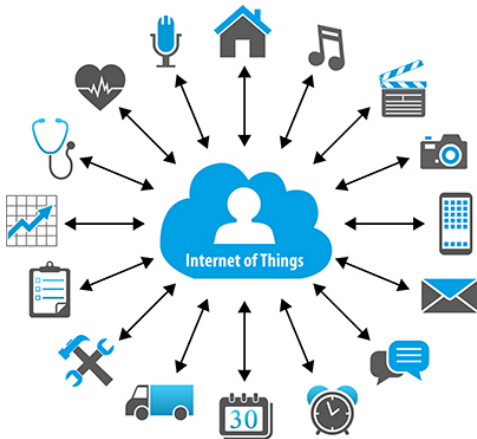
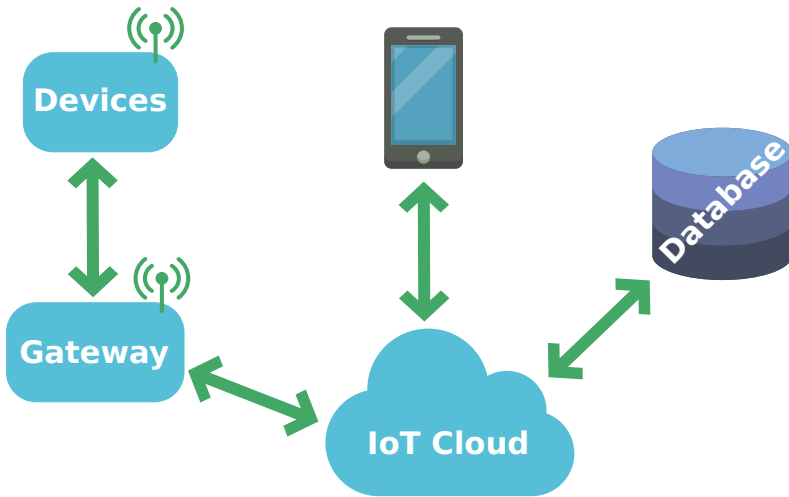


Internet of Things

Roselyne CHOTIN
(largely inspired from Cécile Braunstein)





Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

1 Bases de données

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

1 Bases de données

- Approche
- Modèle relationnel
- Normalisation
- Jointures
- Langage SQL
- Modèle NoSQL

Internet of
Things

Bases de
données

Approche

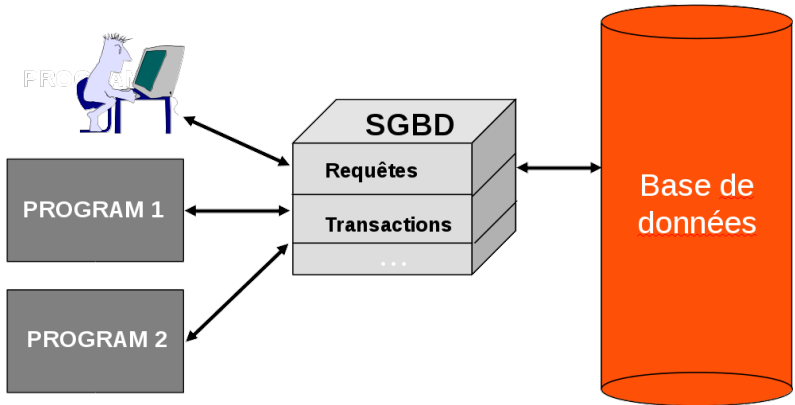
Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL



- Stockage unique des données (pas de redondance) sur une seule machine
 - les infos sur les étudiants de Polytech Sorbonne
- Interrogation à distance
 - sqlplus, phpMyAdmin
- Un intergiciel entre les accès et les données
 - Oracle, MySQL
- Langage déclaratifs, les accès aux données sont intentionnels (pas de structure de contrôle) ; Standard : langage SQL.
 - Qui sont les étudiants en EISE5 ?

Une base de données (BD)

Une grande collection de données structurées sur des entités qui représentent des parties du monde que l'on cherche à stocker dans la base.

- la bases de données des médicaments disponibles dans une pharmacie, la base de données des étudiants inscrits à Polytech Sorbonne, ..

