

# Bases de données

## Cours 4

### Requêtes SQL (suite)

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)

22 novembre 2022

- 1 Notations et exemple fil-rouge
  - Exemple : bibliothèque
  - Notations
- 2 Langage de manipulation des données
- 3 Agrégation et regroupement
- 4 SQL : Jointures avancées

# Exemple fil-rouge : bibliothèque

## Attributs des entités

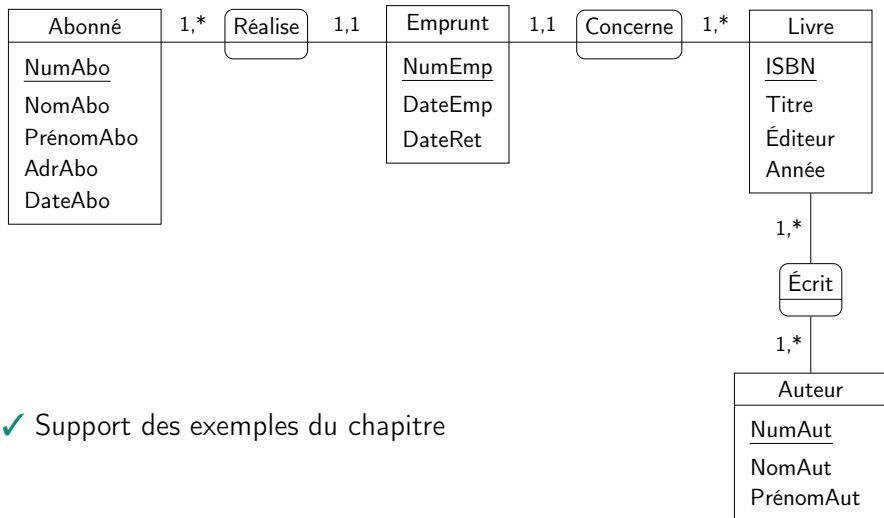
- Chaque **abonné** a un *numéro d'abonné* unique, un *nom*, un *prénom*, une *adresse* et une *date d'abonnement*.
- Les **livres** ont tous un *numéro ISBN*, un *titre*, un *éditeur* et une *année de publication*.
- Les **auteurs** qui écrivent les **livres** sont identifiés par un *numéro d'auteur*,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

## 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
- Paramètres optionnels : [option]
- Valeurs multiples possibles : valeur<sub>1</sub> | valeur<sub>2</sub>
- Options multiples : [option<sub>1</sub> | option<sub>2</sub>]



- 1 Notations et exemple fil-rouge
- 2 Langage de manipulation des données
  - Insertions de n-uplets
  - Suppression de n-uplets
  - Modification de n-uplets
  - Exercice
- 3 Agrégation et regroupement
- 4 SQL : Jointures avancées

# Insertion de n-uplets

- Insertion de n-uplets

```
INSERT INTO TABLE [(attribut1, ..., attributi)]  
VALUES (val11, ..., vali1), (val12, ..., vali2), ... ;
```

- Insertion des n-uplets de valeurs (val<sub>1</sub><sup>1</sup>, ..., val<sub>i</sub><sup>1</sup>), ... dans la table.

## Exemple

```
INSERT INTO Auteur VALUES  
(5, 'Berge', 'Claude');
```

```
INSERT INTO Abonné VALUES  
(7, 'Dunod', 'Charles', '2021-11-28'),  
(8, 'Dupond', 'Hector', '2021-11-30');
```

## Insertion de n-uplets(2)

- Il est possible de ne pas spécifier certaines valeurs :

```
INSERT INTO Emprunt(NumEmp, ISBN, NumAbo, DateEmp) VALUES  
(5, '9782207257357', 5, '2021-11-29');
```

- Les valeurs non renseignées prennent
  - la valeur par défaut pour l'attribut si elle a été définie (contrainte DEFAULT valeur de la création de la table);
  - la valeur NULL sinon.
- Les n-uplets insérés doivent vérifier les contraintes de la table (domaine, NOT NULL, UNIQUE et autres).

