

# Bases de données

## Cours 2

### Conception et premiers éléments de SQL

Marie Pelleau & Laurent Tichit  
[marie.pelleau@univ-cotedazur.fr](mailto:marie.pelleau@univ-cotedazur.fr),  
[laurent.tichit@univ-cotedazur.fr](mailto:laurent.tichit@univ-cotedazur.fr)

8 novembre 2022

- 1 Rappels : modèle relationnel
- 2 Méthode de conception
- 3 SQL

# Résumé sur le modèle relationnel (1)

- Les systèmes de gestion de bases de données relationnelles sont actuellement les logiciels de traitement de données les plus fréquemment utilisés.
- Les relations (entités comme associations) sont représentées de manière physique par des **tables**, dont les lignes correspondent aux  $n$ -uplets individuels et les colonnes aux attributs.

## Résumé sur le modèle relationnel (2)

- Les propriétés d'une base de données sont les suivantes :
  - chaque cellule contient exactement une valeur atomique
  - les noms d'attributs sont distincts les uns des autres
  - l'ordre des attributs est immatériel
  - l'ordre des  $n$ -uplets est immatériel
  - il n'existe pas de  $n$ -uplets en double
- Dans un  $n$ -uplet, un NULL représente une valeur d'un attribut inconnue à l'heure actuelle ou qui ne s'applique pas à ce  $n$ -uplet

## Résumé sur le modèle relationnel (3)

- Une **clé candidate** est un ensemble minimum d'attributs qui identifie les  $n$ -uplets d'une relation de façon unique
- Une **clé primaire** est la clé candidate choisie pour servir à l'identification de  $n$ -uplets
- Une **clé étrangère** est un ensemble d'attributs au sein d'une relation qui constitue une clé candidate d'une autre relation
- l'**intégrité d'entité** établit que, dans une relation de base, aucun attribut faisant partie de la clé primaire ne peut être NULL
- l'**intégrité référentielle** établit que les valeurs d'une clé étrangère doivent correspondre à une valeur d'une clé candidate dans la relation de référence ou être complètement NULL

## 1 Rappels : modèle relationnel

## 2 Méthode de conception

- Dictionnaire de données
- Modèle conceptuel
  - Étapes
  - Entités
  - Associations
  - Exemple : bibliothèque
- Modèle relationnel

## 3 SQL

- Objectif : à partir d'un problème concret, existant (exemple : gestion d'une bibliothèque), concevoir un schéma relationnel pertinent.
- Méthode de conception présentée ici : formalisme Entités-Associations.

- Objectif : à partir d'un problème concret, existant (exemple : gestion d'une bibliothèque), concevoir un schéma relationnel pertinent.
- Méthode de conception présentée ici : formalisme Entités-Associations.
- D'autres méthodes possibles : langage UML (diagramme de classes).



# Description générale

Étapes de la méthode :

- Analyse de l'existant.
  - Analyser système d'informations pour en donner une description aussi précise que possible.
  - Recenser les informations nécessaires, décrire leur organisation.

# Description générale

## Étapes de la méthode :

- Analyse de l'existant.
  - Analyser système d'informations pour en donner une description aussi précise que possible.
  - Recenser les informations nécessaires, décrire leur organisation.
- Modélisation de la BD.
  - Définition du dictionnaire de données.
  - Schéma conceptuel (entités-associations)
  - Transformation en schéma relationnel.

# Description générale

## Étapes de la méthode :

- Analyse de l'existant.
  - Analyser système d'informations pour en donner une description aussi précise que possible.
  - Recenser les informations nécessaires, décrire leur organisation.
- Modélisation de la BD.
  - Définition du dictionnaire de données.
  - Schéma conceptuel (entités-associations)
  - Transformation en schéma relationnel.
- Implantation de la BD.
  - Définitions des clés, des contraintes, des index.
  - Création des requêtes, des interfaces, des applications.

# Dans le cours d'aujourd'hui

- Analyse de l'existant : supposée déjà faite.
- Implantation de la BD : dans les cours suivants.
- Modélisation de la BD : aujourd'hui !

Rappel des étapes :

- Définition du dictionnaire de données.
- Schéma conceptuel (entités-associations).
- Transformation en schéma relationnel.

# Dictionnaire de données

Le dictionnaire de données recense la liste des informations qui devront être stockées dans la base de données.

Chaque unité d'information est un **attribut** (champ).

Pour chaque attribut, on précise :

# Dictionnaire de données

Le dictionnaire de données recense la liste des informations qui devront être stockées dans la base de données.

Chaque unité d'information est un **attribut** (champ).

Pour chaque attribut, on précise :

- un **libellé** (nom de l'attribut). Utiliser des libellés cohérents (mêmes règles de nommage) et éviter les libellés trop longs (par souci de concision) ;

# Dictionnaire de données

Le dictionnaire de données recense la liste des informations qui devront être stockées dans la base de données.

Chaque unité d'information est un **attribut** (champ).

Pour chaque attribut, on précise :

- un **libellé** (nom de l'attribut). Utiliser des libellés cohérents (mêmes règles de nommage) et éviter les libellés trop longs (par souci de concision) ;
- un **type** (par exemple : nombre réel, nombre entier, chaîne de caractères, date, etc.) et une **taille** (par exemple : chaîne de 10 caractères) ;



# Dictionnaire de données

Le dictionnaire de données recense la liste des informations qui devront être stockées dans la base de données.

Chaque unité d'information est un **attribut** (champ).

Pour chaque attribut, on précise :

- un **libellé** (nom de l'attribut). Utiliser des libellés cohérents (mêmes règles de nommage) et éviter les libellés trop longs (par souci de concision) ;
- un **type** (par exemple : nombre réel, nombre entier, chaîne de caractères, date, etc.) et une **taille** (par exemple : chaîne de 10 caractères) ;
- une **description** (sémantique, domaine de valeurs, méthode de calcul, etc.).

## Exemple : bibliothèque

- Description de l'existant :
  - Chaque abonné a un numéro d'abonné unique, un nom, un prénom, une adresse et une date d'abonnement.
  - Les livres ont tous un numéro ISBN, un titre, un éditeur et une année de publication.
  - Les auteurs qui écrivent les livres sont identifiés par un numéro d'auteur, et on stocke leur nom et leur prénom.
  - Lorsqu'un abonné réalise un emprunt d'un livre, on enregistre le numéro et la date de l'emprunt.
  - Lorsqu'il le restitue, on mémorise la date de retour.

# Identifier les informations à stocker

## Attributs des entités

- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.

# Identifier les informations à stocker

## Attributs des entités

- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.
- Les **livres** ont tous un *numéro ISBN*, un *titre*, un *éditeur* et une *année de publication*.

# Identifier les informations à stocker

## Attributs des entités

- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.
- Les **livres** ont tous un *numéro ISBN*, un *titre*, un *éditeur* et une *année de publication*.
- Les **auteurs** qui écrivent les **livres** sont identifiés par un *numéro d'auteur*, et on stocke leur *nom* et *prénom*.

# Identifier les informations à stocker

## Attributs des entités

- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.
- Les **livres** ont tous un *numéro ISBN*, un *titre*, un *éditeur* et une *année de publication*.
- Les **auteurs** qui écrivent les **livres** sont identifiés par un *numéro d'auteur*, et on stocke leur *nom* et *prénom*.
- Lorsqu'un **abonné** réalise un **emprunt** d'un **livre**, on enregistre le *numéro* et la *date de l'emprunt*.

# Identifier les informations à stocker

## Attributs des entités

- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.
- Les **livres** ont tous un *numéro ISBN*, un *titre*, un *éditeur* et une *année de publication*.
- Les **auteurs** qui écrivent les **livres** sont identifiés par un *numéro d'auteur*, et on stocke leur *nom* et *prénom*.
- Lorsqu'un **abonné** réalise un **emprunt** d'un **livre**, on enregistre le *numéro* et la *date de l'emprunt*.
- Lorsqu'il le restitue, on mémorise la *date de retour*.

# Dictionnaire de données

On le résume dans un tableau.

Libellé	Type	Description
NumAbo	entier	Numéro de l'abonné
NomAbo	car(20)	Nom de l'abonné
PrénomAbo	car(20)	Prénom de l'abonné
AdrAbo	car(80)	Adresse de l'abonné
DateAbo	date	Date de l'abonnement (AAAA-MM-JJ)
NumAut	entier	Numéro de l'auteur
NomAut	car(20)	Nom de l'auteur
PrénomAut	car(20)	Prénom de l'auteur
ISBN	car(13)	Code ISBN identifiant un livre
Titre	car(80)	Titre du livre
Editeur	car(20)	Nom de l'editeur
Année	entier	Année de publication
NumEmp	entier	Numéro d'emprunt
DateEmp	date	Date de l'emprunt d'un livre par un abonné
DateRet	date	Date de retour d'un livre emprunté par un abonné



# Définition des attributs

- Relation PERSONNE.

Nom	Prénom	Adresse
Durand	Serge	Rue Barbe, 69000 Lyon
Lamotte	Pierre	Rue Minant, 06000 Nice

# Définition des attributs

- Relation PERSONNE.

