

SQL déclaratif

du standard à la pratique

Langage d'interrogation des Bases de
Données

Plan du cours

Partie 1 : Introduction aux concepts

- Base de données et SGBD 10
- De l'analyse au relationnel 14
- Notions de tables 23
- Contraintes 25

Plan du cours

Partie 2 : DDL – Data Definition Language

(Langage de définition des données)

• CREATE TABLE	35
• IDENTITY et DEFAULT	43
• Contraintes	46
• ALTER TABLE	58
• TRUNCATE TABLE	61
• DROP TABLE	62

Plan du cours

Partie 3 : DRL – Data Retrieval Language

(Langage de d'extraction des données)

- La clause « SELECT » 67
- Limiter et ordonner 78
- Les fonctions 96
- GROUP BY 126
- Jointures 137
- Sous-requêtes 164

Plan du cours

Partie 4 : DML – Data Manipulation Language

(Langage de de manipulation des données)

- Insertion de données 187
- Mise à jour de données 193
- Suppression de données 196
- OUTPUT 197

Partie 5 : Notions avancées

- Gestion des transactions 199
- Fusion de données 202

Auto-Evaluation

Afin de vous assurer que vous êtes en phase avec la formation et que vous intégrez la matière correctement et à un bon rythme, nous vous proposerons, à la fin de chaque module du cours, un petit tableau d'évaluation. Ce tableau a pour but de rappeler les notions phares du module et nous vous invitons à le remplir pour vous-même afin de vous rendre compte des notions que vous devez peut-être revoir, de celles pour lesquelles vous devrez demander plus d'explications au formateur ou encore que vous avez tout compris !

Dans ces petites auto-évaluations, les lettres suivantes sont utilisées pour vous aider évaluer le niveau avec lequel vous sentez avoir compris la matière :

- **Parfait (P)** : vous avez parfaitement compris cette notion et vous vous sentez à votre aise
- **Satisfaisant (S)** : vous avez compris de quoi il s'agit mais la pratique vous manque
- **Vague (V)** : vous savez de quoi il s'agit, mais cela reste un peu vague dans votre esprit. Une explication supplémentaire du formateur ou une bonne révision de votre part s'impose
- **Insatisfaisant (I)** : Vous n'avez pas du tout compris la notion abordée, il faut tout faire pour y remédier !

Auto-Evaluation

Résumé de l'ensemble des notions

Notions	P	S	V	I
Base de données et SGBD				
Schéma relationnel (création, fonctionnement, lecture)				
Contrainte de « PRIMARY KEY »				
Contrainte de « FOREIGN KEY »				
Contraintes « UNIQUE », « CHECK », « NOT NULL »				
Création de table et contraintes (CREATE TABLE)				
Auto-incrémentation				
Modification de la structure d'une table existante (ALTER TABLE)				
Réinitialisation d'une table et suppression (TRUNCATE et DROP)				

Auto-Evaluation

Résumé de l'ensemble des notions

Notions	P	S	V	I
Ordre « SELECT ... FROM » simple				
Clauses WHERE, GROUP BY et HAVING, ORDER BY				
Fonctions simples et fonctions d'agrégation				
Jointures				
Requêtes imbriquées				
Ordres DML (INSERT, UPDATE, DELETE)				
Transaction (principes, loi ACID, ordres COMMIT et ROLLBACK)				

Partie 1

INTRODUCTION AUX CONCEPTS

Base de données et SGBD

De l'analyse au relationnel

Notions de tables

Contraintes

Base de données et SGBD

Une **Base de données** est une structure, le plus souvent informatique, permettant le stockage et l'exploitation de données

Pendant longtemps, nous nous sommes contentés de stocker nos données sur **des supports papiers**, dans des classeurs, dans des livres, dans des bibliothèques. Sans autre ressource technologique, cette technique rendait parfois le stockage, l'organisation et l'exploitation des données **un peu lourd**, sans compter l'**espace nécessaire** pour entreposer ces données

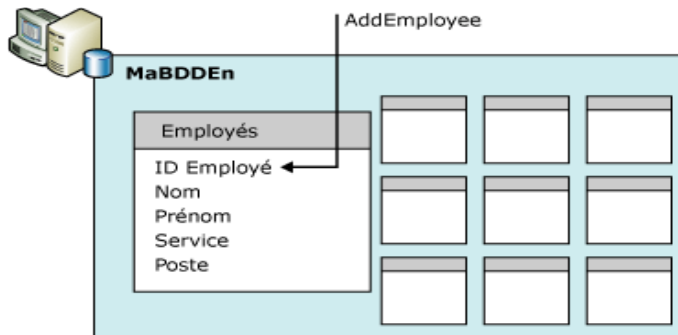
L'**arrivée de l'ordinateur** et des programmes informatiques a ensuite permit de **stocker les données dans des fichiers informatiques**, simplifiant et allégeant déjà énormément la gestion des données. Le gros désavantage des fichiers informatiques tels que ceux utilisés dans Excel ou Access est bien entendu que **ces fichiers ne se mettent pas à jour simultanément** pour l'ensemble des utilisateurs, chacun travaillant avec sa propre copie des données, la faisant évoluer à sa guise

Aujourd'hui, nous nous dirigeons donc vers des **bases de données dites « relationnelles »**, stockées sur un ou plusieurs **serveurs centralisés** qui peuvent être accédés par **de nombreux clients** de façon simultanée. Cela permet de rendre l'information disponible en temps réel, **partout et tout le temps**

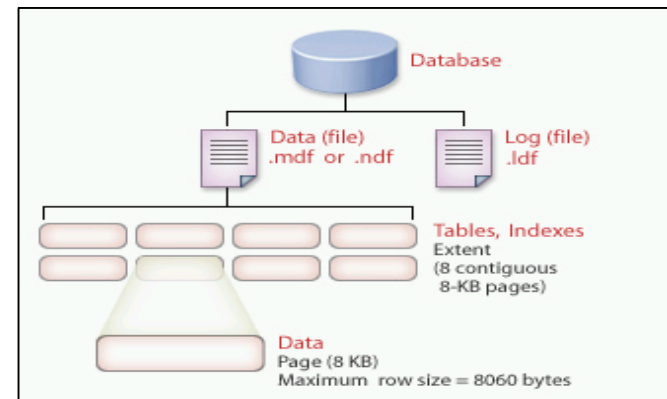
Base de données et SGBD

Un **Système de Gestion de Base de Données (SGBD)** est un programme informatique permettant de gérer des bases de données

Aujourd'hui, nombreux sont les systèmes permettant de créer des bases de données relationnelles. Un SGBD a la caractéristique de venir renforcer une base de données en lui apportant **un certains nombre de fonctionnalités supplémentaires**. C'est le nombre de fonctionnalités, leur efficacité et leur fluidité qui font qu'un SGBD sera plus populaire ou meilleur qu'un autre



Structure logique



Structure physique

Base de données et SGBD

Fonctionnalités principales d'un SGBD :