# Suppression de n-uplets

- Suppression de n-uplets vérifiant les conditions :

```
DELETE FROM table  
WHERE condition1  
[AND|OR] ...  
[AND|OR] conditioni;
```

## Exemple

- Suppression de tous les emprunts

```
DELETE FROM Emprunt;
```

- Suppression de tous les emprunts précédant le 1<sup>er</sup> janvier 2007

```
DELETE FROM Emprunt  
WHERE DateEmp < '2007-01-01';
```

# Suppression de n-uplets référencés

- Attribut référencé par un autre attribut
  - Suppression impossible si un autre n-uplet référence la valeur.

```
CREATE TABLE Livre (  
    ISBN VARCHAR(10) NOT NULL, ...);
```

```
CREATE TABLE Emprunt (  
    ...,  
    FOREIGN KEY (ISBN) REFERENCES Livre(ISBN));
```

## Exemple

```
DELETE FROM Livre WHERE ISBN = '9782207257357';
```

On ne peut pas supprimer de n-uplet de Livre s'il est référencé par un n-uplet d'Emprunt.

## Suppression de n-uplets référencés (2)

- Suppression en cascade : ON DELETE CASCADE
  - Suppression du n-uplet entraîne la suppression des n-uplets qui le référencent.

```
CREATE TABLE Livre (  
    ISBN VARCHAR(10) NOT NULL, ...);
```

```
CREATE TABLE Emprunt (  
    ...,  
    FOREIGN KEY (ISBN) REFERENCES Livre(ISBN)  
    ON DELETE CASCADE);
```

### Exemple

```
DELETE FROM Livre WHERE ISBN = '9782207257357';
```

La suppression de cet n-uplet de Livre entraînera la suppression des n-uplets d'Emprunt correspondants.

## Suppression de n-uplets référencés (3)

- Remplacement : ON DELETE SET NULL
  - NULL est substitué aux valeurs dont la référence est supprimée.

```
CREATE TABLE Emprunt (  
    . . . ,  
    FOREIGN KEY (NumAbo) REFERENCES Abonné(NumAbo)  
    ON DELETE SET NULL);
```

### Exemple

```
DELETE FROM Abonné WHERE NomAbo = 'Bertrand'  
AND PrénomAbo = 'Christine';
```

La suppression de cet n-uplet de Abonné entraînera la mise à la valeur NULL pour le numéro d'abonné correspondant dans la table Emprunt.

- Variante : ON DELETE SET DEFAULT : remplacement par la valeur par défaut (précisée à la création).

# Modification de n-uplets

- Applique la modification `modif` aux n-uplets vérifiant les conditions :

```
UPDATE table SET modif  
WHERE condition1  
[AND|OR] ...  
[AND|OR] conditioni;
```

## Exemple

- Changer le nom de Ferrand Christine en Bertrand :

```
UPDATE Abonné SET NomAbo = 'Bertrand'  
WHERE NomAbo = 'Ferrand'  
AND PrenomAbo = 'Christine';
```

- Ajoute 1000 au numéro des abonnés dont le prénom commence par C :

```
UPDATE Abonné SET NumAbo = NumAbo + 1000  
WHERE PrenomAbo LIKE 'C%';
```



# Modification de n-uplets référencés

- Attribut référencé par un autre attribut
  - Modification impossible si un autre n-uplet référence la valeur.

```
CREATE TABLE Écrit (  
    ...,  
    FOREIGN KEY (NumAut) REFERENCES Auteur(NumAut));
```

## Exemple

```
UPDATE Auteur SET NumAut = NumAut + 10000  
WHERE NomAut LIKE 'L%';
```

On ne peut pas modifier un NumAut d'Auteur s'il est référencé par un n-uplet d'Écrit.