Nom	Prénom	Adresse
Durand	Serge	Rue Barbe, 69000 Lyon
Lamotte	Pierre	Rue Minant, 06000 Nice

- Si on doit fréquemment rechercher des personnes par ville, il faut décomposer les attributs pour améliorer l'efficacité.

Nom	Prénom	Rue	CodePostal	Ville
Durand	Serge	Rue Barbe	69000	Lyon
Lamotte	Pierre	Rue Minant	06000	Nice

# Définition des attributs

- Relation PERSONNE.

Nom	Prénom	Adresse
Durand	Serge	Rue Barbe, 69000 Lyon
Lamotte	Pierre	Rue Minant, 06000 Nice

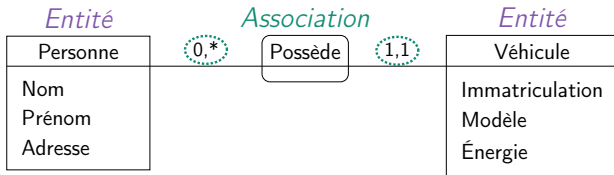
- Si on doit fréquemment rechercher des personnes par ville, il faut décomposer les attributs pour améliorer l'efficacité.

Nom	Prénom	Rue	CodePostal	Ville
Durand	Serge	Rue Barbe	69000	Lyon
Lamotte	Pierre	Rue Minant	06000	Nice

- Une telle opération est compliquée et risquée lorsque les données ont déjà été saisies.

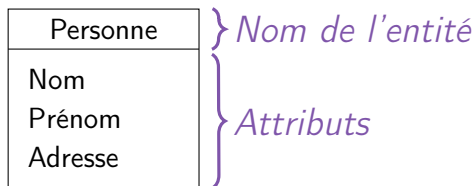
- Après avoir défini le dictionnaire de données et donc les attributs, la deuxième phase consiste à définir le schéma conceptuel (MCD : « modèle conceptuel des données »).
- Il va permettre de représenter la **structure de la base de données**.

- Après avoir défini le dictionnaire de données et donc les attributs, la deuxième phase consiste à définir le schéma conceptuel (MCD : « modèle conceptuel des données »).
- Il va permettre de représenter la **structure de la base de données**.
- Le formalisme présenté est le modèle Entités-Associations (E-A) qui utilise une représentation graphique (un diagramme).
- Exemple de schéma entités-associations.



# Entités

- Une **entité** est un concept qui décrit un ensemble d'objets d'un même type. Par exemple : des personnes, des véhicules, des livres, des cours, des séances de cinéma, ...
- Une entité est définie par son **nom** et ses **attributs**. Les attributs (ou champs) sont les propriétés qui caractérisent l'entité.
- Les entités seront représentées par des rectangles dans le diagramme.
- Exemple d'entité.



# Entités, tables et occurrences

- Chaque entité sera une table (relation) de la base de données.
- Les lignes (occurrences) d'une table sont les données décrivant les objets du monde réel : les *n-uplets* (ou *tuples*).

# Entités, tables et occurrences

- Chaque entité sera une table (relation) de la base de données.
- Les lignes (occurrences) d'une table sont les données décrivant les objets du monde réel : les *n-uplets* (ou *tuples*).
- Exemple : les occurrences de la table PERSONNE sont les personnes.

Nom	Prénom	Adresse
Dupond	Jean	Rue Minant, 06013 Nice
Martin	Marie	Rue Tilant, 06200 Cannes
Durand	Albert	Rue Barbe, 06600 Antibes
Dupond	Jean	Rue Pestre, 06010 Nice
Lavriotte	Jacques	Rue Ade, 06008 Nice

*n-uplet*



# Entités, tables et occurrences

- Chaque entité sera une table (relation) de la base de données.
- Les lignes (occurrences) d'une table sont les données décrivant les objets du monde réel : les *n-uplets* (ou *tuples*).
- Exemple : les occurrences de la table PERSONNE sont les personnes.

Nom	Prénom	Adresse
Dupond	Jean	Rue Minant, 06013 Nice
Martin	Marie	Rue Tilant, 06200 Cannes
Durand	Albert	Rue Barbe, 06600 Antibes
Dupond	Jean	Rue Pestre, 06010 Nice
Lavriotte	Jacques	Rue Ade, 06008 Nice

*n-uplet*

- Un n-uplet contient une valeur (une donnée) pour chaque attribut (chaque colonne) de la table.
- Valeur NULL attribuée quand elle n'est pas connue. Exemple : pour une personne dont on ne connaît pas l'adresse.

# Entités, tables et occurrences

- Chaque entité sera une table (relation) de la base de données.
- Les lignes (occurrences) d'une table sont les données décrivant les objets du monde réel : les *n-uplets* (ou *tuples*).
- Exemple : les occurrences de la table PERSONNE sont les personnes.

Nom	Prénom	Adresse
Dupond	Jean	Rue Minant, 06013 Nice
Martin	Marie	Rue Tilant, 06200 Cannes
Durand	Albert	Rue Barbe, 06600 Antibes
Dupond	Jean	Rue Pestre, 06010 Nice
Lavriotte	Jacques	Rue Ade, 06008 Nice

*n-uplet*

- Un n-uplet contient une valeur (une donnée) pour chaque attribut (chaque colonne) de la table.
- Valeur NULL attribuée quand elle n'est pas connue. Exemple : pour une personne dont on ne connaît pas l'adresse.
- Rappel : *domaine* est l'ensemble des valeurs que peut prendre un attribut. Exemple :  $\text{Âge} \in \{0, 1, \dots, 122\}$ ,  $\text{Prix} \in [0.00, 1500.00]$ .

# Identifiants

- Chaque n-uplet (ligne) d'une entité doit être identifié de manière unique par un (des) attribut(s) qui constitue(nt) la **clé primaire** de l'entité.

# Identifiants

- Chaque n-uplet (ligne) d'une entité doit être identifié de manière unique par un (des) attribut(s) qui constitue(nt) la **clé primaire** de l'entité.
- Exemple : relation VÉHICULE.

Immatriculation	Modèle	Énergie
AA 229 AA	308	Essence
AB 329 TZ	Clio	Gazole
4215 VD 06	Golf	Gazole
CT 293 TZ	Zoé	Électrique

À chaque numéro d'immatriculation correspond un unique véhicule. L'attribut immatriculation est une **clé candidate** (minimale), elle constituera la clé primaire.

# Identifiants (suite)

- Mais il n'est pas toujours possible de trouver un tel sous-ensemble d'attributs.
- Relation PERSONNE.

Nom	Prénom	Âge
Dupond	Jean	55
Marin	Marie	41
Durand	Albert	64
Dupond	Jean	55
Lavirotte	Jacques	72

# Identifiants (suite)

- Mais il n'est pas toujours possible de trouver un tel sous-ensemble d'attributs.
- Relation PERSONNE.

Nom	Prénom	Âge
Dupond	Jean	55
Marin	Marie	41
Durand	Albert	64
Dupond	Jean	55
Lavirotte	Jacques	72

- Plusieurs personnes peuvent avoir les mêmes nom, prénom et âge. Ici, comment distinguer les Jean Dupond qui ont 55 ans ?

## Identifiants (suite)

- En ce cas, on crée un attribut **Numéro de la personne**. Un numéro unique sera affecté à chaque personne.
- Exemple : relation PERSONNE avec identifiant.

Numéro	Nom	Prénom	Âge
1	Dupond	Jean	55
2	Marin	Marie	41
3	Durand	Albert	64
4	Dupond	Jean	55
5	Lavirotte	Jacques	72

## Identifiants (suite)

- En ce cas, on crée un attribut **Numéro de la personne**. Un numéro unique sera affecté à chaque personne.
- Exemple : relation PERSONNE avec identifiant.

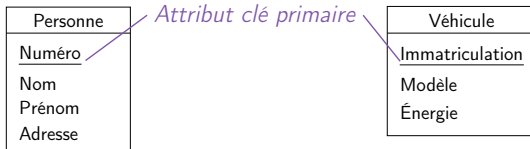
Numéro	Nom	Prénom	Âge
1	Dupond	Jean	55
2	Marin	Marie	41
3	Durand	Albert	64
4	Dupond	Jean	55
5	Lavirotte	Jacques	72

- Certains SGBD proposent un type d'attribut spécifique à cet usage (NumeroAuto, Serial, ...).
- Le SGBD gère alors de manière autonome et transparente la numérotation.



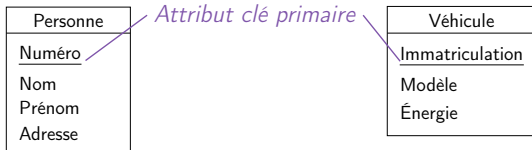
# Clé primaire

- On représente la clé primaire en soulignant le ou les attribut(s) dans l'entité.

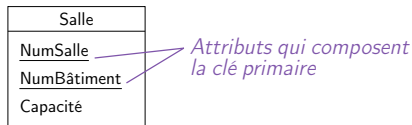


# Clé primaire

- On représente la clé primaire en soulignant le ou les attribut(s) dans l'entité.



- La clé primaire peut être composée de plusieurs attributs (champs). Exemple : les salles de plusieurs bâtiments portent le même numéro (101, 102, ...). L'identifiant de l'entité Salle sera composé des deux champs NumSalle et NumBâtiment.