- **Cohérence des données**

Un SGBD doit garantir que les données contenues dans les bases de données respectent et respecteront toujours les **règles de logique relationnelles établies**

- **Concurrence d'accès aux données**

Lorsque deux utilisateurs essayent d'accéder simultanément aux données, le système doit être capable d'assurer un ordre logique d'exécution des requêtes de chaque utilisateur, **en respectant les règles de transactions** qu'il a établi

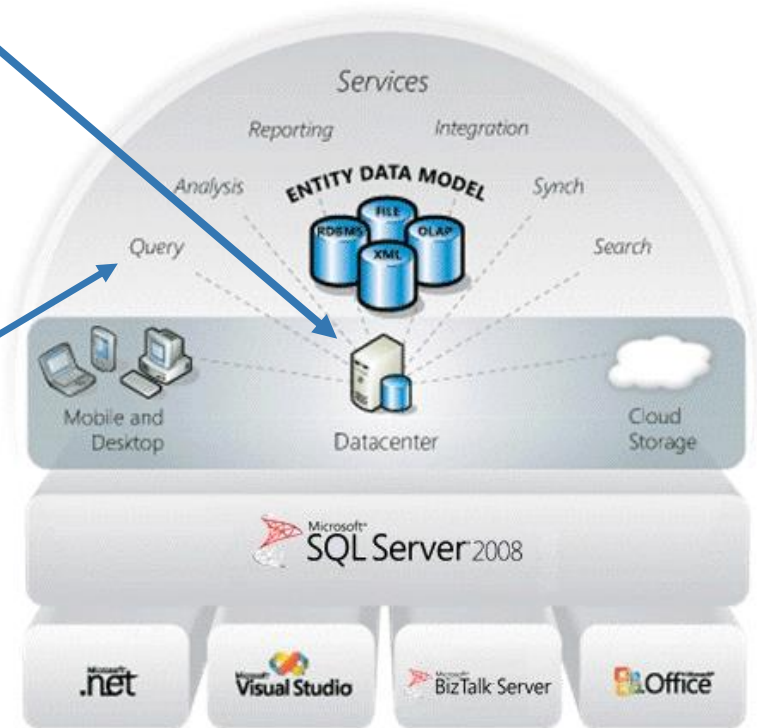
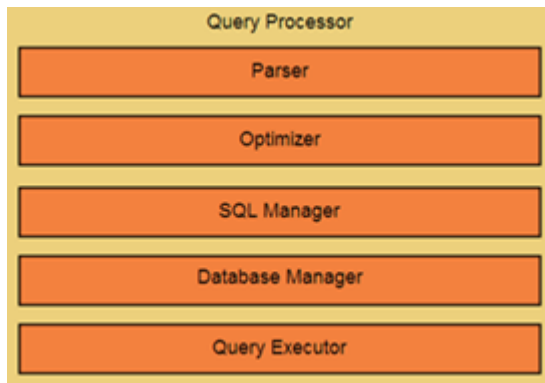
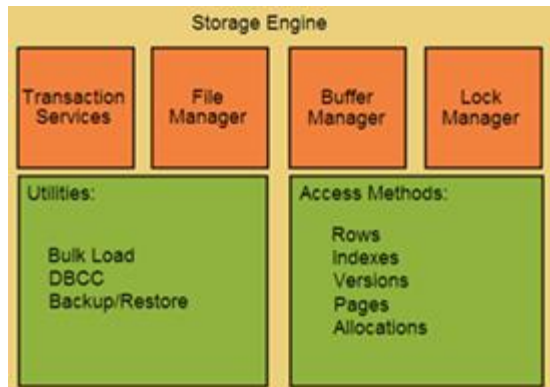
- **Sécurité des données**

Certainement le point le plus délicat qu'un SGBD doit pouvoir gérer, **la sécurité et l'accès aux données** est bien souvent la chose la plus importantes pour les entreprises. Le système doit pouvoir garantir que seuls les utilisateurs autorisés accèderont aux données dont ils ont besoin

- **Pérennité des données**

Avoir des données, c'est bien, les récupérer après un crash, c'est mieux ! Un SGBD doit donner accès à des outils ou des **méthodes de sauvegarde et de récupération** des données afin de pouvoir faire face à n'importe quelle panne logicielle, matérielle ou autre

