Semestre 2

BASES DE DONNÉES LANGAGE SQL

Bases de données

Enjeux

Enjeux : BigData

- La Recherche n° 482 Décembre 2013 Big Data – Traitement massif des données.

- Pour la Science N° spécial n°433 Novembre 2013 Big Bang numérique – Les données massives changent-elles le monde ?

Bases de données

Algorithmique, programmation et structures de données

Choix d'une structure de données

Le choix d'un algorithme et la complexité d'un programme informatique dépendent de façon cruciale du choix d'une structure de représentation des données.

Types simples

	Exemples	Type	Type Python	Remarques Compléments	Programme CPGE
Types simples	-5	Entier	Int (integer)	Voir chapitre « Représentation des nombres »	✓ 1ère année
	3.1415	Flottant	float		✓ 1ère année
	True	Booléen	bool (boolean)		✓ 1ère année

Types composés

		Exemples	Type	Type Python	Remarques Compléments	Programme CPGE
Types composés	***Séquences***	(‘a’, 1, False, 1.5)	n-uplets	tuple	Ces objets sont, en python, des « itérables » : “An object capable of returning its members one at a time. Include all sequence types (list, str, tuple) and some non-sequence types like dict and file. Iterables can be used in a *for* loop.”	
		[1, 3, 5]	**Liste** (1)	list		✓ 1ère année
		‘abcd’ ou "abcd"	**Chaîne** (2)	str (string)		✓ 1ère année
	{1, ‘a’, True}		Ensemble	set	Mêmes propriétés que les ensembles mathématiques	
	{"imp" : (1,3,5), "pair" : (2,4,6)}		Dictionnaire	dict		

Autres types

	Exemples	Type	Type Python	Remarques Compléments	Programme CPGE
Autres types		Fichier	file	Voir TP n°5	✓ 1ère année
	numpy.array([[1,2], [3,4]])	Array (1) (Matrice)	numpy.ndarray	Voir TP « Pivot de Gauss »	✓ 1ère année
		Pile			✓ 2de année
		Arbre			
		File ou queue			

Algorithmique, programmation et structures de données

Malgré la grande variété de structures de données disponibles, certains problèmes nécessitent une ***architecture différente*** (données apparaissant plusieurs fois, données reliées entre elles de façon complexe …).

Les ***bases de données*** ont été conçues pour répondre à ce type de besoins.

Bases de données

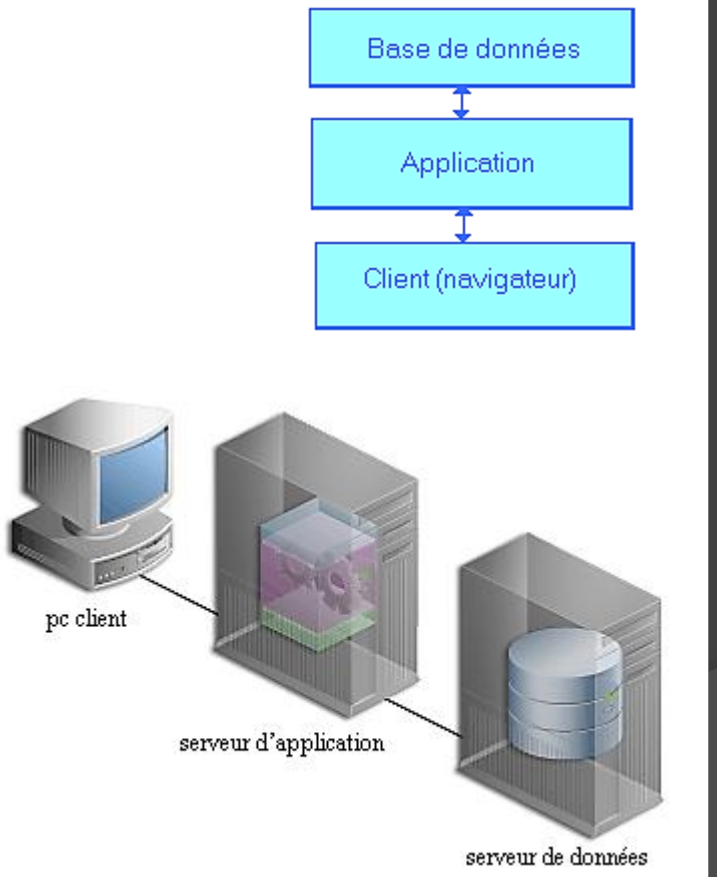
Architecture 3 tiers

Pages web dynamiques

Ce nom provient de l'anglais *tier* signifiant *étage* ou *niveau*.