Un système de gestion de base de données (SGBD)

Un gros logiciel qui facilite la gestion et l'accès à une BD par de nombreux utilisateurs

- Oracle, MySQL, Postgresql, DB2, ...

Accès contrôlé aux données

- Contrôle de la cohérence et intégrité entre les données
- Contrôle de la concurrence entre plusieurs utilisateurs
- Contrôle des performances d'accès des utilisateurs qui interrogent la BD
- Contrôle de la sécurité : toutes les données ne sont pas visibles par tout le monde

Indépendance gestion des données / applications

- Indépendance logique : vues externes cachent les détails de l'organisation logique des données
- Indépendance physique : l'organisation physique des données (fichiers, disques etc.) ne remet pas en cause le schéma logique

■ Utilisateur final (end user)

- Accède la BD par des formes d'écran, des interfaces applicatives ou, pour les plus experts, des requêtes SQL

■ Programmeur d'application

- construit (avec l'utilisateur) le schéma conceptuel
- définit et gère le schéma logique et les vues
- conçoit et implémente des applications qui accèdent la BD

■ Administrateur de BD (DBA)

- gère le schéma physique et règle les performances charge et organise la BD
- gère la sécurité et la fiabilité

- **Langages de conception : E/A, UML**
 - Utilisation : conception haut-niveau d'applications (données et traitements)
- **Langage base de données : SQL**
 - langage déclaratif : l'utilisateur spécifie quoi (et non comment)
 - puissance d'expression limitée (par rapport à un langage de programmation comme C ou Java)
 - utilisation : définition schémas, interrogation et mises-à-jour, administration
- **Langages de programmation : PL/SQL, Java, PHP,...**
 - langages avec une bibliothèque SQL
 - langage complet (au sens d'Alan Turing)
 - utilisation : programmation d'applications complète

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

1 Bases de données

- Approche
- **Modèle relationnel**
- Normalisation
- Jointures
- Langage SQL
- Modèle NoSQL

- Un modèle de données **logique** permet de décrire
 - la structure des données : **schéma**
 - les données : **instances**
 - les **opérations** sur le schéma et les données
- Exemple : Modèle relationnel
 - **schéma** = ensemble de noms de tables ou relations avec des attributs
 - **instance** = ensemble de n-uplets (tuples, lignes) stockés dans les tables
 - **opération** = expression SQL

Avantages

- "Relation" est un concept simple avec des fondements mathématiques solides (théorie des ensembles, logique du premier ordre)
- Langages de requêtes simples, puissants et efficaces (techniques d'optimisation performantes)
- Théorie pour la conception (dépendances fonctionnelles)
- Standard : SQL

Chaque bibliothèque possède des documents qui peuvent être empruntés par les étudiants. Les bibliothèques ont un nom et une localisation. Les étudiants ont un nom, un prénom et une adresse. Les documents ont un titre et sont rédigés par un auteur qui a lui-même un nom et un prénom.

Chaque bibliothèque possède des documents qui peuvent être empruntés par les étudiants. Les bibliothèques ont un nom et une localisation. Les étudiants ont un nom, un prénom et une adresse. Les documents ont un titre et sont rédigés par un auteur qui a lui-même un nom et un prénom.

Bibliothèque	Etudiant	Document
--------------	----------	----------

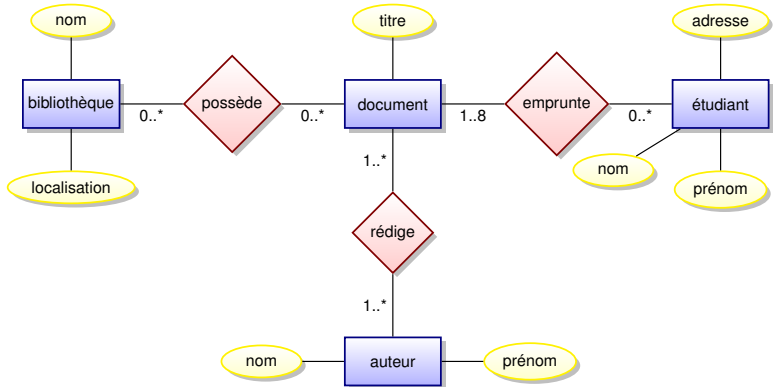
Chaque bibliothèque possède des documents qui peuvent être empruntés par les étudiants. Les bibliothèques ont un nom et une localisation. Les étudiants ont un nom, un prénom et une adresse. Les documents ont un titre et sont rédigés par un auteur qui a lui-même un nom et un prénom.