Base de données et SGBD



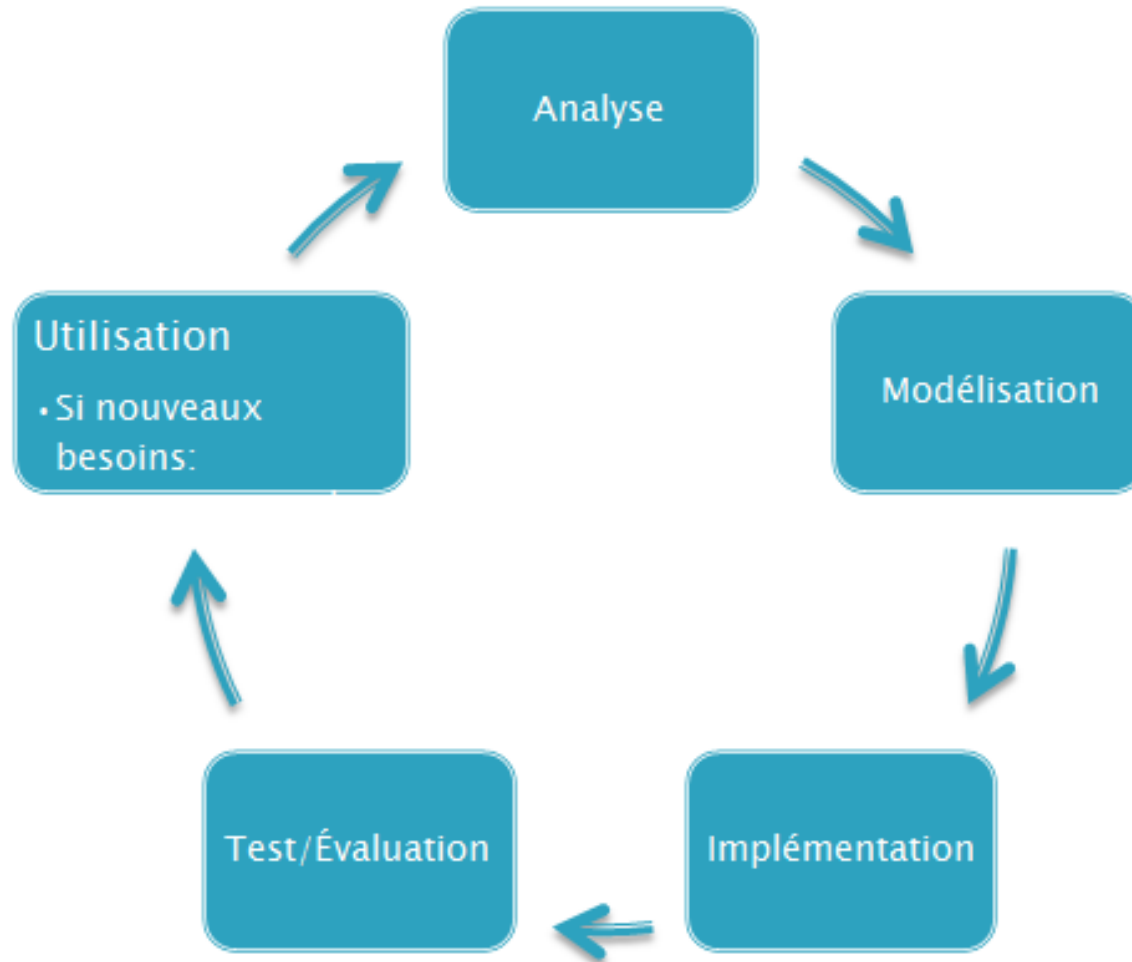
De l'analyse au relationnel

Avant d'arriver à la création de la base de données elle-même, **il est nécessaire d'analyser en profondeur le problème rencontré**. Cette phase d'analyse est nécessaire afin de ne rien oublier et de gagner un temps précieux au niveau du développement et de l'implémentation de la solution

La phase d'analyse passera par **plusieurs étapes** contenant chacune **un certain nombre de schémas et de diagrammes**. Il s'agira la plupart du temps d'appliquer un modèle d'analyse tel que *UML*, dans son intégralité ou partiellement du moins. Ces différents schémas permettront d'établir **le « schéma relationnel » de la base de données**, qui doit permettre aux développeurs de générer la base de données elle-même

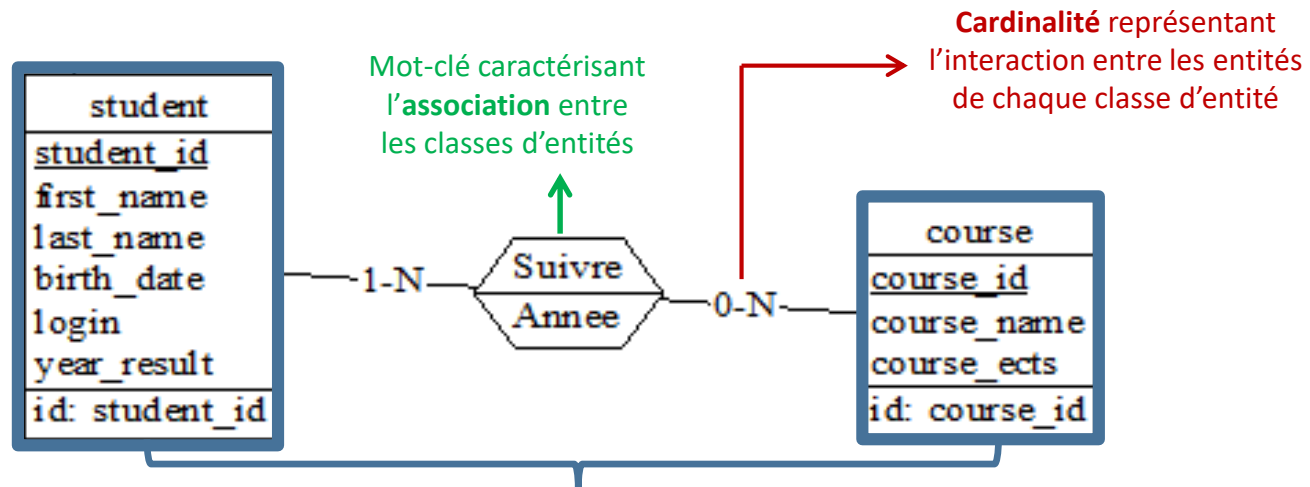
Dans certains cas simples, il est possible de se limiter à deux schémas. Le « **schéma Entités-Associations** » donnera alors directement la possibilité de passer au **schéma relationnel**

De l'analyse au relationnel



De l'analyse au relationnel

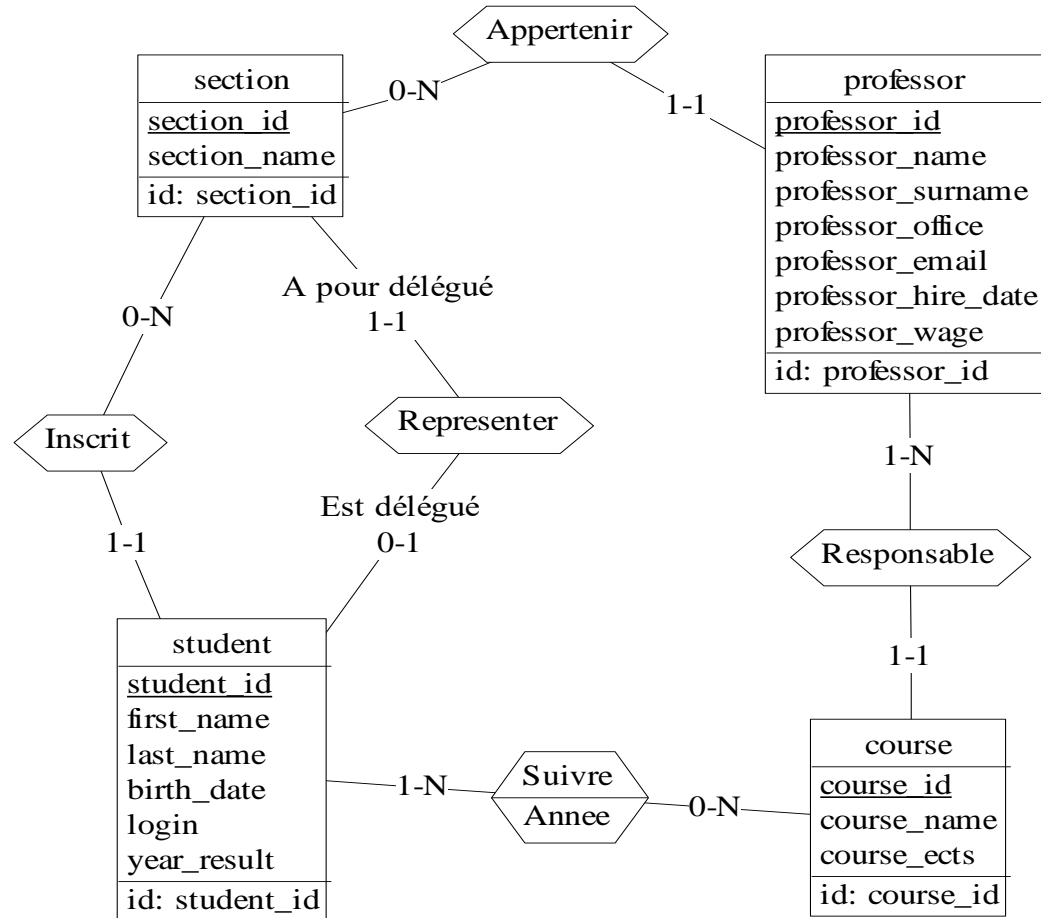
Le schéma entités-associations (EA) modélise simplement chaque acteur (**entité**) d'un système donné, en spécifiant chacun de leurs attributs qu'il est nécessaire d'enregistrer. Il indique **également la façon dont les acteurs interagissent** les uns avec les autres