Il s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles (étages, niveaux, tiers ou strates) dont le rôle est clairement défini :

1. **la présentation des données (affichage, dialogue avec l'utilisateur) ;**
2. **le traitement des données ;**
3. **l'accès aux données persistantes (données destinées à être conservées dans la durée, voire de manière définitive).**



Bases de données

Introduction aux bases de données

Gestion d'une bibliothèque : tableur

index	titre	genre	auteurN	auteurP	nom	prénom	tél	date	retour
1	Salambo	Roman	Flaubert	Gustave	Castel	Claude	0612345112	12-août-10	23-août-10
2	93	Roman	Hugo	Victor	Le Dray	Camille	0422113827	17-sept.-10	03-oct.-10
3	Mme Bovary	Littérature	Flaubert	Gustave	Loutard	Annie	064331282	01-nov.-10	02-déc.-10
1	Salambô	Roman	Flaubert	Gustave	Filatre	Jean	0432168719	02-sept.-10	14-sept.-10
4	Aphorismes	Littérature	Wilde	Oscar	Castel	Marie	0412324494		
5	L'immoraliste	Roman	Gide	André	Biraud	Michèle	0434578612	05-févr.-11	
3	Madame Bovary	Roman	Flaubert	Gustave	Castel	Claude	0412324494	06-janv.-10	
2	Quatrevingt-treize	Roman	Hugo	Victor	Filâtre	Jean	0432168719	20-oct.-10	03-nov.-10
1	Salambo	Littérature	Flaubert	gustave	Ledray	Camille	0422113627	16-sept.-10	27-sept.-10
…	…	…	…	…	…	…	…	…	…

Limites des structures de données plates pour le stockage et la recherche d'informations

- ***Intégrité / cohérence des données***
  (doublons orthographiques, formats variables $\Rightarrow$ problèmes pour les recherches).

- ***Efficacité des recherches***
  (lecture séquentielle de la totalité du tableau $\Rightarrow$ complexité en temps).

Gestion d'une bibliothèque : base de données

Solution : ***modèle relationnel***
Les données sont stockées en fonction des ***relations*** qu'elles entretiennent entre elles.
À chaque ***relation*** correspond une ***table*** de la base de données.

Dans l'exemple ci-dessus, il paraît judicieux de prévoir une *table* « Auteurs » contenant le nom, le prénom ainsi qu'un numéro d'identification unique (id) en cas d'homonymie (ci-contre).

Auteurs		
id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	Wilde	Oscar
…	…	…

Pour communiquer et effectuer des opérations (recherches, traitements) sur les bases de données, on utilise un ***système de gestion de base de données*** (abr. ***SGBD*** et en anglais DBMS pour database management system) : SQLite, MySQL (Oracle), PostgreSQL…

Un *langage spécifique* nommé ***SQL*** (Structured Query Language) ou langage de requêtes structurées permet d'effectuer des ***requêtes*** (opérations) sur les bases de données.

Il existe des *interfaces graphiques* permettant d'effectuer ces *requêtes* dans un environnement graphique (permettant également de visualiser les tables et de les modifier) : phpMyAdmin, SQLite Database Browser…

Extrait du fichier initial :

index	titre	genre	auteurN	auteurP	nom	prénom	tél	date	retour
1	Salambo	Roman	Flaubert	Gustave	Castel	Claude	0612345112	12-août-10	23-août-10

Livres

id	auteur	titre	genre
1	1	Salambô	1
2	2	Quatre-vingt-treize	1
3	3	L'immoraliste	1
4	4	Aphorismes	2
5	1	Madame Bovary	1
…	…	…	…