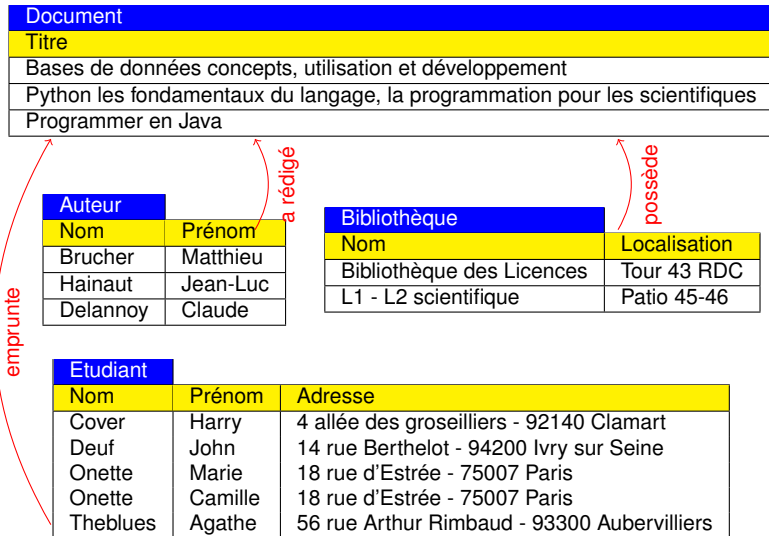
Bibliothèque	Etudiant	Document
--------------	----------	----------

Bibliothèque des Licences, Tour 43 RDC
Cover, Harry, 4 allée des groseilliers - 92140 Clamart
Programmer en Java, Delannoy, Claude

Chaque bibliothèque possède des documents qui peuvent être empruntés par les étudiants. Les bibliothèques ont un nom et une localisation. Les étudiants ont un nom, un prénom et une adresse. Les documents ont un titre et sont rédigés par un auteur qui a lui-même un nom et un prénom.

entité association attribut





Marie Onette emprunte le livre "Bases de données concepts, utilisation et développement" jusqu'au 14/10/2020

Document
Titre
Bases de données concepts, utilisation et développement
Python les fondamentaux du langage, la programmation pour les scientifiques
Programmer en Java

emprunte

Emprunte
Jusqu'au
14/10/2020

Etudiant		
Nom	Prénom	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

Décrit la connectivité entre les entités

- Un à un (1 :1) : *une personne ne peut emprunter qu'un livre*
- Un à plusieurs (1 :N) : *une personne peut emprunter plusieurs livres*
- Plusieurs à un (N :1) : *plusieurs personnes peuvent emprunter le même livre*
- Plusieurs à plusieurs (N :N) : *des personnes peuvent emprunter plusieurs livres et un livre peut être emprunté par plusieurs personnes*

- Sous-ensemble d'attributs
- Identifier de manière unique une instance
- Irréductible

Nom	Prénom	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

- Clé primaire : une parmi les clés candidates
- Identifiant unique : clé artificielle simple

- Sous-ensemble d'attributs
- Identifier de manière unique une instance
- Irréductible

Id	Nom	Prénom	Adresse
1	Cover	Harry	4 allée des groseilliers - 92140 Clamart
2	Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
3	Onette	Marie	18 rue d'Estrée - 75007 Paris
4	Onette	Camille	18 rue d'Estrée - 75007 Paris
5	Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

- Clé primaire : une parmi les clés candidates
- Identifiant unique : clé artificielle simple

Marie Onette emprunte le livre "Bases de données concepts, utilisation et développement" jusqu'au 14/10/2020

Id	Nom	Prénom	Adresse
1	Cover	Harry	4 allée des groseilliers - 92140 Clamart
2	Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
3	Onette	Marie	18 rue d'Estrée - 75007 Paris
4	Onette	Camille	18 rue d'Estrée - 75007 Paris
5	Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

Id	Titre
1	Bases de données concepts, utilisation et développement
2	Python les fondamentaux du langage, la programmation pour les scientifiques
3	Programmer en Java