Classes d'entités reprenant les **attributs** (caractéristiques) de chacune des **entités** (acteurs) du système modélisé

De l'analyse au relationnel

Exemple de
schéma EA
représentant le
système
d'information
d'une université



De l'analyse au relationnel

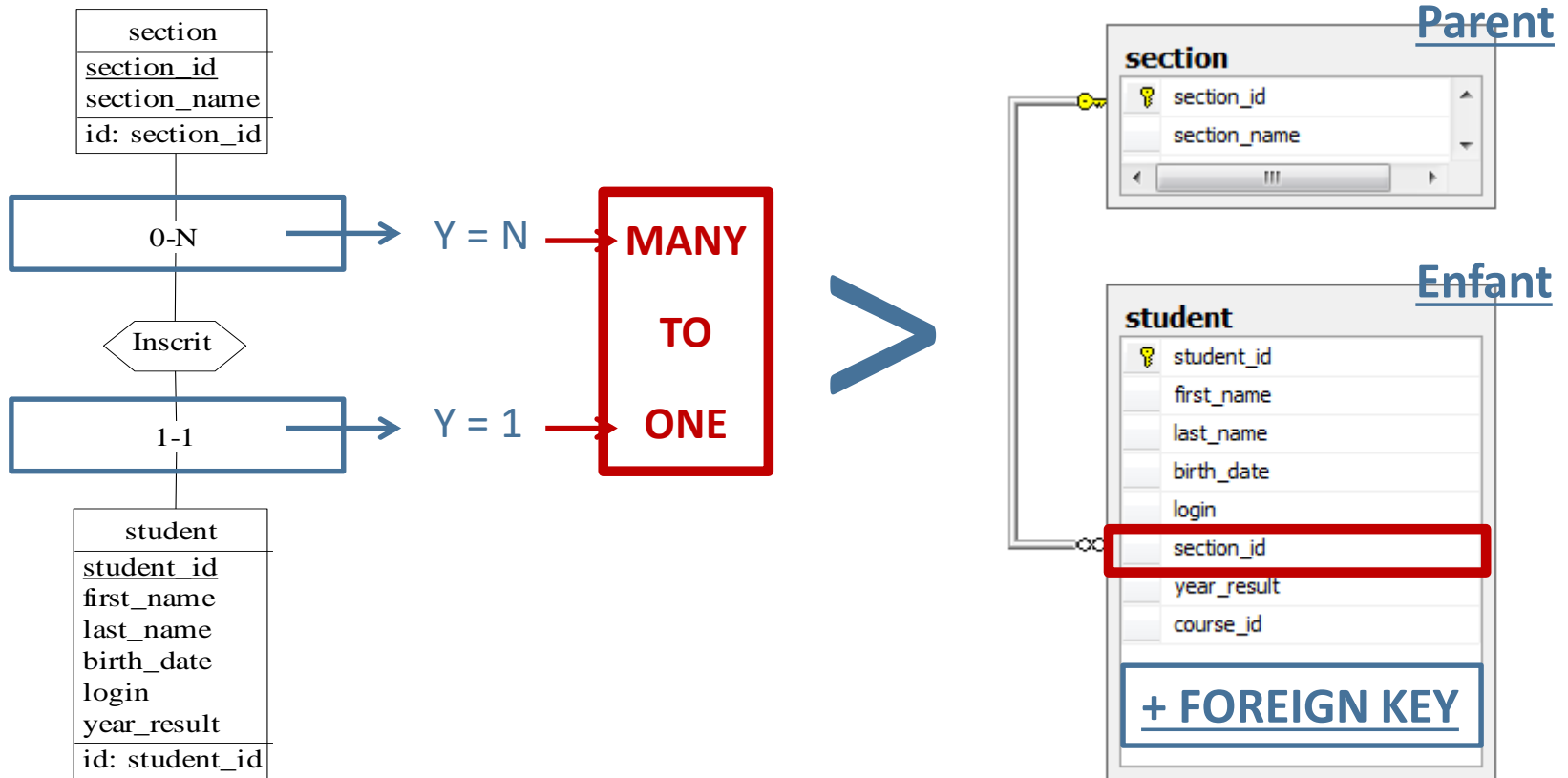
Le schéma relationnel est la traduction du schéma entités-associations
Un schéma relationnel représente le plan de la base de données
Il peut être lu aussi bien par l'analyste que par le programmeur

La traduction du schéma entités-associations se fait en appliquant certaines règles simples dont voici un aperçu succinct :

- Les **classes d'entités** deviennent des **tables**
- Les **attributs** deviennent des **colonnes**
- Les **associations** deviennent des **contraintes d'intégrité référentielles** selon les règles établies dans les slides suivants