Auteurs

id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	Wilde	Oscar
…	…	…

Emprunts

qui	quoi	date	rendu
1	1	12-août-10	23-août-10
6	1	02-sept.-10	14-sept.-10
2	1	16-sept.-10	27-sept.-10
2	2	17-sept.-10	03-oct.-10
6	2	20-oct.-10	03-nov.-10
3	5	01-nov.-10	03-déc.-10
1	5	06-janv.-11	NULL
5	3	05-févr.-11	NULL
…	…	…	…

Emprunteurs

id	nom	prénom	tél
1	Castel	Claude	0612345112
2	Le Dray	Camille	0422113827
3	Loutard	Annie	0654331282
4	Castel	Marie	0412324494
5	Biraud	Michèle	0434578612
6	Filâtre	Jean	0432168719
…	…	…	…

Genre

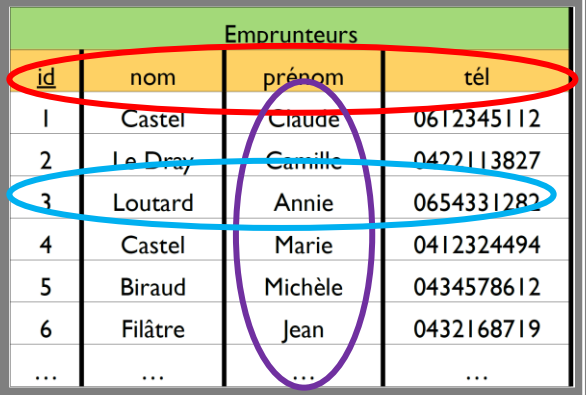
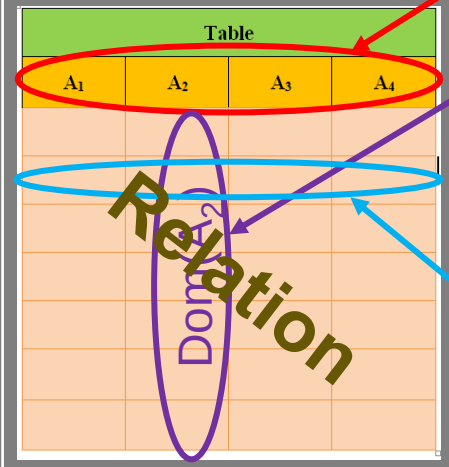
id	genre
1	Roman
2	Littérature
…	…

Aucune répétition : cohérence des données

Bases de données

Introduction au modèle relationnel

Vocabulaire

Exemple	Table formelle	Définitions
		***Attributs*** = noms des colonnes. ***Domaine d'un attribut*** = ensemble des valeurs que peut prendre cet attribut (entier, chaîne…). On note *Dom(A_i)* le domaine de l'attribut A_i. ***n-uplet*** (**t**uple) ou ***enregistrement*** = une ligne. Élément de *Dom(A_1)*×*Dom(A_2)*×*Dom(A_3)*×… On note *t* un tel n-uplet. Ex : si *t* = (3, Loutard Annie, 0654331282) alors *t [nom]* = (Loutard) et *t [id, prénom]* = (3, Annie) ***Relation*** = ensemble des n-uplets d'une table.

Dom(prénom)=chaîne

Une base de données est constituée d'un *nombre fini de relations* : ***une relation par table***.

On dit que la relation notée *r* d'une table suit le ***schéma de relation*** $\boldsymbol{R(A_1, A_2, A_3,\ldots)}$ (i.e. la table associée à la relation *r* met en relation les attributs A_1, A_2, A_3… ou encore associe les attributs A_1, A_2, A_3…les uns aux autres).

L'ensemble des schémas de relation s'appelle le ***schéma de la base de données***.

Notion de clé primaire (Primary Key)

Dans les 4 tables Livres, Auteurs, Emprunteurs et Genre (mais pas dans la table Emprunts) apparaît un *attribut* noté id permettant d'identifier un et un seul *enregistrement* de la table.

Cet attribut est la *clé primaire* qui permet une indexation des données (comme l'index d'une liste), ce qui rend les procédures d'interrogation de la table (*requêtes*) plus efficaces.

