriction:**

C'est une opération sur les relations.

Elle permet à partir d'une relation existante d'en obtenir une nouvelle en ne conservant qu'une partie de ses **enregistrements**

etudiants

Projection aux Nanterrois			pays
nom	prenom	ville	
MARTIN	Marie	Versailles	France
BERNARD	Louis	Nanterre	France
THOMAS	Alice	Nanterre	France
MARTIN	Bernard	Suresnes	France

nom	prenom	ville	pays
BERNARD	Louis	Nanterre	France
THOMAS	Alice	Nanterre	France



```
SELECT * FROM etudiants WHERE ville="Nanterre";
```

# Premiers pas avec SQL

***On peut combiner les deux :***

Étudiants

nom	prenom	ville	pays
MARTIN	Marie	Versailles	France
BERNARD	Louis	Nanterre	France
THOMAS	Alice	Nanterre	France
MARTIN	Bernard	Suresnes	France

nom	prenom
BERNARD	Louis
THOMAS	Alice

**SELECT nom,prenom FROM etudiants WHERE ville="Nanterre" ;**



# Premiers pas avec SQL

## Création d'une autre base de données

Nom de la base de données : *universite\_2*

Deux tables : *etudiants*

Éditer la définition de la table

Table

Avancé

Champs

Capture Forme libre

Ajouter un champ Supprimer un champ Remonter le champ Descendre le champ

Nom	Type	No	CP	IA	U	Défaut	Vérifier
nom	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
prenom	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ville	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```
1 CREATE TABLE `etudiants` (  
2   `nom` TEXT,  
3   `prenom` TEXT,  
4   `ville` INTEGER  
5 );
```

OK Cancel

Éditer la définition de la table

Table

Avancé

Champs

Ajouter un champ Supprimer un champ Remonter le champ Descendre le champ

Nom	Type	No	CP	IA	U	Défaut	Vérifier
ref	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
ville	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
pays	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```
1 CREATE TABLE `villes` (  
2   `ref` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,  
3   `ville` TEXT,  
4   `pays` TEXT  
5 );
```

OK Cancel

# Premiers pas avec SQL

## *Projection, produit cartésien et restriction :*

etudiants

nom	prenom	ville
MARTIN	Marie	3
BERNARD	Louis	2
THOMAS	Alice	2
MARTIN	Bernard	4

villes

ref	ville	pays
1	Paris	France
2	Nanterre	France
3	Versailles	France
4	Suresnes	France

```
SELECT etudiants.nom,etudiants.prenom  
FROM villes,etudiants WHERE (etudiants.ville=villes.ref) AND (villes.ville="Nanterre");
```

⌘

nom	prenom
BERNARD	Louis
THOMAS	Alice

# Premiers pas avec SQL

## *Liens logiques :*

Le champ Ville de la relation Étudiants relie chaque enregistrement à un enregistrement de la relation Villes. C'est une **clé externe**.

Le champ Ref de la relation Ville identifie chaque enregistrement de la relation Villes. Les valeurs de ce champ doivent être uniques. C'est une **clé primaire**.

etudiants

villes

nom	prénom	Ville
MARTIN	Marie	3
BERNARD	Louis	2
THOMAS	Alice	2
MARTIN	Bernard	4

ref	ville	pays
1	Paris	France
2	Nanterre	France
3	Versailles	France
4	Suresnes	France