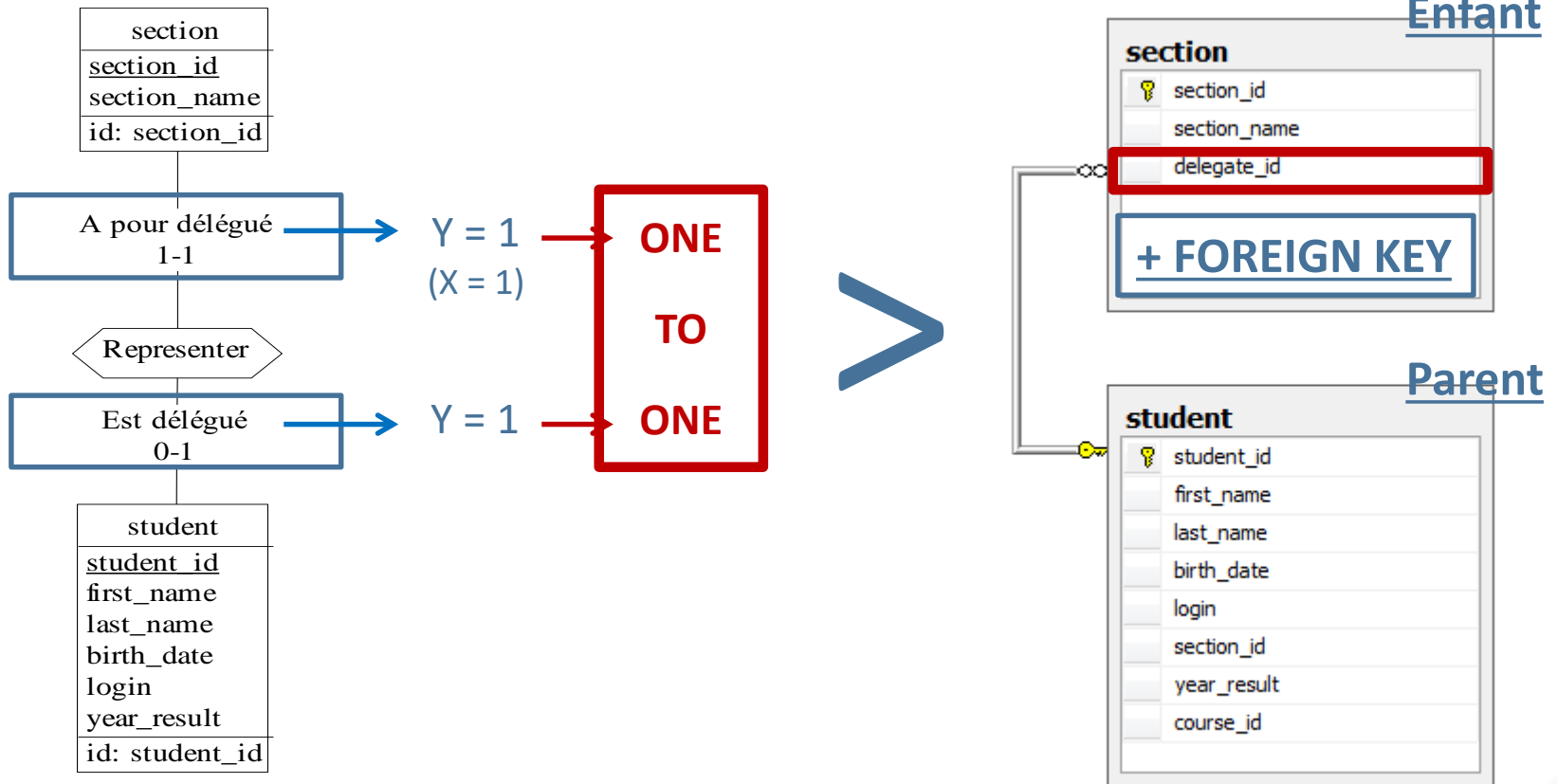
De l'analyse au relationnel

Traduction d'une association de type « One-to-Many » :



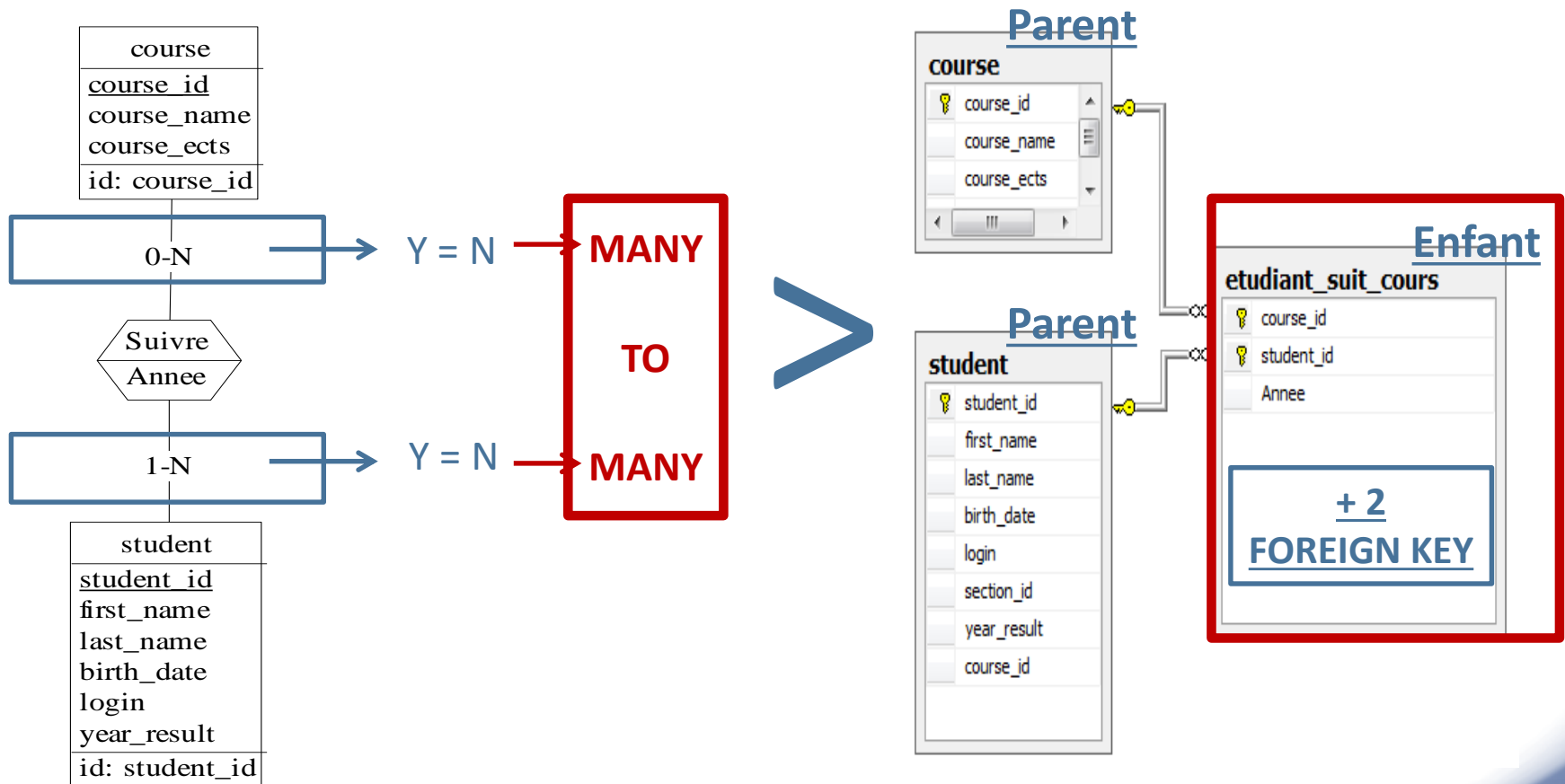
De l'analyse au relationnel

Traduction d'une association de type « One-to-One » :