Dans la table Emprunts, la *clé primaire* est constituée des *3 attributs* qui, quoi, date.

De façon simplifiée, une ***clé*** (pour une relation *r*) est un **ensemble *K* d'attributs** qui donne accès à un unique enregistrement :

$$\forall\ t, \mathrm{u} \in \mathrm{r},\ \mathrm{t[K]} = u\mathrm{[K]} \Rightarrow \mathrm{t} = \mathrm{u}$$

(i.e. si les enregistrements *t* et *u* sont identiques pour les attributs du sous ensemble *K* des attributs de la table alors ils sont identiques pour tous les autres attributs de la table).

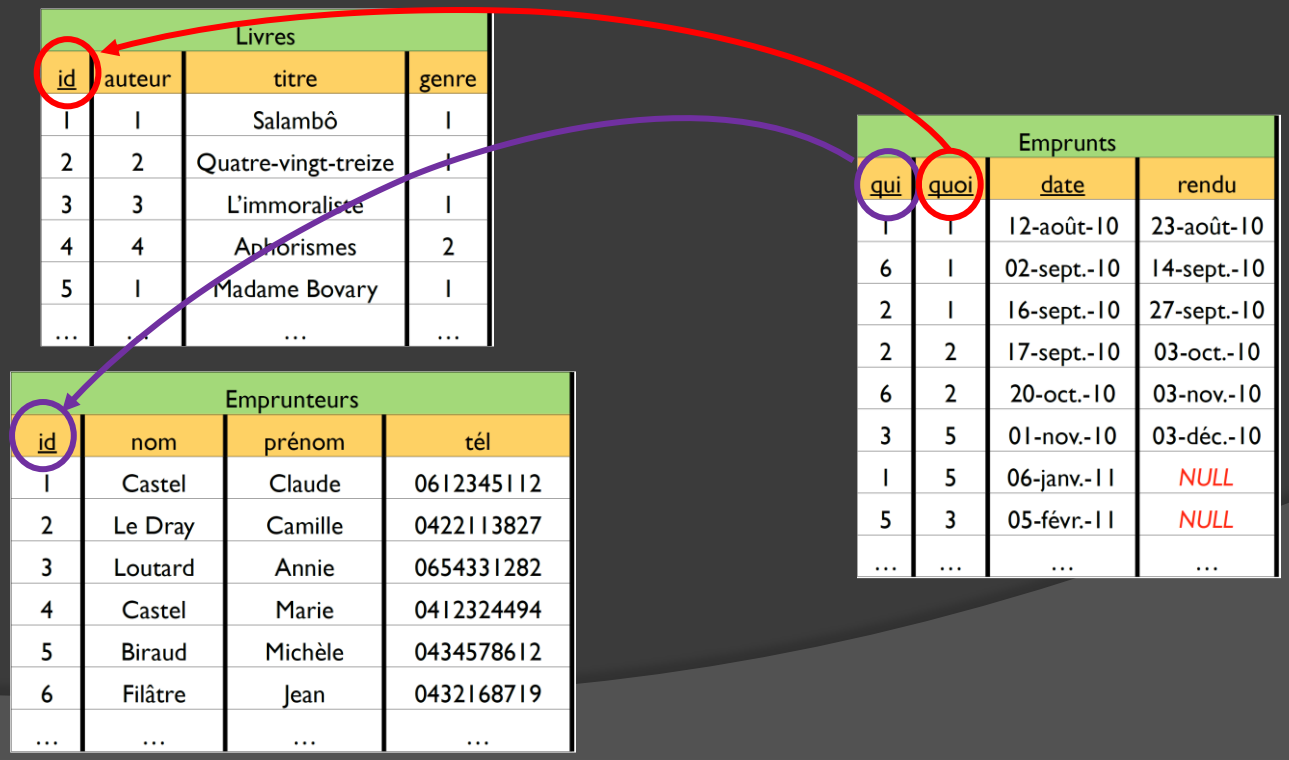
Une ***clé primaire*** est une **clé de taille minimale** (souvent notée id ou PK pour primary key).