Id_doc	Id_etu	Jusqu'au
1	3	14/10/2020

Id	Adresse	Code postal	Ville
1	4 allée des groseilliers	92140	Clamart
2	14 rue Berthelot	94200	Ivry sur Seine
3	18 rue d'Estrée	75007	Paris
4	56 rue Arthur Rimbaud	93300	Aubervilliers

Id	Nom	Prénom	Id_ad
1	Cover	Harry	1
2	Deuf	John	2
3	Onette	Marie	3
4	Onette	Camille	3
5	Theblues	Agathe	4

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

1 Bases de données

- Approche
- Modèle relationnel
- **Normalisation**
- Jointures
- Langage SQL
- Modèle NoSQL

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

- Redondance d'informations
- Anomalie d'insertion
- Anomalie de mise à jour
- Anomalie de suppression

- Redondance d'informations
- Anomalie d'insertion
- Anomalie de mise à jour
- Anomalie de suppression

<u>Nom</u>	<u>Prénom</u>	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

- 1^{re} forme normale (1FN) : atomicité des attributs
- 2^e forme normale (2FN) : un attribut non clé ne peut pas dépendre que d'une partie de la clé
- 3^e forme normale (3FN) : un attribut non clé ne peut pas dépendre d'un autre attribut non clé
- Forme normale de Boyce-Codd (FNBC) : aucun attribut clé ne peut dépendre d'un attribut non clé
- Dépendances multi-valuées : 4^e à 6^e forme normale et forme normale domaine clé (FNDC)

<u>Nom</u>	<u>Prénom</u>	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

Exemple

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

<u>Nom</u>	<u>Prénom</u>	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

<u>Nom</u>	<u>Prénom</u>	n°	Rue	Code	Ville
Cover	Harry	4	allée des groseilliers	92140	Clamart
Deuf	John	14	rue Berthelot	94200	Ivry sur Seine
Onette	Marie	18	rue d'Estrée	75007	Paris
Onette	Camille	18	rue d'Estrée	75007	Paris
Theblues	Agathe	56	rue Arthur Rimbaud	93300	Aubervilliers

- 1^{re} forme normale (1FN) : atomicité des attributs
- 2^e forme normale (2FN) : un attribut non clé ne peut pas dépendre que d'une partie de la clé
- 3^e forme normale (3FN) : un attribut non clé ne peut pas dépendre d'un autre attribut non clé
- Forme normale de Boyce-Codd (FNBC) : aucun attribut clé ne peut dépendre d'un attribut non clé
- Dépendances multi-valuées : 4^e à 6^e forme normale et forme normale domaine clé (FNDC)

Exemple

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

<u>Nom</u>	<u>Prénom</u>	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

<u>Nom</u>	<u>Prénom</u>	n°	Rue	Code	Ville
Cover	Harry	4	allée des groseilliers	92140	Clamart
Deuf	John	14	rue Berthelot	94200	Ivry sur Seine
Onette	Marie	18	rue d'Estrée	75007	Paris
Onette	Camille	18	rue d'Estrée	75007	Paris
Theblues	Agathe	56	rue Arthur Rimbaud	93300	Aubervilliers

<u>Nom</u>	<u>Prénom</u>
Cover	Harry
Deuf	John
Onette	Marie
Onette	Camille
Theblues	Agathe

<u>Nom</u>	n°	Rue	Code	Ville
Cover	4	allée des groseilliers	92140	Clamart
Deuf	14	rue Berthelot	94200	Ivry sur Seine
Onette	18	rue d'Estrée	75007	Paris
Theblues	56	rue Arthur Rimbaud	93300	Aubervilliers

- 1^{re} forme normale (1FN) : atomicité des attributs
- 2^e forme normale (2FN) : un attribut non clé ne peut pas dépendre que d'une partie de la clé
- 3^e forme normale (3FN) : un attribut non clé ne peut pas dépendre d'un autre attribut non clé
- Forme normale de Boyce-Codd (FNBC) : aucun attribut clé ne peut dépendre d'un attribut non clé
- Dépendances multi-valuées : 4^e à 6^e forme normale et forme normale domaine clé (FNDC)

<u>Nom</u>	<u>Prénom</u>
Cover	Harry
Deuf	John
Onette	Marie
Onette	Camille
Theblues	Agathe