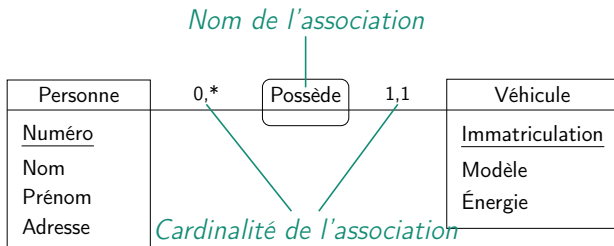
- Pas de limite au nombre d'attributs, mais coûteux d'en avoir beaucoup.

# Associations

- Les **associations** sont les liens entre plusieurs entités représentant un lien entre les concepts.
- On représente les associations par des rectangles arrondis.

# Associations

- Les **associations** sont les liens entre plusieurs entités représentant un lien entre les concepts.
- On représente les associations par des rectangles arrondis.
- Exemple : une personne possède une voiture.



# Cardinalité des associations

- Les cardinalités des associations donnent le nombre minimal et maximal d'uplets de chaque entité.
- Il existe plusieurs types de cardinalité (notées 1,1 ; 0,1 ; 1,\* ; 0,\*).

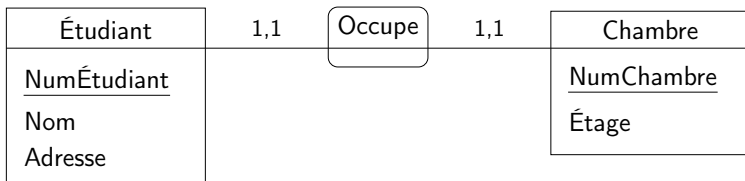
# Cardinalité des associations

- Les cardinalités des associations donnent le nombre minimal et maximal d'uplets de chaque entité.
- Il existe plusieurs types de cardinalité (notées 1,1 ; 0,1 ; 1,\* ; 0,\*).
- Association « un à un »**

Occupation des chambres universitaires.

- Un étudiant occupe une seule chambre universitaire.
- Une chambre est occupée par un seul étudiant.

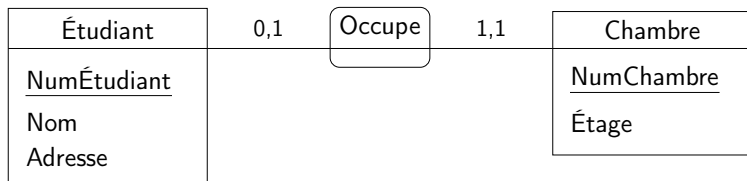
Eric Dupond ● — ● Chambre 15  
 Marc Ducros ● — ● Chambre 37  
 Franck Dupont ● — ● Chambre 02



# Association 1 à 1 (suite)

- La cardinalité minimale d'une association « un à un » peut être zéro.
- Occupation des chambres universitaires (suite).
  - Un étudiant occupe ou non une seule chambre universitaire.
  - Une chambre est occupée par un et un seul étudiant.

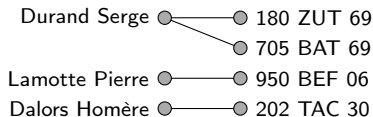
Eric Dupond ● — ● Chambre 15  
Marc Ducros ● — ● Chambre 37  
Franck Dupont ● — ● Chambre 02  
Tom Bombadil ●



# Association « un à plusieurs »

Exemple : propriétaires de véhicules.

- Une personne peut posséder un à plusieurs véhicules.
- Un véhicule appartient à une et une seule personne.

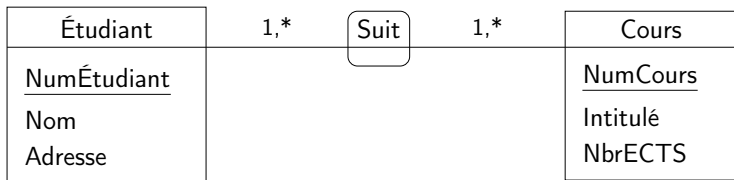
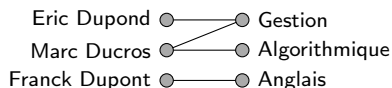




# Association « plusieurs à plusieurs »

Exemple : cours suivis par les étudiants.

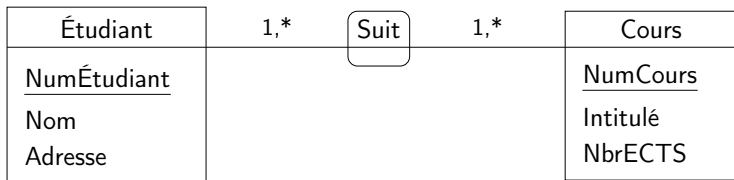
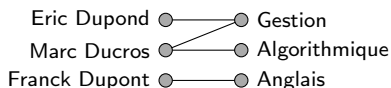
- Un étudiant suit un à plusieurs cours.
- Un cours est suivi par un à plusieurs étudiants.



# Association « plusieurs à plusieurs »

Exemple : cours suivis par les étudiants.

- Un étudiant suit un à plusieurs cours.
- Un cours est suivi par un à plusieurs étudiants.



- Zéro ou un ? La cardinalité minimale d'une association « un à plusieurs » ou « plusieurs à plusieurs » peut être zéro.
- C'est au moment de l'implantation que les cardinalités minimales de types 0,1 ou 0,\* seront considérées (propriétés NULL des clés).

# Attributs des associations

Les associations « plusieurs à plusieurs » peuvent posséder des attributs et ce sont les seules.

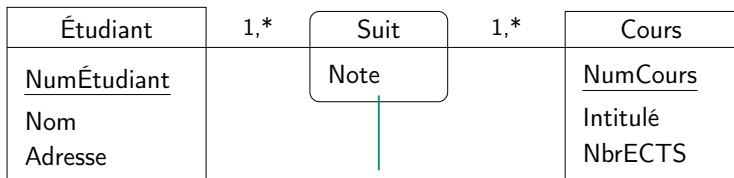
Exemple : cours suivis par les étudiants et notes.

Un étudiant a une note pour chaque cours qu'il suit.

Eric Dupond — 12 — Gestion

Marc Ducros — 9 — Algorithmique

Franck Dupont — 11 — Anglais



*Attribut de l'association*

# Identification des entités et des associations

- Les entités sont les concepts (c'est-à-dire les types d'objets manipulés). Exemples : les **personnes** possèdent des **véhicules**, les **étudiants** suivent des **cours**.

# Identification des entités et des associations

- Les entités sont les concepts (c'est-à-dire les types d'objets manipulés). Exemples : les **personnes** possèdent des **véhicules**, les **étudiants** suivent des **cours**.
- Les noms des entités sont toujours au singulier.

# Identification des entités et des associations

- Les entités sont les concepts (c'est-à-dire les types d'objets manipulés). Exemples : les **personnes** possèdent des **véhicules**, les **étudiants** suivent des **cours**.
- Les noms des entités sont toujours au singulier.
- Les associations sont les **liens** entre les concepts. Exemple : les **personnes** possèdent des **véhicules**.
- Une association décrit en général une action ou un fait.

# Identification des entités et des associations

- Les entités sont les concepts (c'est-à-dire les types d'objets manipulés). Exemples : les **personnes** possèdent des **véhicules**, les **étudiants** suivent des **cours**.
- Les noms des entités sont toujours au singulier.
- Les associations sont les **liens** entre les concepts. Exemple : les **personnes** possèdent des **véhicules**.
- Une association décrit en général une action ou un fait.
- Le dictionnaire de données permet de vérifier que toutes les informations sont contenues.

# Identification des entités et attributs

À partir de l'existant, on identifie les entités et leurs attributs.

- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.
- Les **livres** ont tous un *numéro ISBN*, un *titre*, un *éditeur* et une *année de publication*.
- Les **auteurs** qui écrivent les **livres** sont identifiés par un *numéro d'auteur*, et on stocke leur *nom* et *prénom*.
- Lorsqu'un **abonné** réalise un **emprunt** d'un **livre**, on enregistre le *numéro* et la *date de l'emprunt*.
- Lorsqu'il le restitue, on mémorise la *date de retour*.



# Représentation des entités

Abonné
<u>NumAbo</u>
NomAbo
PrénomAbo
AdrAbo
DateAbo

Livre
<u>ISBN</u>
Titre
NbrPages
Édition

Auteur
<u>NumAut</u>
NomAut
PrénomAut

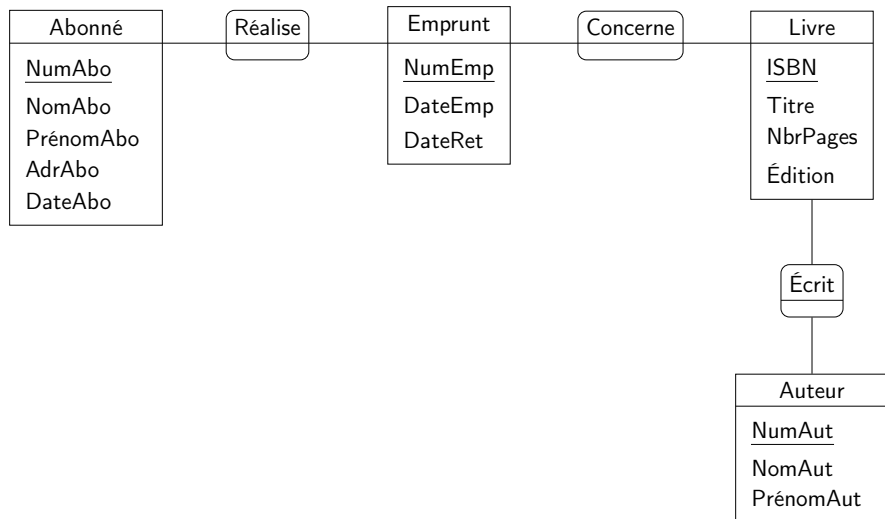
Emprunt
<u>NumEmp</u>
DateEmp
DateRet

Chaque information ne doit apparaître qu'**une et une seule fois** dans le schéma.