Une ***clé secondaire*** est une **clé autre que la clé primaire**.

Notion de clé étrangère (Foreign Key)

La table Emprunts possède un attribut quoi (noté Emprunts.quoi) qui fait le lien avec l'attribut id de la table Livres (noté Livres.id). Cet attribut constitue une ***clé étrangère***. De même l'attribut Emprunts.qui est une *clé étrangère* faisant le lien avec l'attribut Emprunteurs.id.

Une clé étrangère fait référence à une clé primaire d'une autre table.



Livres

id	auteur	titre	genre
1	1	Salambô	1
2	2	Quatre-vingt-treize	1
3	3	L'immoraliste	1
4	4	Aphorismes	2
5	1	Madame Bovary	1
…	…	…	…

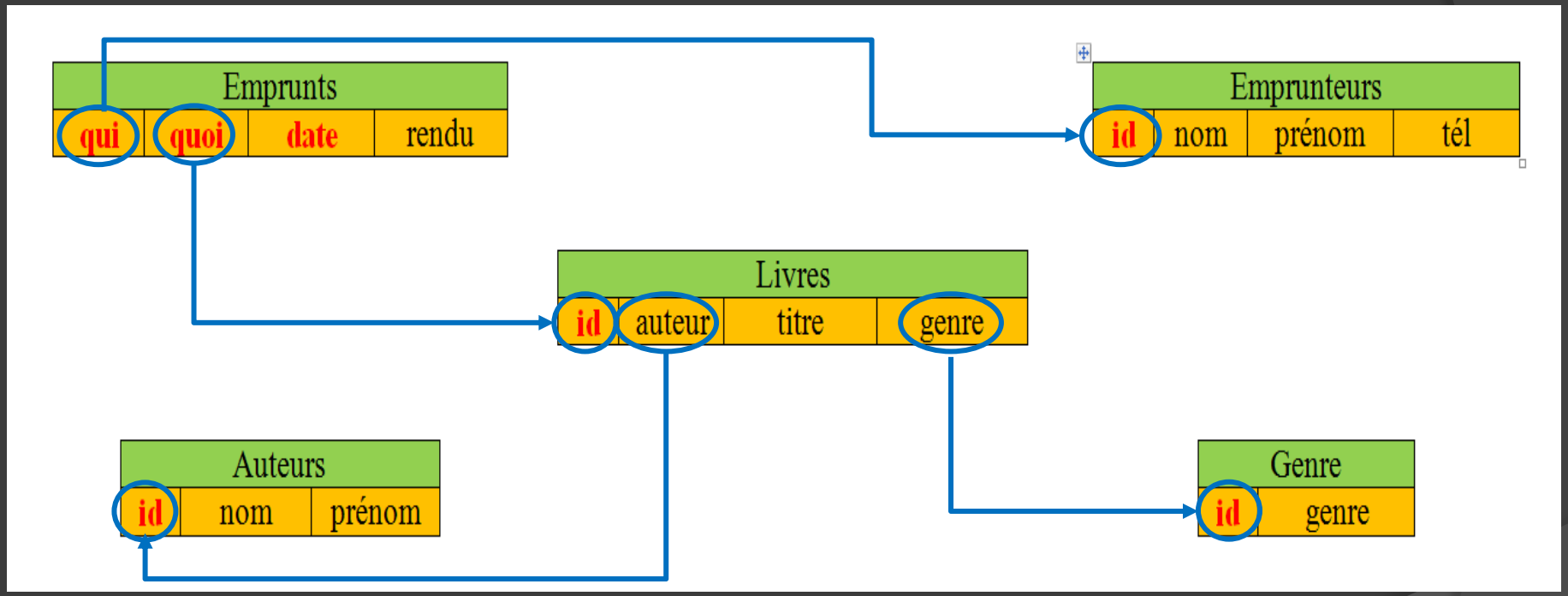
Emprunteurs

id	nom	prénom	tél
1	Castel	Claude	0612345112
2	Le Dray	Camille	0422113827
3	Loutard	Annie	0654331282
4	Castel	Marie	0412324494
5	Biraud	Michèle	0434578612
6	Filâtre	Jean	0432168719
…	…	…	…