<u>Nom</u>	n°	Rue	Code	Ville
Cover	4	allée des groseilliers	92140	Clamart
Deuf	14	rue Berthelot	94200	Ivry sur Seine
Onette	18	rue d'Estrée	75007	Paris
Theblues	56	rue Arthur Rimbaud	93300	Aubervilliers

<u>Nom</u>	n°	Rue	Code
Cover	4	allée des groseilliers	92140
Deuf	14	rue Berthelot	94200
Onette	18	rue d'Estrée	75007
Theblues	56	rue Arthur Rimbaud	93300

<u>Code</u>	Ville
92140	Clamart
94200	Ivry sur Seine
75007	Paris
93300	Aubervilliers

- 1^{re} forme normale (1FN) : atomicité des attributs
- 2^e forme normale (2FN) : un attribut non clé ne peut pas dépendre que d'une partie de la clé
- 3^e forme normale (3FN) : un attribut non clé ne peut pas dépendre d'un autre attribut non clé
- Forme normale de Boyce-Codd (FNBC) : aucun attribut clé ne peut dépendre d'un attribut non clé
- Dépendances multi-valuées : 4^e à 6^e forme normale et forme normale domaine clé (FNDC)

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

<u>Nom</u>	Prénom	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers

- 1^{re} forme normale (1FN) : atomicité des attributs
- 2^e forme normale (2FN) : un attribut non clé ne peut pas dépendre que d'une partie de la clé
- 3^e forme normale (3FN) : un attribut non clé ne peut pas dépendre d'un autre attribut non clé
- Forme normale de Boyce-Codd (FNBC) : aucun attribut clé ne peut dépendre d'un attribut non clé
- Dépendances multi-valuées : 4^e à 6^e forme normale et forme normale domaine clé (FNDC)

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

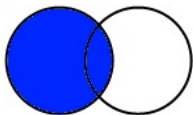
Modèle NoSQL

1 Bases de données

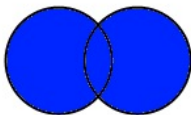
- Approche
- Modèle relationnel
- Normalisation
- **Jointures**
- Langage SQL
- Modèle NoSQL

- Permet d'accéder aux données liées à une clé étrangère
- Différents types de jointures

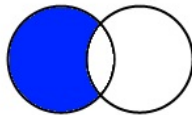
LEFT JOIN



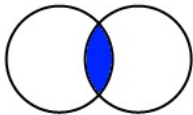
FULL OUTER JOIN



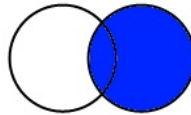
**LEFT JOIN
(if NULL)**



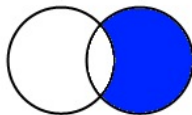
INNER JOIN



RIGHT JOIN



**RIGHT JOIN
(if NULL)**



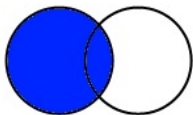
<u>Nom</u>	<u>Prénom</u>	Document
Cover	Harry	1
Deuf	John	
Onette	Marie	
Onette	Camille	
Theblues	Agathe	2

Id	Titre
1	Bases de données concepts, utilisation et développement
2	Python les fondamentaux du langage, la programmation pour les scientifiques
3	Programmer en Java

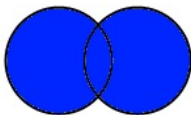
Left joint : tous les étudiants

- Permet d'accéder aux données liées à une clé étrangère
- Différents types de jointures

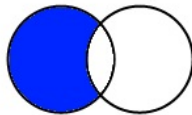
LEFT JOIN



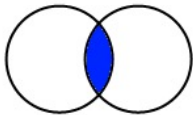
FULL OUTER JOIN



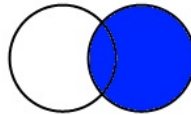
**LEFT JOIN
(if NULL)**



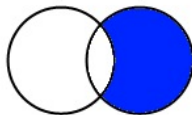
INNER JOIN



RIGHT JOIN



**RIGHT JOIN
(if NULL)**



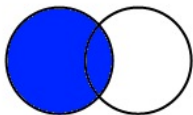
Nom	Prénom	Document
Cover	Harry	1
Deuf	John	
Onette	Marie	
Onette	Camille	
Theblues	Agathe	2