# Identification des associations

- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.
- Les **livres** ont tous un *numéro ISBN*, un *titre*, un *éditeur* et une *année de publication*.
- Les **auteurs** qui écrivent les **livres** sont identifiés par un *numéro d'auteur*, et on stocke leur *nom* et *prénom*.
- Lorsqu'un **abonné** réalise un **emprunt** d'un livre, on enregistre le *numéro* et la *date de l'emprunt*.
- Lorsqu'il le restitue, on mémorise la *date de retour*.

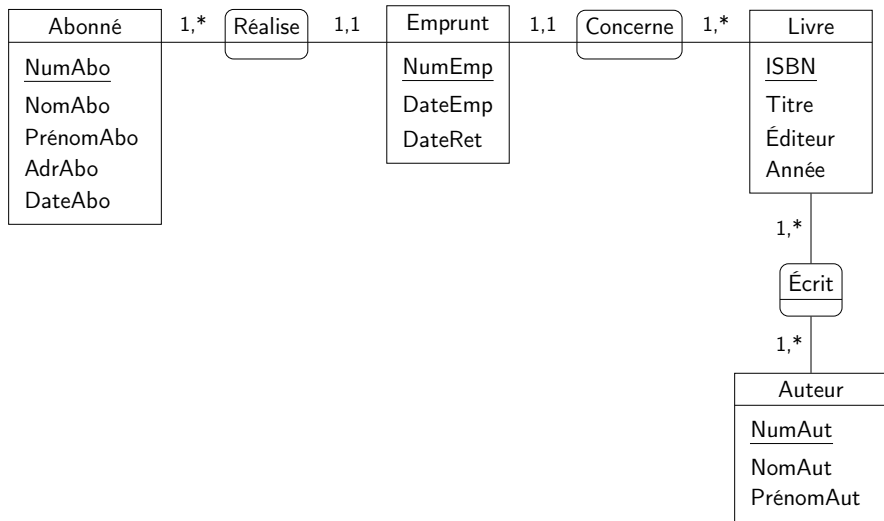
# Représentation des associations



On vérifie ensuite les attributs.

	Libellé	Type	Description
<i>Abonné</i>	NumAbo	entier	Numéro de l'abonné
	NomAbo	car(20)	Nom de l'abonné
	PrénomAbo	car(20)	Prénom de l'abonné
	AdrAbo	car(80)	Adresse de l'abonné
	DateAbo	date	Date de l'abonnement (AAA-MM-JJ)
<i>Auteur</i>	NumAut	entier	Numéro de l'auteur
	NomAut	car(20)	Nom de l'auteur
	PrénomAut	car(20)	Prénom de l'auteur
<i>Livre</i>	ISBN	car(13)	Code ISBN identifiant un livre
	Titre	car(80)	Titre du livre
	Editeur	car(20)	Nom de l'editeur
	Année	entier	Année de publication
<i>Emprunt</i>	NumEmp	entier	Numéro d'emprunt
	DateEmp	date	Date de l'emprunt d'un livre par un abonné
	DateRet	date	Date de retour d'un livre emprunté par un abonné

Enfin, on précise les cardinalités.

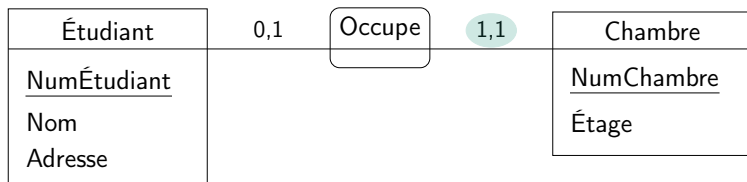


- On suppose qu'on a obtenu une représentation conceptuelle (schéma entités-associations) de notre BD.
- But : obtenir un schéma relationnel.

# Relations

- Les entités et certaines associations constituent les relations (c'est-à-dire les tables qui seront implantées dans la base de données).
- Notations :
  - Les attributs clés primaires sont soulignés par un trait plein.
  - Les attributs clés externes (ou étrangères) sont soulignés par un trait discontinu.
- Exemple :  
PRODUIT(NumProduit, Designation, PrixUnitaire)  
COMMANDE(NumCmd, Date, NumProduit)

# Associations « un à un »



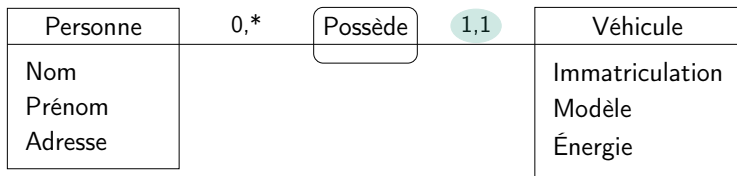
On crée une relation pour chaque entité. La clé primaire du côté 0,1 de l'association est clé externe dans l'entité du côté 1,1.

ÉTUDIANT(NumÉtudiant, Nom, Adresse)

CHAMBRE(NumChambre, Étage, NumÉtudiant)



# Associations « un à plusieurs »

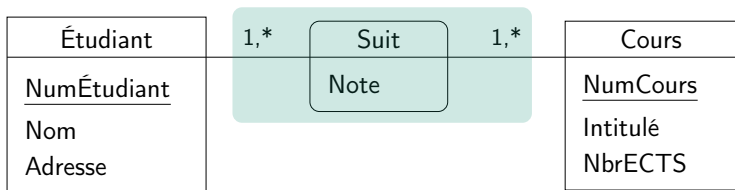


La clé primaire de l'entité du côté « plusieurs » (0,\* ici) de l'association est clé externe dans l'entité du côté « un » (1,1 ici)

PERSONNE(Numéro, Nom, Prénom, Adresse)

VÉHICULE(Immatriculation, Modèle, Énergie, Numéro)

## Associations « plusieurs à plusieurs »



On crée une relation pour l'association dont la clé primaire composée est la concaténation des clés primaires des entités.

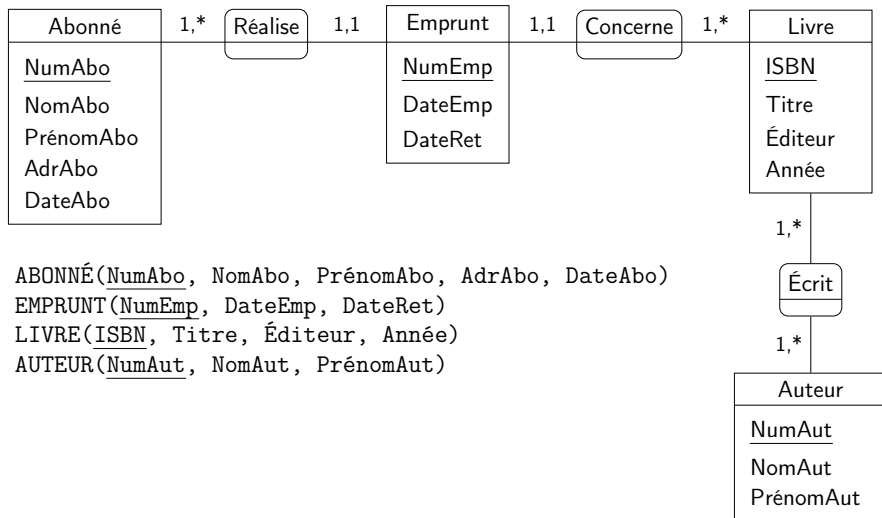
ÉTUDIANT(NumÉtudiant, Nom, Adresse)

COURS(NumCours, Intitulé, NbrECTS)

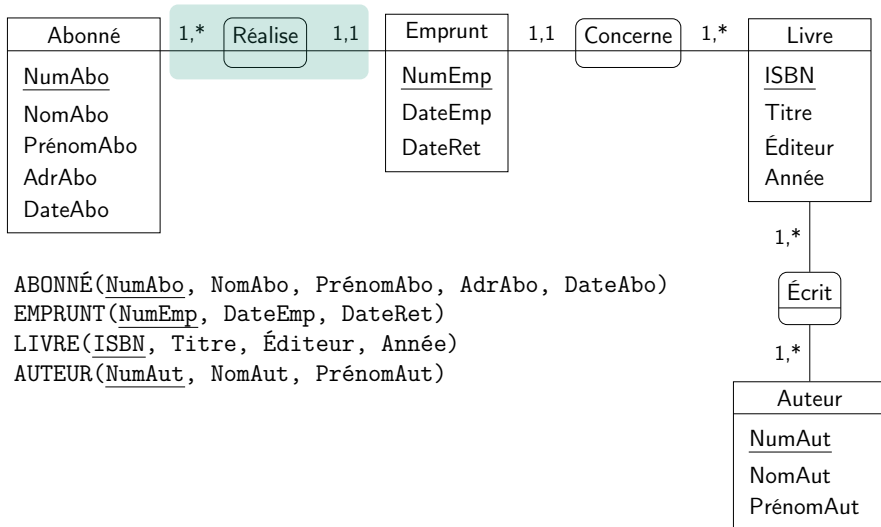
SUIT(NumÉtudiant, NumCours, Note)

La clé primaire de la relation SUIT est composée de deux attributs. Chacun de ces attributs étant aussi clé étrangère, on les souligne avec un double trait.