## Modification de n-uplets référencés

- On peut contourner cette restriction en définissant le comportement en cas de modification :

```
CREATE TABLE Écrit (  
    ...,  
    FOREIGN KEY (NumAut) REFERENCES Auteur(NumAut))  
ON UPDATE CASCADE;
```

- En cas de modification de NumAut dans Auteur, les modifications sont répercutées dans Écrit
- En général, c'est une (très) mauvaise idée de changer la valeur d'une clé primaire, en particulier d'un numéro (à auto-incrémentation).
- L'option ON UPDATE CASCADE est plus utile quand une clé étrangère fait référence à une clé non primaire.

# Exercice

- Ajout de l'auteur Claude Berge
- Ajout du livre Graphes et Hypergraphes écrit par Claude Berge en 1970 aux éditions Dunod (ISBN : 9782040097554)
- Modification de l'année pour 1973 du livre Graphes et Hypergraphes écrit par Claude Berge
- Ajout de l'abonné Maire Pekkeau qui s'est abonné aujourd'hui
- Modification du prénom et du nom des abonnés dont le prénom est Maire et le nom Pekkeau pour Marie Pelleau
- Suppression des abonnés dont le prénom est Marie et le nom Pelleau

- 1 Notations et exemple fil-rouge
- 2 Langage de manipulation des données
- 3 Agrégation et regroupement**
  - Agrégation
  - Regroupement
  - Exercice
- 4 SQL : Jointures avancées

# Opérateurs d'agrégation

- Opérateurs d'agrégation

`SELECT FONCTION(att) FROM ...;`

- Fonction de calcul appliquée aux valeurs des n-uplets sélectionnés.

<code>SUM(att)</code>	somme des valeurs de l'attribut att
<code>AVG(att)</code>	moyenne
<code>MIN(att)</code>	valeur minimale
<code>MAX(att)</code>	valeur maximale
<code>COUNT(att)</code>	compte le nombre de valeurs (sauf NULL)
<code>COUNT(DISTINCT att)</code>	compte le nombre de valeurs distinctes
<code>COUNT(*)</code>	compte le nombre de n-uplets renvoyés

- D'autres fonctions peuvent exister (en fonction du SGBD).

## Opérateurs d'agrégation : exemples (1)

- Compter le nombre d'emprunts (lignes) effectués par l'abonné 145  

```
SELECT COUNT(*)  
FROM Emprunt  
WHERE NumAbo = 145;
```
- Compter le nombre de valeurs non NULL dans la colonne DateEmp (y compris identiques) pour les emprunts effectués par l'abonné 145  

```
SELECT COUNT(DateEmp)  
FROM Emprunt  
WHERE NumAbo = 145;
```
- Compter le nombre de valeurs distinctes dans la colonne DateEmp pour les emprunts effectués par l'abonné 145  

```
SELECT COUNT(DISTINCT DateEmp)  
FROM Emprunt  
WHERE NumAbo = 145;
```

## Opérateurs d'agrégation : exemples (2)

- Date du dernier emprunt du livre numéro 9782207257357

```
SELECT MAX(DateEmp)
FROM Emprunt
WHERE ISBN = '9782207257357';
```

- Durée moyenne des emprunts (en nombre de jours)

```
SELECT AVG(DateRet - DateEmp)
FROM Emprunt;
```

- Durée maximale des emprunts (en nombre de semaines)

```
SELECT MAX((DateRet - DateEmp)/7)
FROM Emprunt;
```

## Opérateurs d'agrégation : exemples (3)

- On peut renommer les colonnes

```
SELECT MAX(DateRet - DateEmp) AS "Durée Maximale"  
FROM Emprunt;
```

- Durée maximale des emprunts.
- Renommage : s'il comporte des espaces, le nouveau nom doit être entouré de *guillemets*

- Sous-requête

```
SELECT *  
FROM Emprunt  
WHERE DateRet - DateEmp =  
      (SELECT MAX(DateRet - DateEmp)  
       FROM Emprunt);
```

- Liste des emprunts de durée maximale (il peut y en avoir plusieurs).