Id	Titre
1	Bases de données concepts, utilisation et développement
2	Python les fondamentaux du langage, la programmation pour les scientifiques
3	Programmer en Java

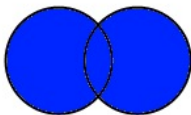
Full outer : tous les étudiants et tous les documents

- Permet d'accéder aux données liées à une clé étrangère
- Différents types de jointures

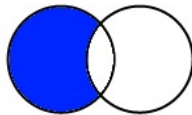
LEFT JOIN



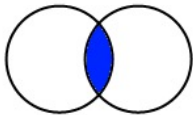
FULL OUTER JOIN



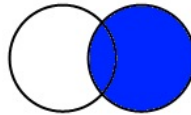
**LEFT JOIN
(if NULL)**



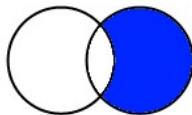
INNER JOIN



RIGHT JOIN



**RIGHT JOIN
(if NULL)**



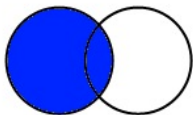
<u>Nom</u>	<u>Prénom</u>	Document
Cover	Harry	1
Deuf	John	
Onette	Marie	
Onette	Camille	
Theblues	Agathe	2

Id	Titre
1	Bases de données concepts, utilisation et développement
2	Python les fondamentaux du langage, la programmation pour les scientifiques
3	Programmer en Java

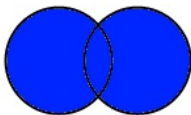
Left joint (if null) : tous les étudiants qui n'ont pas emprunté de document

- Permet d'accéder aux données liées à une clé étrangère
- Différents types de jointures

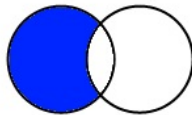
LEFT JOIN



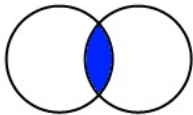
FULL OUTER JOIN



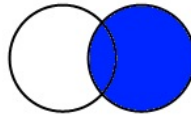
**LEFT JOIN
(if NULL)**



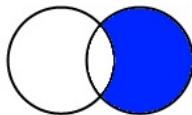
INNER JOIN



RIGHT JOIN



**RIGHT JOIN
(if NULL)**



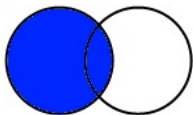
<u>Nom</u>	<u>Prénom</u>	<u>Document</u>
Cover	Harry	1
Deuf	John	2
Onette	Marie	
Onette	Camille	
Theblues	Agathe	

<u>Id</u>	<u>Titre</u>
1	Bases de données concepts, utilisation et développement
2	Python les fondamentaux du langage, la programmation pour les scientifiques
3	Programmer en Java

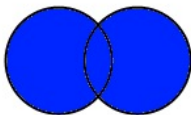
Inner joint : tous les étudiants qui ont emprunté un document
ou tous les documents qui ont été empruntés

- Permet d'accéder aux données liées à une clé étrangère
- Différents types de jointures

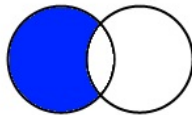
LEFT JOIN



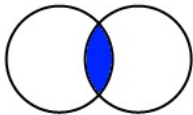
FULL OUTER JOIN



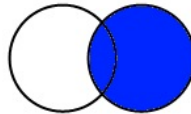
**LEFT JOIN
(if NULL)**



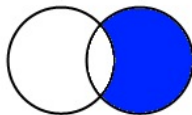
INNER JOIN



RIGHT JOIN



**RIGHT JOIN
(if NULL)**



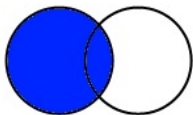
<u>Nom</u>	<u>Prénom</u>	Document
Cover	Harry	1
Deuf	John	
Onette	Marie	
Onette	Camille	
Theblues	Agathe	2

Id	Titre
1	Bases de données concepts, utilisation et développement
2	Python les fondamentaux du langage, la programmation pour les scientifiques
3	Programmer en Java

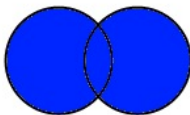
Right joint : tous les documents

- Permet d'accéder aux données liées à une clé étrangère
- Différents types de jointures