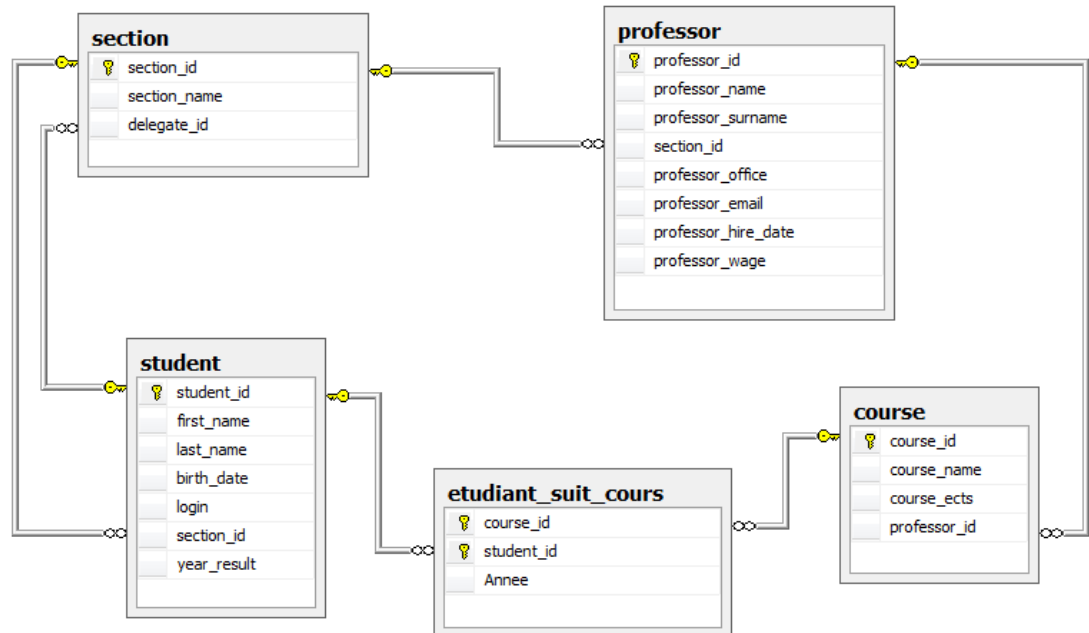
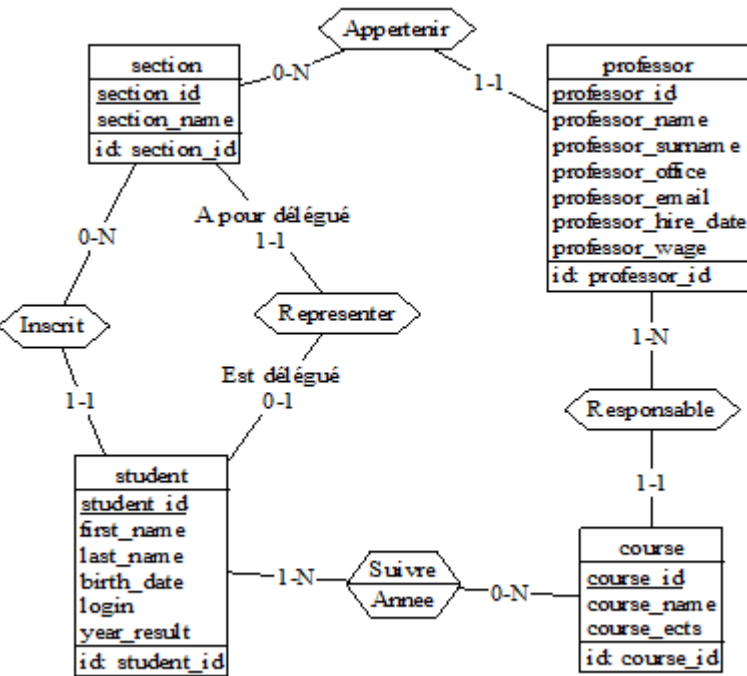


De l'analyse au relationnel

Traduction d'une association de type « Many-to-Many » :