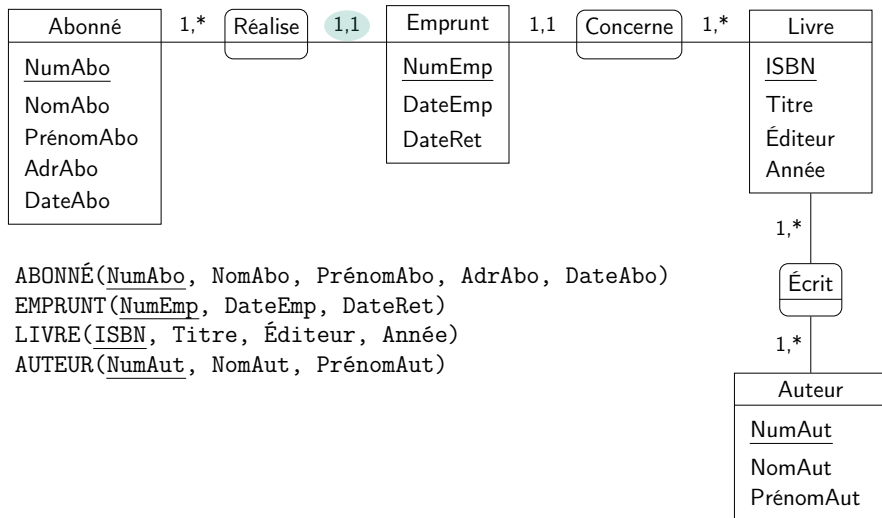
# Exemple : bibliothèque



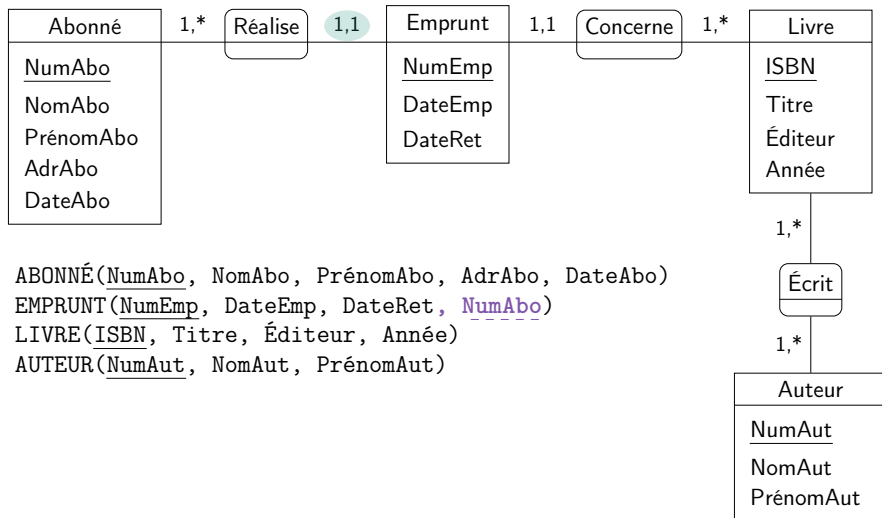
# Exemple : bibliothèque



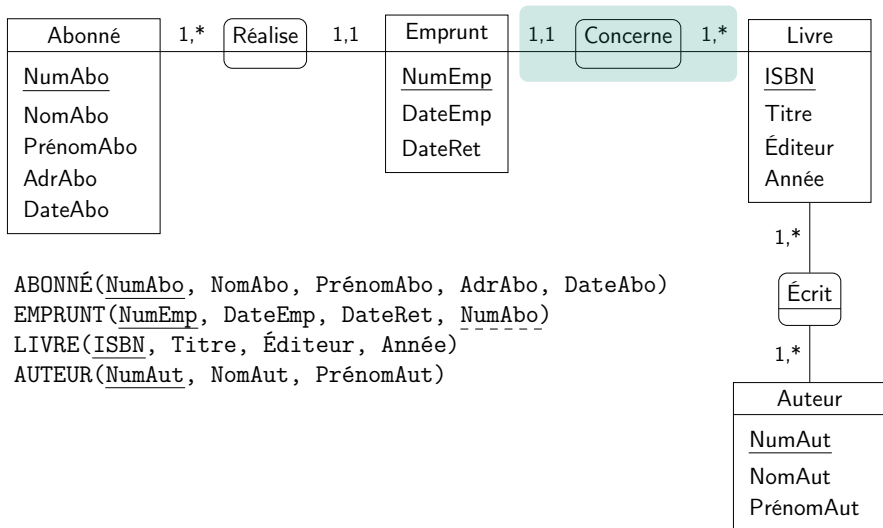
# Exemple : bibliothèque



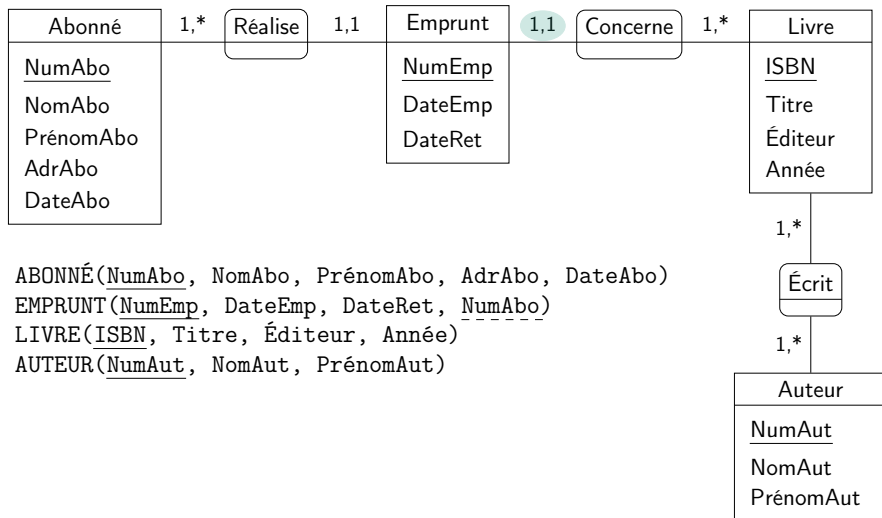
# Exemple : bibliothèque



# Exemple : bibliothèque

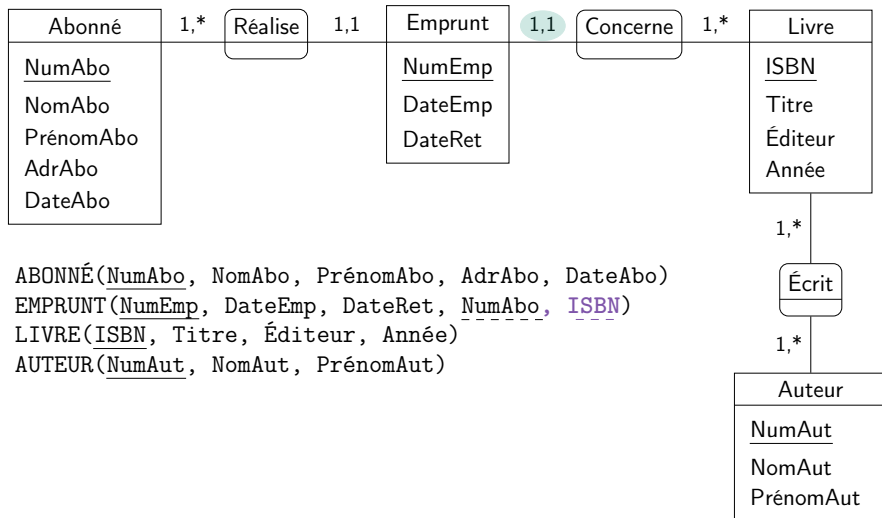


# Exemple : bibliothèque

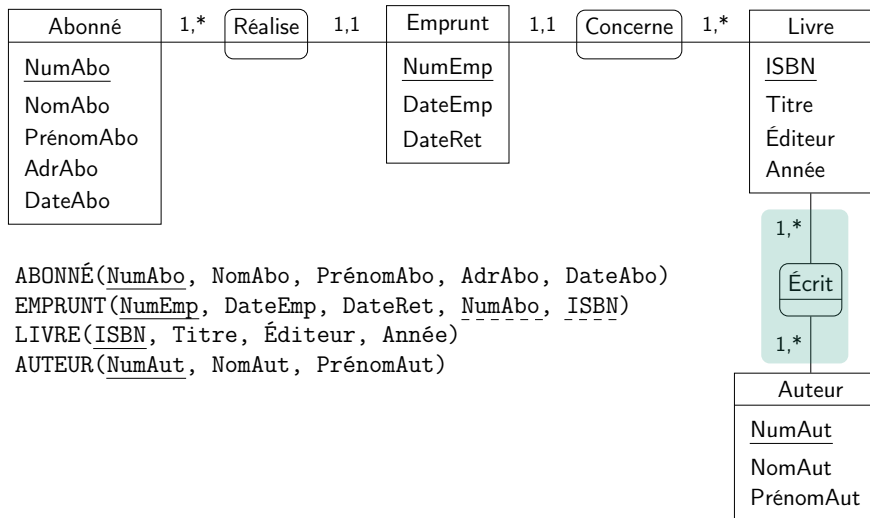




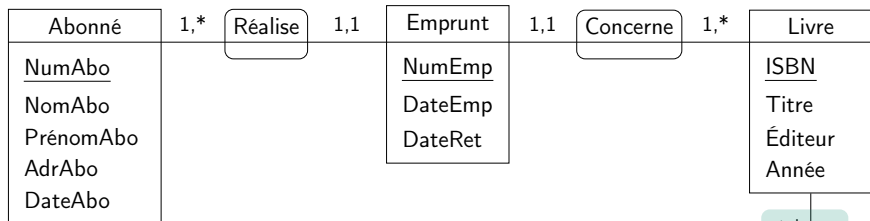
## Exemple : bibliothèque



# Exemple : bibliothèque



# Exemple : bibliothèque



ABONNÉ(NumAbo, NomAbo, PrénomAbo, AdrAbo, DateAbo)

EMPRUNT(NumEmp, DateEmp, DateRet, NumAbo, ISBN)

LIVRE(ISBN, Titre, Éditeur, Année)

AUTEUR(NumAut, NomAut, PrénomAut)

ÉCRIT(ISBN, NumAut)

# Résumé : Passage du modèle Entité-Association au modèle relationnel

- ➊ À chaque entité correspond une relation.

## Résumé : Passage du modèle Entité-Association au modèle relationnel

- ➊ À chaque entité correspond une relation.
- ➋ Si dans une association  $A$ , il existe une entité  $E$  pour laquelle la cardinalité de  $(E, A)$  est égale à 1, ajouter comme clé étrangère, dans la relation qui traduit  $E$ , la clé primaire des autres entités participant à  $A$ .

# Résumé : Passage du modèle Entité-Association au modèle relationnel

- ➊ À chaque entité correspond une relation.
- ➋ Si dans une association  $A$ , il existe une entité  $E$  pour laquelle la cardinalité de  $(E, A)$  est égale à 1, ajouter comme clé étrangère, dans la relation qui traduit  $E$ , la clé primaire des autres entités participant à  $A$ .
- ➌ Si dans une association  $A$ , il n'existe pas d'entité  $E$  pour laquelle la cardinalité de  $(E, A)$  est égale à 1, créer une nouvelle relation contenant la clé primaire de chaque entité participant à l'association. Ces attributs feront partie de la clé primaire de la nouvelle relation.

- 1 Rappels : modèle relationnel
- 2 Méthode de conception
- 3 SQL
  - Présentation
  - Requêtes d'interrogation de données
    - Requêtes simples
    - Comparaisons

# Objectifs de SQL

- Créer la structure de la base de données et de ses tables
- Exécuter les tâches de base de la gestion des données, telle que l'insertion, la modification et la suppression de données des tables
- Effectuer des requêtes simples ou complexes



# Évolutions de la norme SQL

- Norme SQL1 (ANSI X3.135-1986 puis 1989)
  - Bases du langage SQL : verbes, clauses, opérateurs et syntaxe
  - Implantation des fonctions à la discrétion des éditeurs de SGBD
  - Intégrité référentielle (clés primaires et étrangères)

# Évolutions de la norme SQL

- Norme SQL1 (ANSI X3.135-1986 puis 1989)
  - Bases du langage SQL : verbes, clauses, opérateurs et syntaxe
  - Implantation des fonctions à la discrétion des éditeurs de SGBD