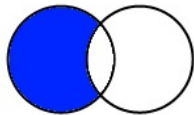
LEFT JOIN



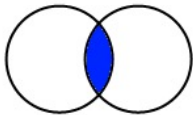
FULL OUTER JOIN



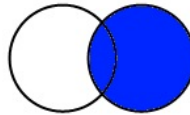
**LEFT JOIN
(if NULL)**



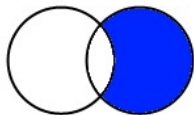
INNER JOIN



RIGHT JOIN



**RIGHT JOIN
(if NULL)**



<u>Nom</u>	<u>Prénom</u>	Document
Cover	Harry	1
Deuf	John	
Onette	Marie	
Onette	Camille	
Theblues	Agathe	2

Id	Titre
1	Bases de données concepts, utilisation et développement
2	Python les fondamentaux du langage, la programmation pour les scientifiques
3	Programmer en Java

Right joint (if null) : tous les documents non empruntés

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

1 Bases de données

- Approche
- Modèle relationnel
- Normalisation
- Jointures
- **Langage SQL**
- Modèle NoSQL

- **CREATE** : pour créer une table
- **SELECT** : pour extraire des données d'une table (conditions, tri)
- **DELETE** / **INSERT** : pour supprimer / insérer des données dans une table
- **DROP TABLE** : pour supprimer une table
- etc.

Exemple

```
CREATE TABLE etudiants (id INTEGER PRIMARY  
KEY AUTOINCREMENT, ...);  
CREATE TABLE etudiants (... , idAd INTEGER,  
FOREIGN KEY (idAd) REFERENCES Adresse(id));
```

- **CREATE** : pour créer une table
- **SELECT** : pour extraire des données d'une table (conditions, tri)
- **DELETE / INSERT** : pour supprimer / insérer des données dans une table
- **DROP TABLE** : pour supprimer une table
- etc.

Exemple

```
SELECT nom FROM etudiants;
```

```
SELECT * FROM etudiants WHERE nom="Onette";
```


- **CREATE** : pour créer une table
- **SELECT** : pour extraire des données d'une table (conditions, tri)
- **DELETE / INSERT** : pour supprimer / insérer des données dans une table
- **DROP TABLE** : pour supprimer une table
- etc.

Exemple

```
INSERT INTO etudiants VALUES ("The Blues",  
"Agathe");  
DELETE FROM etudiants WHERE nom="The Blues";
```

- **CREATE** : pour créer une table
- **SELECT** : pour extraire des données d'une table (conditions, tri)
- **DELETE / INSERT** : pour supprimer / insérer des données dans une table
- **DROP TABLE** : pour supprimer une table
- etc.


Exemple

```
DROP TABLE etudiants;
```

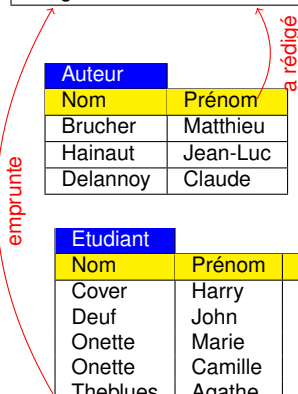
- CREATE : pour créer une table
- SELECT : pour extraire des données d'une table (conditions, tri)
- DELETE / INSERT : pour supprimer / insérer des données dans une table
- DROP TABLE : pour supprimer une table
- etc.

Document
Titre
Bases de données concepts, utilisation et développement
Python les fondamentaux du langage, la programmation pour les scientifiques
Programmer en Java

Auteur	
Nom	Prénom
Brucher	Matthieu
Hainaut	Jean-Luc
Delannoy	Claude

Bibliothèque	/ 
Nom	Localisation
Bibliothèque des Licences	Tour 43 RDC
L1 - L2 scientifique	Patio 45-46

Etudiant		
Nom	Prénom	Adresse
Cover	Harry	4 allée des groseilliers - 92140 Clamart
Deuf	John	14 rue Berthelot - 94200 Ivry sur Seine
Onette	Marie	18 rue d'Estrée - 75007 Paris
Onette	Camille	18 rue d'Estrée - 75007 Paris
Theblues	Agathe	56 rue Arthur Rimbaud - 93300 Aubervilliers