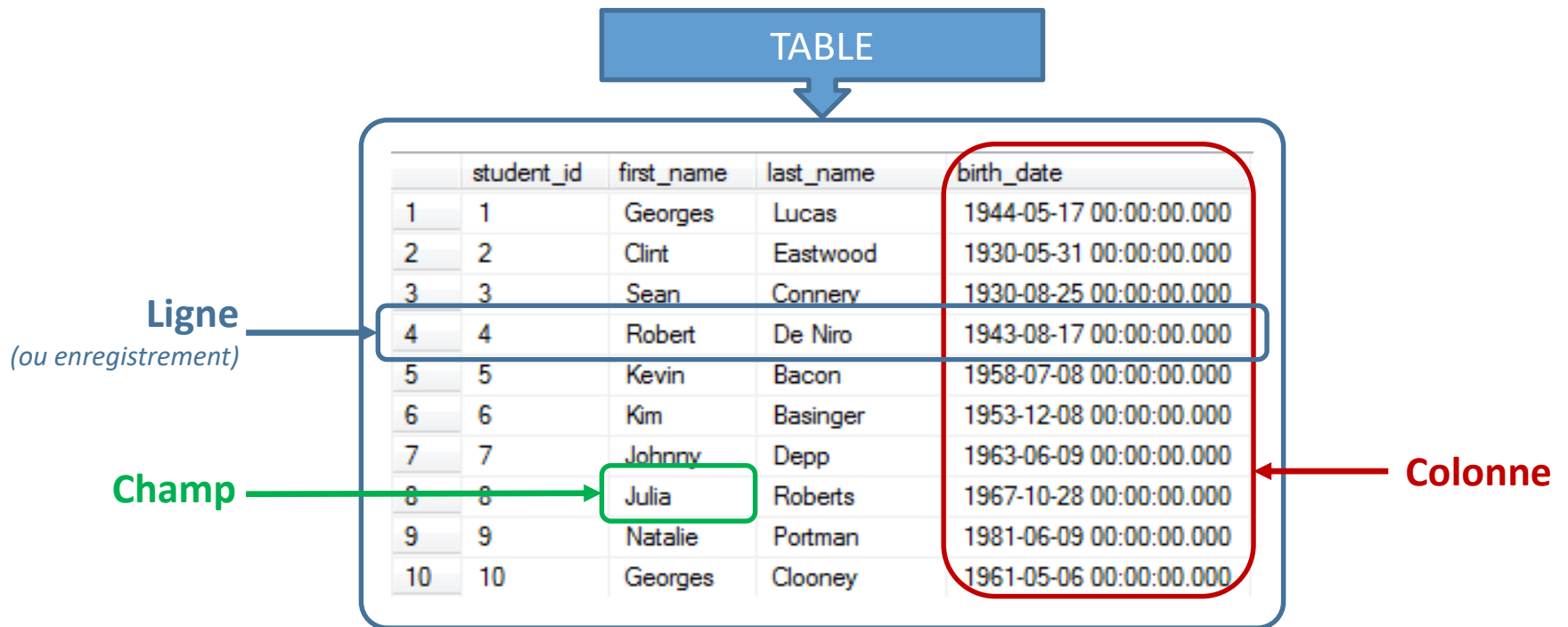


'se au relationnel



Notions de tables

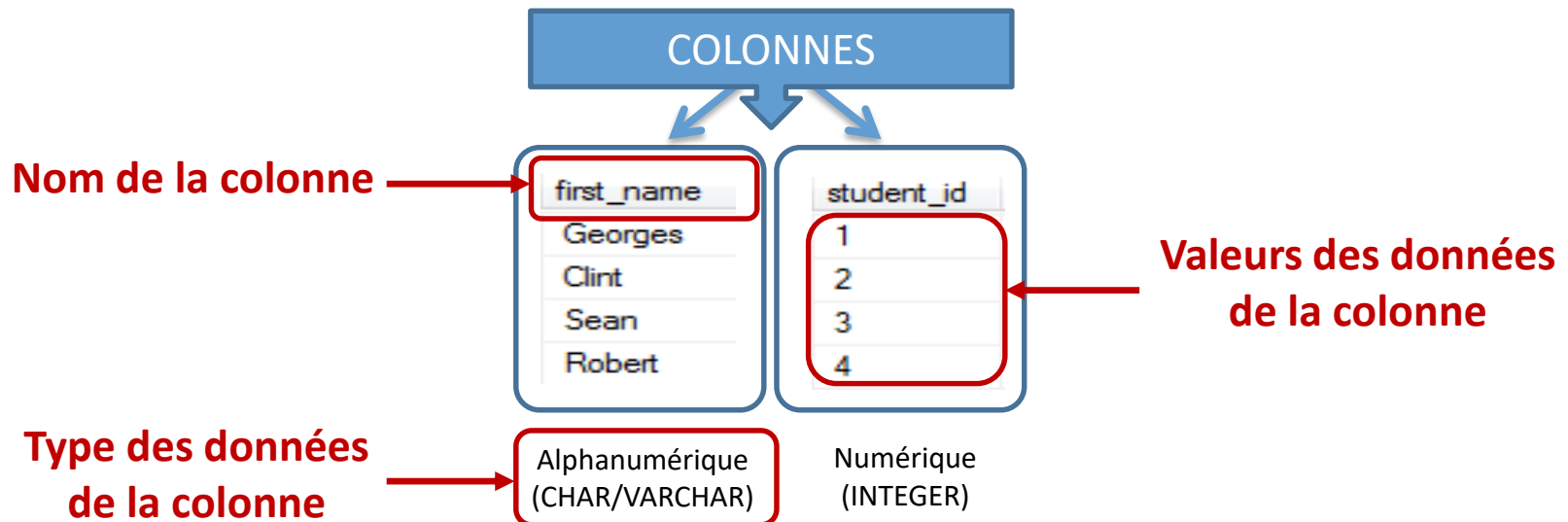
Une **Table** regroupe des **ensembles de données (d'informations)** stockés de façon permanente et décrivant chacun un acteur du système réel modélisé (**Entité**)



La Table étudiant contient des informations sur les étudiants uniquement

Notions de tables

Une **Colonne** est un **attribut** d'une table, elle représente une **caractéristique particulière de l'objet réel** représenté



Contraintes

Une **Contrainte** est objet intégré à une table (*comme les colonnes*)
Une contrainte possède **un nom, un type**
et **concerne au moins une colonne** de la table

On distingue principalement **5 types de contraintes différentes**

Seule la contrainte « **NOT NULL** » n'aura **pas de nom**

Il n'existera au maximum **qu'une seule contrainte de « Clé Primaire »** pour chaque table

Contraintes	Description
NOT NULL	Force la présence d'une valeur (valeur obligatoire)
UNIQUE	Empêche les valeurs-doublons
CHECK	Conditionne les valeurs (expression conditionnelle)
FOREIGN KEY	Conditionne les valeurs par rapport à une autre table

Contraintes : NOT NULL

Ajouter la contrainte « NOT NULL » à la colonne d'une table obligera cette colonne à **contenir une valeur** pour chacune des lignes de la table

Une colonne non-obligatoire est **facultative**, elle peut contenir la valeur **NULL**

La valeur **NULL** spécifie que aucune valeur n'est attribuée

NULL représente l'absence de valeur, elle n'est ni 0 ni une case vide

The diagram shows a table with 9 columns: student_id, first_name, last_name, birth_date, login, section_id, year_result, and course_id. Annotations highlight specific constraints:

- Pas de valeur renseignée**: Points to the empty cell in the first_name column of the first row.
- Colonne facultative acceptant des valeurs NULL**: Points to the year_result column, which contains a NULL value in the sixth row.
- Colonne obligatoire NULL**: Points to the course_id column, which contains 0 values in the fifth, sixth, and eighth rows.

	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	3	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0

Contraintes : **UNIQUE**

La contrainte d'unicité « **UNIQUE** » a la particularité d'empêcher **une colonne** ou **une combinaison de colonnes**, d'accepter deux valeurs identiques pour deux lignes différentes de la table

Colonne **non candidate** contient des **doublons**

Colonne **UNIQUE** aucun doublon

	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	NULL	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0
9	9	Natalie	Portman	1981-06-09 00:00:00.000	nportman	1010	4	EG2210
10	10	Georges	Clooney	1961-05-06 00:00:00.000	gclooney	1020	4	EG2110

Combinaison de colonnes **UNIQUE** aucun doublon

Contraintes : PRIMARY KEY

La contrainte de clé primaire d'une table désigne un ensemble (souvent minimal) de colonnes qui identifient de manière unique les enregistrements d'une table. La combinaison des colonnes qui la composent doit être **UNIQUE** et **NON NULL**.

- Il ne peut exister qu'une et une seule clé primaire par table **MAIS** celle-ci peut être composée d'une combinaison de plusieurs colonnes
- **Une clé primaire est dite « naturelle »** si la ou les colonnes qui la composent sont des attributs représentant réellement une caractéristique de l'objet que la table décrit
- **Une clé primaire sera « artificielle »** si elle est représentée par une colonne dont les valeurs ne représentent rien dans le monde réel. Ces colonnes artificielles sont régulièrement utilisées car elles sont faciles à manipuler et, comme elles ont le plus souvent comme valeurs des nombres, leur contenu peut être géré directement par le SGBD. On dira alors, qu'en plus d'être une clé primaire artificielle, **les valeurs de la colonne sont auto-incrémentées**, le plus souvent de 1 à l'infini

Contraintes : PRIMARY KEY

Combinaison **non candidate**
(contient des valeurs **NULL**)

	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	NULL	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0
9	9	Natalie	Portman	1981-06-09 00:00:00.000	nportman	1010	4	EG2210
10	10	Georges	Clooney	1961-05-06 00:00:00.000	gclooney	1020	4	EG2110

PRIMARY KEY
dite « **artificielle** »
(**PEUT** être auto-incrémentée)

Combinaison **UNIQUE ET NOT NULL**
candidate à la création d'une
PRIMARY KEY « **naturelle** »

Contraintes : **FOREIGN KEY**

La contrainte de clé étrangère d'une table (**enfant**) permet d'éviter la redondance d'information au niveau de cette table par le biais d'une colonne de référence pointant vers une colonne identifiante d'une autre table (**parent**)
Contenant, elle, le détail de l'objet référencé

- La clé étrangère est une contrainte de la table, elle concerne une ou plusieurs colonnes de la table, mais ces colonnes ne sont pas implicitement créées lors de la création de la clé étrangère. De la même manière, modifier ou supprimer la clé étrangère ne modifie pas les caractéristiques de la colonne concernée
- Il peut exister plusieurs clés étrangères dans chaque table
- Plusieurs colonnes peuvent composer la même clé étrangère d'une table, si cette clé étrangère fait référence à plusieurs colonnes d'une autre table (clé primaire composite, clé d'unicité composite)
- Les colonnes référencées entre elles doivent être du même type
- Les valeurs d'une colonne utilisée comme clé étrangère peuvent être **NULL**