  - Intégrité référentielle (clés primaires et étrangères)
- Norme SQL2 (ANSI X3.135-1992)
  - Inclut des détails de mise en œuvre des fonctions
  - Norme étudiée dans ce cours
  - Différences mineures entre les implantations

# Évolutions de la norme SQL

- Norme SQL1 (ANSI X3.135-1986 puis 1989)
  - Bases du langage SQL : verbes, clauses, opérateurs et syntaxe
  - Implantation des fonctions à la discrétion des éditeurs de SGBD
  - Intégrité référentielle (clés primaires et étrangères)
- Norme SQL2 (ANSI X3.135-1992)
  - Inclut des détails de mise en œuvre des fonctions
  - Norme étudiée dans ce cours
  - Différences mineures entre les implantations
- Norme SQL3 (ANSI X3.135-1999)
  - Propose aussi des extensions objet.

# Langage de requêtes SQL

- SQL : Structured Query Language
- Langage structuré de

# Langage de requêtes SQL

- SQL : Structured Query Language
- Langage structuré de
  - Définition de données :
    - Créer des tables, contraintes, etc.
    - Contrôler l'accès aux données (définition des droits)

# Langage de requêtes SQL

- SQL : Structured Query Language
- Langage structuré de
  - Définition de données :
    - Créer des tables, contraintes, etc.
    - Contrôler l'accès aux données (définition des droits)
  - Manipulation de données :
    - Mettre à jour les données (ajout, suppression, modification de n-uplets)

# Langage de requêtes SQL

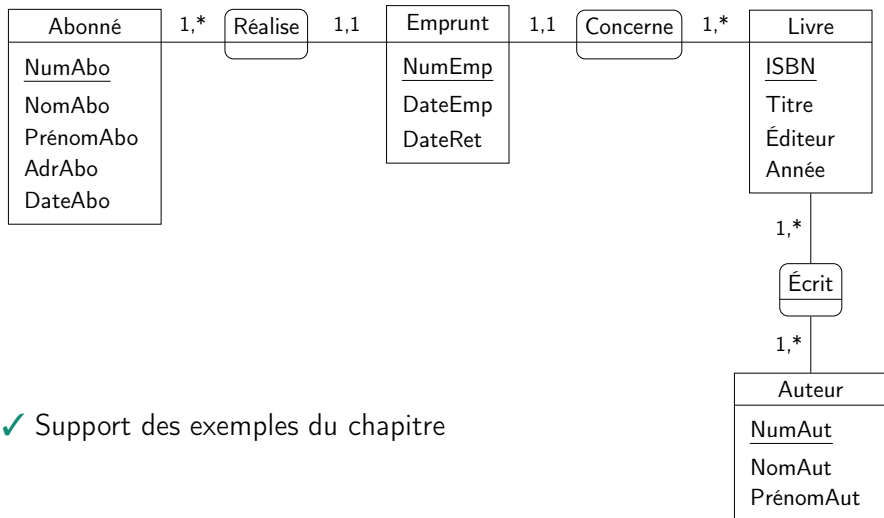
- SQL : Structured Query Language
- Langage structuré de
  - Définition de données :
    - Créer des tables, contraintes, etc.
    - Contrôler l'accès aux données (définition des droits)
  - Manipulation de données :
    - Mettre à jour les données (ajout, suppression, modification de n-uplets)
  - Interrogation des données :
    - Recherches sélectives efficaces

# Langage de requêtes SQL

- SQL : Structured Query Language
- Langage structuré de
  - Définition de données :
    - Créer des tables, contraintes, etc.
    - Contrôler l'accès aux données (définition des droits)
  - Manipulation de données :
    - Mettre à jour les données (ajout, suppression, modification de n-uplets)
  - Interrogation des données :
    - Recherches sélectives efficaces
- Langage standard utilisé par tous les SGBD relationnels
  - PostgreSQL, MySQL, SQLite, Oracle, DB2, SQL Server, Access, ...



## Exemple BD Bibliothèque : schéma E-A



✓ Support des exemples du chapitre

# Exemple BD Bibliothèque : schéma relationnel

- Schéma relationnel de la BD

ABONNÉ(NumAbo, NomAbo, PrénomAbo, AdrAbo, DateAbo)

LIVRE(ISBN, Titre, Éditeur, Année)

AUTEUR(NumAut, NomAut, PrénomAut)

ÉCRIT(ISBN, NumAut)

EMPRUNT(NumEmp, NumAbo, ISBN, DateEmp, DateRet)

# Exemple BD Bibliothèque : tables

- Tables (relations).