Emprunts

qui	quoi	date	rendu
1	1	12-août-10	23-août-10
6	1	02-sept.-10	14-sept.-10
2	1	16-sept.-10	27-sept.-10
2	2	17-sept.-10	03-oct.-10
6	2	20-oct.-10	03-nov.-10
3	5	01-nov.-10	03-déc.-10
1	5	06-janv.-11	NULL
5	3	05-févr.-11	NULL
…	…	…	…

Graphe des clés étrangères



Bases de données

Opérateurs de l'algèbre relationnelle – Langage SQL

On appelle ***formule de sélection*** *F* une formule construite à partir des attributs, des fonctions usuelles, de constantes, des opérateurs de comparaison (<,>,=…), de connecteurs logiques (et, ou, non).

Exemple :

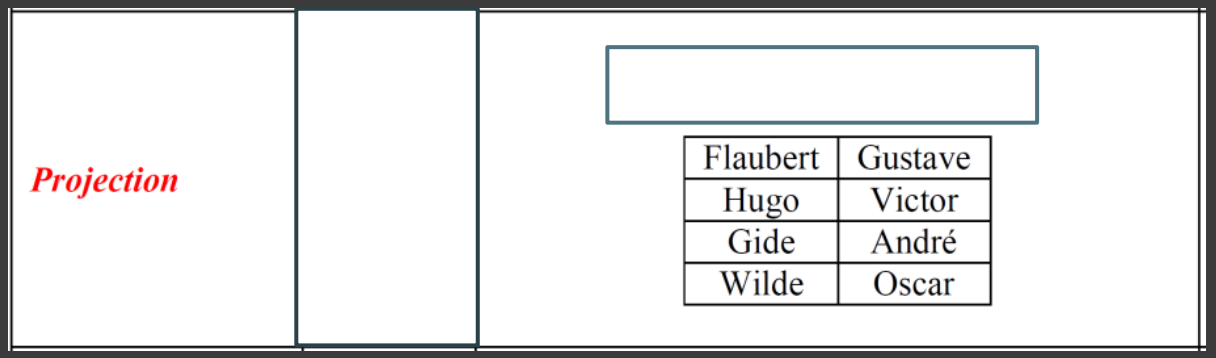
nom= « Loutard » ET date < 1/1/2011.

Affichage

Opération	*Résultat (exemple)*
Affichage	Affichage(Auteurs) 1 Flaubert Gustave 2 Hugo Victor 3 Gide André 4 Wilde Oscar

Requête en langage SQL
SELECT * FROM Auteurs

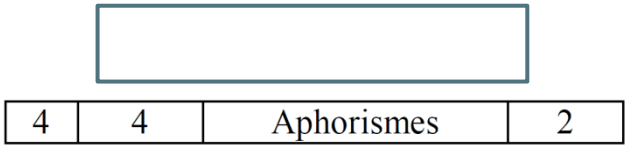
Projection




```
SELECT
        nom,prénom
FROM
        Auteurs
```

Sélection ou restriction

Sélection ou restriction



4	4	Aphorismes	2

```
SELECT
        *
FROM
        Livres
WHERE
        genre=2        #( 2 ↔ Littérature)
```

Jointure

Jointure

1	1	Salambô	1	Roman
2	2	Quatre-vingt-treize	1	Roman
3	3	L'immoraliste	1	Roman
4	4	Aphorismes	2	Littérature
5	1	Madame Bovary	1	Roman

```
SELECT
        *
FROM
        (Livres JOIN Genre
                ON Livres.genre=Genre.id)
```

La jointure est en général associée à d'autres opérateurs.

Agrégation

Agrégation		6

	SELECT COUNT(*) FROM Emprunteurs

Renommage

Renommage		23-août-10 14-sept.-10 27-sept.-10 03-oct.-10 03-nov.-10 03-déc.-10

Renommer un attribut (utile en cas de jointure de tables ayant des attributs de même nom). On peut aussi préciser la table pour éviter toute ambiguïté (table.nom_attribut).	SELECT rendu ***AS*** dateRetour FROM Emprunts