# Regroupement des n-uplets en sortie

- Clause GROUP BY
  - Regrouper les n-uplets qui ont la ou les mêmes valeurs pour un ou plusieurs attributs
  - Une seule ligne en sortie pour chaque groupe formé
  - Fonction d'agrégation calculée pour chaque groupe

## Exemple

Numéro ISBN et nombre d'emprunts de chaque livre

```
SELECT ISBN, COUNT(*)  
FROM Emprunt  
GROUP BY ISBN;
```

# Regroupement des n-uplets en sortie

- Les emprunts sont groupés par ISBN puis agrégés

Emprunt

NumEmp	ISBN	NumAbo	...
1	1234123410	1	...
3	1234123410	2	...
123	1234123410	33	...
2	1234123425	3	...
4	1234123425	2	...
12	1234123425	9	...
15	1234123425	15	...
101	1234123425	2	...
4	1234123475	3	...
21	1234123475	12	...
...	...	...	...

Résultat

ISBN	COUNT(*)
1234123410	3
1234123425	5
1234123475	2
...	...

# Regroupement sur plusieurs attributs

- Une ligne pour chaque combinaison de valeurs des attributs du GROUP BY
  - Nombre d'emprunts de chaque livre pour chaque abonné.

```
SELECT ISBN, NumAbo, COUNT(*)  
FROM Emprunt  
GROUP BY ISBN, NumAbo;
```

ISBN	NumAbo	COUNT(*)
1234123410	1	1
1234123410	2	1
1234123425	2	2
1234123425	3	1
1234123475	3	1
...	...	...

# Attributs de regroupement

- Le regroupement doit porter sur un attribut de la clause SELECT
- Exemple erroné

```
SELECT Abonné.NomAbo, MAX(DateRet - DateEmp)
FROM Abonné, Emprunt
WHERE Abonné.NumAbo = Emprunt.NumAbo
GROUP BY Emprunt.NumAbo;
```

- Erreur : l'attribut Emprunt.NumAbo n'apparaît pas dans la clause SELECT
- Exemple corrigé

```
SELECT Abonné.NumAbo, NomAbo, MAX(DateRet - DateEmp)
FROM Abonné, Emprunt
WHERE Abonné.NumAbo = Emprunt.NumAbo
GROUP BY Abonné.NumAbo;
```

- Correct : l'attribut Abonné.NumAbo apparaît dans la clause SELECT

# Condition de sélection des groupes (1)

- Clause HAVING : condition de sélection sur l'agrégat
- Exemple
  - Numéro d'abonné et nombre d'emprunts pour les abonnés ayant fait strictement plus de deux emprunts

```
SELECT NumAbo, COUNT(*)  
FROM Emprunt  
GROUP BY NumAbo  
HAVING COUNT(*) > 2;
```

- Renommage : s'il comporte des espaces, le nouveau nom doit être entouré de *guillemets*

```
SELECT NumAbo, COUNT(*) AS "Nombre d'Emprunts"  
FROM Emprunt  
GROUP BY NumAbo  
HAVING "Nombre d'Emprunts" > 2;
```

## Condition de sélection des groupes (2)

- Exemple

```
SELECT NumAbo, MAX(DateRet - DateEmp) AS "Durée maximale"  
FROM Emprunt  
GROUP BY NumAbo  
HAVING MAX(DateRet - DateEmp) =  
        (SELECT MAX(DateRet - DateEmp)  
         FROM Emprunt);
```

- Sous-requête : durée maximale des emprunts.
- Requête principale : calcule la durée maximale des emprunts pour chaque abonné et la compare à la valeur renvoyée par la sous-requête.
- Résultat : numéro d'abonné et durée maximale d'emprunt pour les abonnés ayant emprunté un livre le plus longtemps.