LIVRE

ISBN	Titre	Éditeur	Année
9782212112818	Bases de Données	Eyrolles	1989
9782225805158	Le Langage C	Masson	1985
9782207257357	Fondation	Denoël	2006

AUTEUR

NumAut	NomAut	PrénomAut
1	Gardarin	Georges
2	Kernighan	Brian
3	Ritchie	Dennis
4	Asimov	Isaac

ÉCRIT

ISBN	NumAut
9782212112818	1
9782225805158	2
9782225805158	3
9782207257357	4

ABONNE

NumAbo	NomAbo	PrénomAbo	DateAbo
1	Dupont	Philippe	2008-06-18
2	Durand	Arthur	2009-01-02
3	Dupont	Charlie	2015-05-03
4	Ducros	Marie	2020-07-04
5	Vernier	Alain	2021-09-15

EMPRUNT

NumEmp	ISBN	NumAbo	DateEmp	DateRet
1	9782225805158	2	2021-09-06	2021-09-20
2	9782225805158	3	2021-09-25	2021-10-11
3	9782212112818	1	2021-10-28	2021-11-10
4	9782212112818	1	2021-11-08	NULL

# Conventions de notations

- Mots-clés de SQL : caractères COURIER majuscules
- Paramètres des requêtes : caractères courier minuscules

# Conventions de notations

- Mots-clés de SQL : caractères COURIER majuscules
- Paramètres des requêtes : caractères courier minuscules
- Paramètres optionnels : [option]
- Valeurs multiples possibles : valeur<sub>1</sub> | valeur<sub>2</sub>
- Options multiples : [option<sub>1</sub> | option<sub>2</sub>]

## Format général des requêtes :

- SELECT spécifie les colonnes qui doivent apparaître dans les résultats
- FROM spécifie la table ou les tables à utiliser
- WHERE filtre les lignes selon une condition donnée
- GROUP BY forme des groupes de lignes de même valeur de colonne
- HAVING filtre les groupes sujets à une certaine condition
- ORDER BY spécifie l'ordre d'apparition des données dans le résultat

# Langage d'interrogation de données

## Requête simple de sélection

- Retrouver les attributs des  $n$ -uplets (lignes/tuples) vérifiant certains critères.

```
SELECT attribut1, ..., attributi
FROM table1, ..., tablej
WHERE condition1
[AND|OR] ...
[AND|OR] conditionk;
```

Résultat de la requête

*$n$ -uplets  
vérifiant  
les  
conditions*

Attribut <sub>1</sub>	...	Attribut <sub>i</sub>
p	...	x
q	...	y
⋮	⋮	⋮
r	...	z

Table <sub>1</sub>	...	Table <sub>j</sub>
...		...



# Requête simple de sélection

- Exemple :

```
SELECT NomAbo, PrénomAbo, DateAbo  
FROM Abonné;
```



# Requête simple de sélection

- Exemple :

```
SELECT NomAbo, PrénomAbo, DateAbo  
FROM Abonné;
```

- Sélection des noms, prénoms et dates d'abonnement de tous les abonnés.

NomAbo	PrénomAbo	DateAbo
Dupont	Philippe	2008-06-18
Durand	Arthur	2009-01-02
Dupont	Charlie	2015-05-03
Ducros	Marie	2020-07-04
Vernier	Alain	2021-09-15

## Sélection de tous les attributs

- Le symbole \* est synonyme de « tous les attributs » :

```
SELECT *  
FROM Abonné;
```

# Sélection de tous les attributs

- Le symbole \* est synonyme de « tous les attributs » :

```
SELECT *  
FROM Abonné;
```

- Sélection de tous les attributs pour tous les abonnés.

NumAbo	NomAbo	PrénomAbo	DateAbo
1	Dupont	Philippe	2008-06-18
2	Durand	Arthur	2009-01-02
3	Dupont	Charlie	2015-05-03
4	Ducros	Marie	2020-07-04
5	Vernier	Alain	2021-09-15

# La clause WHERE

- La clause WHERE permet de définir des critères de sélection des n-uplets :

```
SELECT *  
FROM Abonné  
WHERE NumAbo > 2;
```

# La clause WHERE

- La clause WHERE permet de définir des critères de sélection des n-uplets :

```
SELECT *  
FROM Abonné  
WHERE NumAbo > 2;
```

- Sélection de tous les attributs pour tous les abonnés dont le numéro est supérieur (strictement) à 2.

NumAbo	NomAbo	PrénomAbo	DateAbo
3	Dupont	Charlie	2015-05-03
4	Ducros	Marie	2020-07-04
5	Vernier	Alain	2021-09-15

## Requêtes multi-critères

- Combinaison de critères de sélection des n-uplets :

```
SELECT *  
FROM Abonné  
WHERE NumAbo >= 2 AND NumAbo <=4;
```

# Requêtes multi-critères

- Combinaison de critères de sélection des n-uplets :

```
SELECT *  
FROM Abonné  
WHERE NumAbo >= 2 AND NumAbo <=4;
```

- Sélection de tous les attributs pour les abonnés dont le numéro est compris entre 2 et 4.

NumAbo	NomAbo	PrénomAbo	DateAbo
2	Durand	Arthur	2009-01-02
3	Dupont	Charlie	2015-05-03
4	Ducros	Marie	2020-07-04

## Requêtes multi-critères (2)

- Combinaison de critères de sélection des n-uplets :

```
SELECT *  
FROM Abonné  
WHERE NumAbo <= 2 OR NumAbo > 4;
```



## Requêtes multi-critères (2)

- Combinaison de critères de sélection des n-uplets :

```
SELECT *  
FROM Abonné  
WHERE NumAbo <= 2 OR NumAbo > 4;
```

- Sélection de tous les attributs pour les abonnés dont le numéro est inférieur ou égal à 2 ou supérieur strictement à 4.

NumAbo	NomAbo	PrénomAbo	DateAbo
1	Dupont	Philippe	2008-06-18
2	Durand	Arthur	2009-01-02
5	Vernier	Alain	2021-09-15

## Disjonctions et conjonctions de critères

- Si la requête mélange conjonctions (AND) et disjonctions (OR) de critères, il faut faire attention à la priorité des opérateurs.

## Disjonctions et conjonctions de critères

- Si la requête mélange conjonctions (AND) et disjonctions (OR) de critères, il faut faire attention à la priorité des opérateurs.
- Exemple :

```
SELECT *  
FROM Emprunt  
WHERE NumAbo = 1 AND DateEmp = '2021-10-28'  
OR DateEmp = '2021-09-06';
```

est équivalent à

```
SELECT *  
FROM Emprunt  
WHERE (NumAbo = 1 AND DateEmp = '2021-10-28')  
OR DateEmp = '2021-09-06';
```

Résultat : liste des emprunts de l'abonné n°1 le 28 octobre 2021 et de **tous** les emprunts du 6 septembre 2021.

## Disjonctions et conjonctions de critères

- Si la requête mélange conjonctions (AND) et disjonctions (OR) de critères, il faut faire attention à la priorité des opérateurs.
- Exemple :

```
SELECT *  
FROM Emprunt  
WHERE NumAbo = 1 AND DateEmp = '2021-10-28'  
OR DateEmp = '2021-09-06';
```

est équivalent à

```
SELECT *  
FROM Emprunt  
WHERE (NumAbo = 1 AND DateEmp = '2021-10-28')  
OR DateEmp = '2021-09-06';
```

Résultat : liste des emprunts de l'abonné n°1 le 28 octobre 2021 et de **tous** les emprunts du 6 septembre 2021.

- **AND est prioritaire sur OR.**

# Opérateurs de comparaison

- Opérateurs de comparaison :
  - =, <>, <, >, >=, <= : comparaisons classiques

# Opérateurs de comparaison

- Opérateurs de comparaison :
  - =, <>, <, >, >=, <= : comparaisons classiques
  - BETWEEN val<sub>1</sub> AND val<sub>2</sub> : valeurs comprises entre val<sub>1</sub> et val<sub>2</sub>

# Opérateurs de comparaison

- Opérateurs de comparaison :
  - =, <>, <, >, >=, <= : comparaisons classiques
  - BETWEEN val<sub>1</sub> AND val<sub>2</sub> : valeurs comprises entre val<sub>1</sub> et val<sub>2</sub>
  - IS NULL : valeur NULL
  - IS NOT NULL : valeurs non NULL

# Opérateurs de comparaison

- Opérateurs de comparaison :
  - =, <>, <, >, >=, <= : comparaisons classiques
  - BETWEEN val<sub>1</sub> AND val<sub>2</sub> : valeurs comprises entre val<sub>1</sub> et val<sub>2</sub>
  - IS NULL : valeur NULL
  - IS NOT NULL : valeurs non NULL
  - IN (listeValeurs) : valeurs appartenant à la liste
  - NOT IN (listeValeurs) : valeurs n'appartenant pas à la liste



# Opérateurs de comparaison

- Opérateurs de comparaison :
  - =, <>, <, >, >=, <= : comparaisons classiques
  - BETWEEN val<sub>1</sub> AND val<sub>2</sub> : valeurs comprises entre val<sub>1</sub> et val<sub>2</sub>
  - IS NULL : valeur NULL
  - IS NOT NULL : valeurs non NULL
  - IN (listeValeurs) : valeurs appartenant à la liste
  - NOT IN (listeValeurs) : valeurs n'appartenant pas à la liste
- Exemple :

```
SELECT *  
FROM Abonnés  
WHERE NumAbo BETWEEN 2 AND 4;
```

## Sélection sur les chaînes de caractères

- Sélection des n-uplets contenant une chaîne

```
SELECT *  
FROM Abonnés  
WHERE NomAbo = 'Durand';
```

# Sélection sur les chaînes de caractères

- Sélection des n-uplets contenant une chaîne

```
SELECT *  
FROM Abonnés  
WHERE NomAbo = 'Durand';
```

- Sélection des abonnés dont le nom est Durand.