Contraintes : FOREIGN KEY

Table Livres

Information redondante

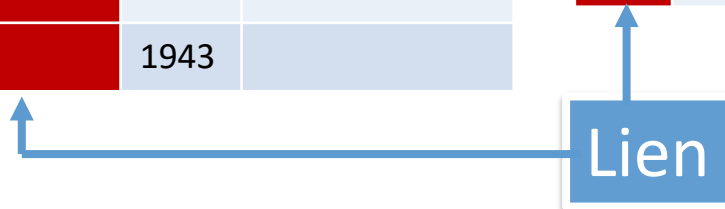
Titre	PrenomAuteur	NomAuteur	PseudoAuteur	NaissAuteur	DateLivre	ISBN
La Peste	Albert	Camus	Bebert	07/11/1913	1974	978-20703604
La Chute	Albert	Camus	Bebert	07/11/1913	1973	978-20703109
Huis-clos	Jean-Paul	Sartre	Jy-Pé	21/06/1905	1943	

Table Livres

Titre	RefAuteur	Date	ISBN
La Peste	1	1974	978-20703604
La Chute	1	1973	978-20703109
Huis-clos	2	1943	

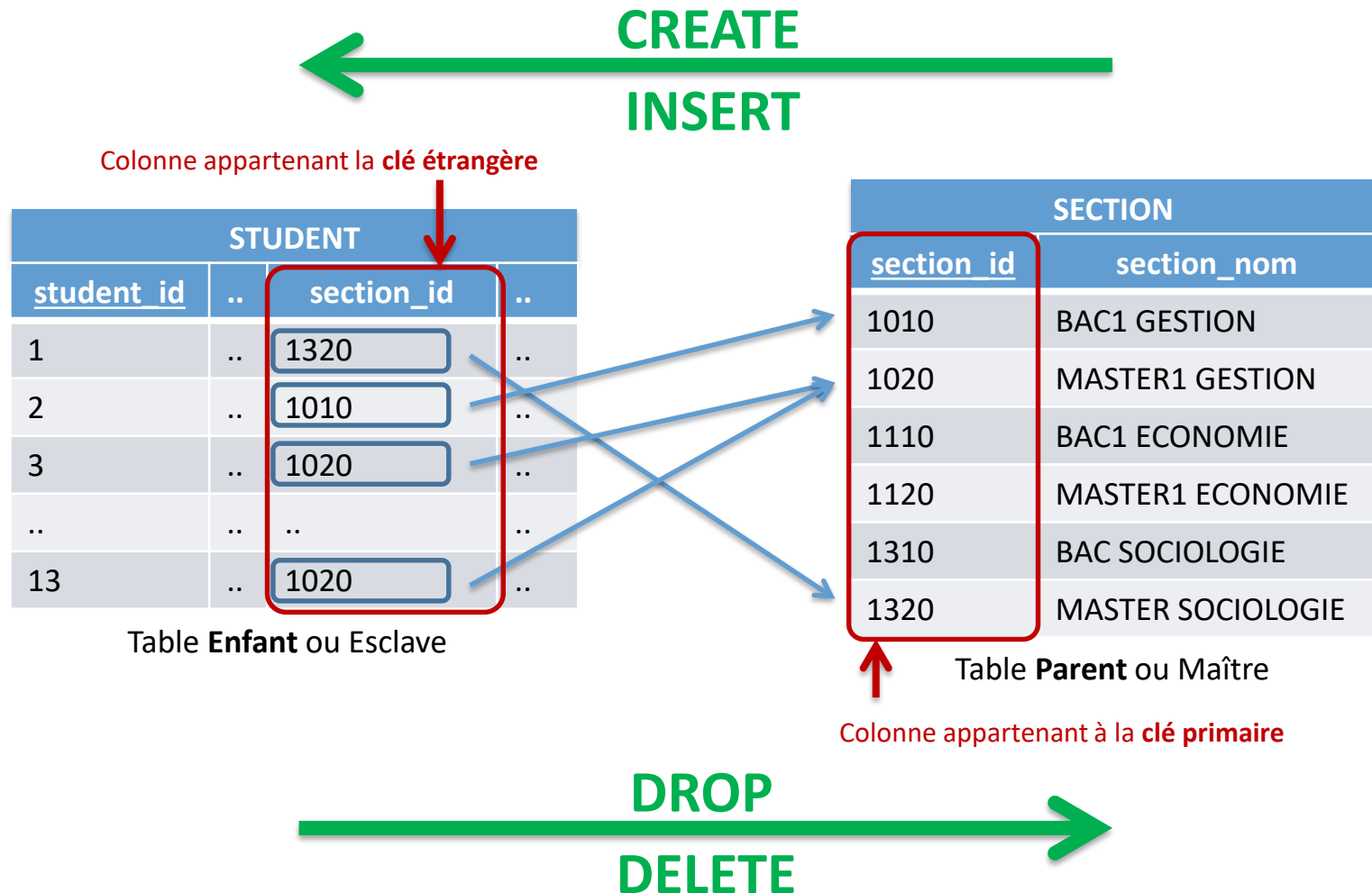
Table Auteurs

Ref	Nom	Prenom	Pseudo	DateNaiss
1	Albert	Camus	Bebert	07/11/1913
2	Jean-Paul	Sartre	Jy-Pé	21/06/1905



La table **Livres** est liée à la table **Auteurs** par le champ **RefAuteur**.

Contraintes : FOREIGN KEY



Contraintes : CHECK

La contrainte **CHECK** d'une table permet de poser une condition spécifique sur les colonnes de la tables, afin d'y empêcher l'insertion de n'importe quelle valeur

	student_id	first_name	last_name	birth_date	login	section_id	year_result	course_id
1	1	Georges	Lucas	1944-05-17 00:00:00.000	glucas	1320	10	EG2210
2	2	Clint	Eastwood	1930-05-31 00:00:00.000	ceastwoo	1010	4	EG2210
3	3	Sean	Connery	1930-08-25 00:00:00.000	sconnery	1020	12	EG2110
4	4	Robert	De Niro	1943-08-17 00:00:00.000	rde niro	1110	NULL	EG2210
5	5	Kevin	Bacon	1958-07-08 00:00:00.000	kbacon	1120	16	0
6	6	Kim	Basinger	1953-12-08 00:00:00.000	kbasinge	1310	NULL	0
7	7	Johnny	Depp	1963-06-09 00:00:00.000	jdepp	1110	11	EG2210
8	8	Julia	Roberts	1967-10-28 00:00:00.000	jroberts	1120	17	0
9	9	Natalie	Portman	1981-06-09 00:00:00.000	nportman	1010	4	EG2210
10	10	Georges	Clooney	1961-05-06 00:00:00.000	gclooney	1020	4	EG2110

$0 \leq \text{year_result} \leq 20$

$\text{Birth_date} > 01-01-1930$