Exemple : CREATE

```
CREATE TABLE Document (id INTEGER PRIMARY KEY AUTOINCREMENT,  
Titre TEXT NOT NULL);  
CREATE TABLE Auteur (id INTEGER PRIMARY KEY AUTOINCREMENT,  
Nom TEXT NOT NULL, Prenom TEXT NOT NULL);  
CREATE TABLE Bibliotheque (id INTEGER PRIMARY KEY  
AUTOINCREMENT, Nom TEXT NOT NULL, Localisation TEXT NOT  
NULL);  
CREATE TABLE Etudiant (id INTEGER PRIMARY KEY AUTOINCREMENT,  
Nom TEXT NOT NULL, Prenom TEXT NOT NULL, idAd INTEGER  
NOT NULL, FOREIGN KEY (idAd) REFERENCES Adresse(id));  
CREATE TABLE Adresse (id INTEGER PRIMARY KEY AUTOINCREMENT,  
Numero INTEGER NOT NULL, Rue TEXT NOT NULL, code INTEGER  
NOT NULL, FOREIGN KEY (code) REFERENCES Ville(code));  
CREATE TABLE Ville (code INTEGER PRIMARY KEY, Nom TEXT NOT  
NULL);
```

Exemple : CREATE

```
CREATE TABLE Emprunte (idEtu INTEGER NOT NULL, idDoc INTEGER NOT NULL, Date TEXT NOT NULL, FOREIGN KEY (idEtu) REFERENCES Etudiant(id), FOREIGN KEY (idDoc) REFERENCES Document(id), PRIMARY KEY (idEtu,idDoc));  
CREATE TABLE Possede (idBibli INTEGER NOT NULL, idDoc INTEGER NOT NULL, Ref TEXT NOT NULL, FOREIGN KEY (idBibli) REFERENCES Bibliotheque(id), FOREIGN KEY (idDoc) REFERENCES Document(id), PRIMARY KEY (idBibli ,idDoc));  
CREATE TABLE Redige (idAut INTEGER NOT NULL, idDoc INTEGER NOT NULL, Date TEXT NOT NULL, FOREIGN KEY (idAut) REFERENCES Auteur(id), FOREIGN KEY (idDoc) REFERENCES Document(id), PRIMARY KEY (idAut ,idDoc));
```

Exemple : SELECT

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

```
SELECT * FROM Etudiant;  
SELECT * FROM Etudiant ORDER BY Nom ASC;  
SELECT * FROM Etudiant WHERE Nom = "Onette";  
SELECT Nom FROM Etudiant INNER JOIN Document ON Document.id =  
    Etudiant.idDoc;
```

Exemple : INSERT

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

```
INSERT INTO Document (Titre) VALUES("Bases_de_donnees_
concepts,_utilisation_et_developpement");
INSERT INTO Emprunte VALUES (3, 1, "14/10/2020");
INSERT INTO Emprunte (idEtu, idDoc, Date)
SELECT
  (SELECT id FROM Etudiant WHERE Nom = "Onette" AND Prenom
   = "Marie") as idEtu,
  (SELECT id FROM Document WHERE Titre = "Bases_de_donnees_
concepts,_utilisation_et_developpement") as idDoc,
  "14/10/2020" as Date;
```


Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

1 Bases de données

- Approche
- Modèle relationnel
- Normalisation
- Jointures
- Langage SQL
- **Modèle NoSQL**

- Grandes entreprises de Web
 - Volumes de données importants
 - Architectures distribuées
- Tableau associatif unidimensionnel : clé,valeur
- Google (BigTable), Amazon (Dynamo), SourceForge (MongoDB)

- SGBD NoSQL orienté documents
- Collections de documents
- Données structurées au format JSON
- Champs libres, mais clé principale obligatoire
- Pas de schéma relationnel prédéfini

```
{
  "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris_Park_Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris_Park_Bake_Shop",
  "restaurant_id": "30075445"
}
```

Internet of
Things

Bases de
données

Approche

Modèle relationnel

Normalisation

Jointures

Langage SQL

Modèle NoSQL

- Par l'utilisation d'un langage de programmation
- Requêtes simples : insertion, recherche, mise à jour, suppression