NumAbo	NomAbo	PrénomAbo	DateAbo
2	Durand	Arthur	2009-01-02

## Sélection de sous-chaînes de caractères

- Motifs de sélection des chaînes

`attribut [NOT] LIKE expr`

## Sélection de sous-chaînes de caractères

- Motifs de sélection des chaînes

`attribut [NOT] LIKE expr`

- L'attribut `attribut` doit / ne doit pas correspondre à l'expression `expr`

# Sélection de sous-chaînes de caractères

- Motifs de sélection des chaînes

`attribut [NOT] LIKE expr`

- L'attribut `attribut` doit / ne doit pas correspondre à l'expression `expr`
- Jokers utilisables dans `expr`
  - Symbole `%` : n'importe quelle chaîne de caractères, vide y compris
  - Symbole `_` : n'importe quel caractère (un seul, pas vide).

## Sélection de sous-chaînes de caractères

- Motifs de sélection des chaînes

`attribut [NOT] LIKE expr`

- L'attribut `attribut` doit / ne doit pas correspondre à l'expression `expr`
- Jokers utilisables dans `expr`
  - Symbole `%` : n'importe quelle chaîne de caractères, vide y compris
  - Symbole `_` : n'importe quel caractère (un seul, pas vide).
- Exemple :

```
SELECT *  
FROM Abonné  
WHERE NomAbo LIKE 'Dupon_';
```

# Sélection de sous-chaînes de caractères

- Motifs de sélection des chaînes

`attribut [NOT] LIKE expr`

- L'attribut `attribut` doit / ne doit pas correspondre à l'expression `expr`
- Jokers utilisables dans `expr`
  - Symbole `%` : n'importe quelle chaîne de caractères, vide y compris
  - Symbole `_` : n'importe quel caractère (un seul, pas vide).

- Exemple :

```
SELECT *  
FROM Abonné  
WHERE NomAbo LIKE 'Dupon_';
```

- Sélection de tous les attributs pour les abonnés dont le nom est Dupond, Dupont, ... (mais pas Dupon).



## Sélection sur les chaînes de caractères

- Exemple :

```
SELECT *  
FROM Abonné  
WHERE NomAbo LIKE 'Du%n_';
```

# Sélection sur les chaînes de caractères

- Exemple :

```
SELECT *  
FROM Abonné  
WHERE NomAbo LIKE 'Du%n_';
```

- Sélection des abonnés dont le nom est Dupont, Dupond, Durant, Durand, Dune, etc.

NumAbo	NomAbo	PrénomAbo	DateAbo
1	Dupont	Philippe	2008-06-18
2	Durand	Arthur	2009-01-02
3	Dupont	Charlie	2015-05-03

## Sélections sur les dates

- Dates entourées de ' '
- Valeurs séparées par -

```
SELECT *  
FROM Abonné  
WHERE DateAbo > '2007-10-01';
```

## Sélections sur les dates

- Dates entourées de ' '
- Valeurs séparées par -

```
SELECT *  
FROM Abonné  
WHERE DateAbo > '2007-10-01';
```

- Sélection des abonnés inscrits après le 1<sup>er</sup> octobre 2007.

NumAbo	NomAbo	PrénomAbo	DateAbo
1	Dupont	Philippe	2008-06-18
2	Durand	Arthur	2009-01-02
3	Dupont	Charlie	2015-05-03
4	Ducros	Marie	2020-07-04
5	Vernier	Alain	2021-09-15

# Intervalles de dates et de chaînes de caractères

## Opérateur BETWEEN

- Applicable aux mêmes types que les opérateurs =, <, >
- Sélection des abonnés inscrits entre le 1<sup>er</sup> et le 31 octobre inclus :

```
SELECT *  
FROM Abonné  
WHERE DateAbo BETWEEN '2007-10-01' AND '2007-10-31';
```

# Intervalles de dates et de chaînes de caractères

## Opérateur BETWEEN

- Applicable aux mêmes types que les opérateurs =, <, >
- Sélection des abonnés inscrits entre le 1<sup>er</sup> et le 31 octobre inclus :

```
SELECT *  
FROM Abonné  
WHERE DateAbo BETWEEN '2007-10-01' AND '2007-10-31';
```

- Sélection des abonnés de nom supérieur à « Dupond » et inférieur à « Ferrand » selon l'ordre lexicographique (alphabétique) :

```
SELECT *  
FROM Abonné  
WHERE NomAbo BETWEEN 'Dupond' AND 'Ferrand';
```

# Valeurs NULL

- Opérateur IS NULL
  - Vrai si la valeur est NULL (absente, inappropriée, valeur par défaut)
  - Liste des emprunts en cours

```
SELECT *  
FROM Emprunt  
WHERE DateRet IS NULL;
```

# Valeurs NULL

- Opérateur IS NULL

- Vrai si la valeur est NULL (absente, inappropriée, valeur par défaut)
- Liste des emprunts en cours

```
SELECT *  
FROM Emprunt  
WHERE DateRet IS NULL;
```

- Opérateur IS NOT NULL

- Vrai si la valeur est différente de NULL
- Liste des emprunts terminés (livres restitués)

```
SELECT *  
FROM Emprunt  
WHERE DateRet IS NOT NULL;
```



# Sélection dans une liste de valeurs

- Opérateur IN
  - Sélectionne les valeurs dans la liste
  - Sélection des livres des éditeurs Denoël et Eyrolles :

```
SELECT *  
FROM Livre  
WHERE Editeur IN 'Denoël', 'Eyrolles';
```

# Sélection dans une liste de valeurs

- Opérateur IN

- Sélectionne les valeurs dans la liste
- Sélection des livres des éditeurs Denoël et Eyrolles :

```
SELECT *  
FROM Livre  
WHERE Editeur IN 'Denoël', 'Eyrolles';
```

- Opérateur NOT IN

- Sélectionne les valeurs qui ne sont pas dans la liste
- Sélection des livres édités par d'autres éditeurs que Masson et Springer

```
SELECT *  
FROM Livre  
WHERE Editeur NOT IN 'Masson', 'Springer';
```

## Ordre des n-uplets

- Ordonner la sortie selon les attributs spécifiés

`ORDER BY att1, ..., atti [ASC|DESC]`

- Affiche la sortie en triant les n-uplets selon les attributs  $att_1, \dots, att_i$  dans l'ordre croissant (ASC), par défaut, ou décroissant (DESC)

## Ordre des n-uplets

- Ordonner la sortie selon les attributs spécifiés

`ORDER BY att1, ..., atti [ASC|DESC]`

- Affiche la sortie en triant les n-uplets selon les attributs `att1, ..., atti` dans l'ordre croissant (ASC), par défaut, ou décroissant (DESC)
- Exemple :

```
SELECT * FROM Abonné  
WHERE DateAbo > '2007-10-01'  
ORDER BY NomAbo, PrénomAbo;
```

NumAbo	NomAbo	PrénomAbo	DateAbo
4	Ducros	Marie	2020-07-04
3	Dupont	Charlie	2015-05-03
1	Dupont	Philippe	2008-06-18
2	Durand	Arthur	2009-01-02
5	Vernier	Alain	2021-09-15

# Renommage de colonnes et valeurs distinctes

- Renommage des colonnes du résultat :

```
SELECT NomAbo AS "Nom de l'abonné",  
DateAbo AS Date  
FROM Abonné;
```

## Renommage de colonnes et valeurs distinctes

- Renommage des colonnes du résultat :

```
SELECT NomAbo AS "Nom de l'abonné",  
DateAbo AS Date  
FROM Abonné;
```

- Notez l'utilisation des guillemets (") car nouvel identifiant (nouveau nom d'attribut) et pas simplement chaîne de caractères. Beaucoup de SGBD se servent également de l'accent grave (') à la place des guillemets.

# Renommage de colonnes et valeurs distinctes

- Renommage des colonnes du résultat :

```
SELECT NomAbo AS "Nom de l'abonné",  
DateAbo AS Date  
FROM Abonné;
```

- Notez l'utilisation des guillemets (") car nouvel identifiant (nouveau nom d'attribut) et pas simplement chaîne de caractères. Beaucoup de SGBD se servent également de l'accent grave (') à la place des guillemets.

- Sélection de valeurs distinctes

```
SELECT DISTINCT NomAbo  
FROM Abonné  
WHERE NumAbo <= 4;
```

NomAbo
Dupont
Durand
Dupont



NomAbo
Dupont
Durand

## Valeurs calculées

- La requête peut renvoyer des valeurs calculées.



# Valeurs calculées

- La requête peut renvoyer des valeurs calculées.
- Exemple :

```
SELECT *, DateRet - DateEmp  
FROM Emprunt;
```

Valeur NULL

NumEmp	ISBN	NumAbo	DateEmp	DateRet	DateRet-DateEmp
1	9782225805158	2	2021-09-06	2021-09-20	14
2	9782225805158	3	2021-09-25	2021-10-11	16
3	9782212112818	1	2021-10-28	2021-11-10	13
4	9782212112818	1	2021-11-08	NULL	NULL

## À suivre