# Exercice

- Afficher les informations du dernier abonné
- Afficher le nombre de livres distincts qui ont été emprunté au moins une fois
- Afficher pour chaque abonné la durée totale de ses emprunts
- Afficher pour chaque livre emprunté le nombre de fois qu'il a été emprunté
- Afficher pour chaque livre emprunté la durée moyenne de l'emprunt
- Afficher pour chaque livre emprunté la durée totale des emprunts

- 1 Notations et exemple fil-rouge
- 2 Langage de manipulation des données
- 3 Agrégation et regroupement
- 4 SQL : Jointures avancées**



# Opérateur JOIN

- Jointure (interne) entre deux tables

```
SELECT ...  
FROM table1 JOIN table2  
ON table1.att11 op1 table2.att12 [AND|OR  
ON table1.att21 op2 table2.att22 [AND|OR  
ON ... ]];
```

## Exemple

```
SELECT Abonné.*  
FROM Abonné JOIN Emprunt  
ON Abonné.NumAbo = Emprunt.NumAbo;
```

- Opérateur arithmétique op<sub>i</sub> de comparaison : =, <>, <, >, <=, >=
- On peut aussi écrire INNER JOIN

## Opérateurs LEFT JOIN et RIGHT JOIN

- Jointure externe gauche et droite entre deux tables

```
SELECT ...
```

```
FROM table1 [LEFT|RIGHT] JOIN table2
```

```
ON table1.att11 op1 table2.att12 [AND|OR
```

```
ON table1.att21 op2 table2.att22 [AND|OR
```


```
ON ... ]];
```

- LEFT JOIN : inclure tous les n-uplets de table<sub>1</sub> même s'ils n'ont pas de n-uplet lié dans table<sub>2</sub>
- RIGHT JOIN : inclure tous les n-uplets de table<sub>2</sub> même s'ils n'ont pas de n-uplet lié dans table<sub>1</sub>

# Opérateurs LEFT JOIN et RIGHT JOIN (2)

- Exemple

```
SELECT *
FROM Abonné LEFT JOIN Emprunt
ON Abonné.NumAbo = Emprunt.NumAbo;
```



NumAbo	NomAbo	PrénomAbo	DateAbo	NumEmp	ISBN	NumAbo	DateEmp	DateRet
1	Dupont	Philippe	2021-09-17	3	9782212112818	1	2021-10-28	2021-11-10
1	Dupont	Philippe	2021-09-17	4	9782212112818	1	2021-11-08	NULL
2	Durand	Arthur	2009-01-02	1	9782225805158	2	2021-09-06	2021-09-20
3	Dupont	Charlie	2015-05-03	2	9782225805158	3	2021-09-25	2021-10-11
4	Ducros	Marie	2020-07-04	NULL	NULL	NULL	NULL	NULL
5	Vernier	Alain	2021-09-15	NULL	NULL	NULL	NULL	NULL

- Les abonnés n'ayant pas fait d'emprunt apparaissent dans le résultat.
- Attributs non renseignés : valeur NULL

# Jointures imbriquées

- Il est possible de combiner les opérateurs JOIN, LEFT JOIN et RIGHT JOIN.

- Exemple

```
SELECT * FROM Livre
JOIN (Ecrit JOIN Auteur
      ON Auteur.NumAut = Ecrit.NumAut)
ON Ecrit.ISBN = Livre.ISBN;
```

- Une jointure externe peut être imbriquée dans une jointure interne.
- La réciproque n'est pas vraie.

## Exercice

- Afficher les noms et prénoms des abonnés qui n'ont pas encore rendu un livre emprunté
- Afficher les noms et prénoms des abonnés qui ont déjà emprunté un livre par un auteur de même prénom qu'eux
- Titre des livres écrits par (au moins) deux auteurs qui portent le même prénom

