CC BY NC SA

Bases de données : un peu plus loin dans SQL

Karën Fort

karen.fort@sorbonne-universite.fr

Sources d'inspiration

- Cours de B. Habert (ENS Lyon)
- Cours de G. Lejeune (SU)
- https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1961564-selectionnez-des-donnees
- https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/916084-les-jointures-entre-tables

Retours sur le TD

Agencer les résultats

Rechercher, approximativement

Combiner les informations de plusieurs tables

Pour finir

Requêtes : correction

1. SELECT filiere FROM Programme (ou, sans doublon, SELECT DISTINCT filiere FROM Programme)
2. SELECT auteur FROM Livres (ou trié, sans doublon : SELECT DISTINCT auteur FROM Livres ORDER BY auteur)
3. SELECT titre FROM Livres WHERE nb_pages=150
4. SELECT nom, prenom FROM Etudiants WHERE prenom="Marie"
5. SELECT titre FROM Emprunts WHERE date_ret IS NULL
6. SELECT num_et FROM Emprunts WHERE date_ret IS NULL

Retours sur le TD

Agencer les résultats
- Trier les résultats
- Limiter les résultats

Rechercher, approximativement

Combiner les informations de plusieurs tables

Pour finir

Retours sur le TD

Agencer les résultats
- Trier les résultats
- Limiter les résultats

Rechercher, approximativement

Combiner les informations de plusieurs tables

Pour finir

Ordonner les résultats

Dans une table d'un SGBDR :

- Chaque ligne représente de manière indépendante des autres lignes les informations connues sur une entité du domaine visé
- L'ordre des lignes
  - n'est pas significatif
  - n'est pas utilisable

Ordonner les résultats : ORDER BY

ORDER BY (colonne)

specifie l'ordre d'apparition des donnees dans le resultat :

- par défaut, tri ascendant
- sinon, on peut spécifier : ASC pour ascendant, DESC pour descendant

possibilité de trier sur plusieurs colonnes

ORDER BY : exemple

Dans la BD Enquete

SELECT DISTINCT auteur FROM Livres ORDER BY auteur

Par défaut

SELECT DISTINCT auteur FROM Livres ORDER BY auteur

Tout afficher | Nombre de lignes : 25

+ Options

←T→	auteur ▲ 1
Éditer Copier Supprimer	Anonyme
Éditer Copier Supprimer	Baleine
Éditer Copier Supprimer	Carmin
Éditer Copier Supprimer	Corneille
Éditer Copier Supprimer	Godoc
Éditer Copier Supprimer	Larsson
Éditer Copier Supprimer	Levi
Éditer Copier Supprimer	Molto
Éditer Copier Supprimer	Picasso
Éditer Copier Supprimer	Saint-Exupery

Avec DESC

SELECT DISTINCT auteur FROM Livres ORDER BY auteur DESC

Tout afficher | Nombre de lignes : 25

+ Options

←T→	auteur ▼ 1
Éditer Copier Supprimer	Saint-Exupery
Éditer Copier Supprimer	Picasso
Éditer Copier Supprimer	Molto
Éditer Copier Supprimer	Levi
Éditer Copier Supprimer	Larsson
Éditer Copier Supprimer	Godoc
Éditer Copier Supprimer	Corneille
Éditer Copier Supprimer	Carmin
Éditer Copier Supprimer	Baleine
Éditer Copier Supprimer	Anonyme

Retours sur le TD

Agencer les résultats
- Trier les résultats
- Limiter les résultats

Rechercher, approximativement

Combiner les informations de plusieurs tables

Pour finir

Supprimer les doublons : SELECT DISTINCT

Tous les codes équipes dans la table Coureur sans doublon

```
SELECT DISTINCT CodeEquipe
FROM Coureur
```

Restreindre le nombre de lignes : LIMIT

LIMIT

deux paramètres :

- le nombre de lignes que l'on veut récupérer
- le décalage, introduit par le mot-clé OFFSET, qui indique à partir de quelle ligne on récupère les résultats (facultatif)

LIMIT : exemple

Dans la BD Enquete

SELECT DISTINCT auteur FROM Livres LIMIT 4

Par défaut

```
SELECT DISTINCT auteur FROM Livres LIMIT 4
```

+ Options

←T→			auteur
Éditer	Copier	Supprimer	Baleine
Éditer	Copier	Supprimer	Molto
Éditer	Copier	Supprimer	Carmin
Éditer	Copier	Supprimer	Larsson

Avec OFFSET 2

```
SELECT DISTINCT auteur FROM Livres LIMIT 4 OFFSET 2
```

+ Options

←T→			auteur
Éditer	Copier	Supprimer	Carmin
Éditer	Copier	Supprimer	Larsson
Éditer	Copier	Supprimer	Anonyme
Éditer	Copier	Supprimer	Picasso

Retours sur le TD

Agencer les résultats

Rechercher, approximativement
- Recherches approximatives dans les chaînes de caractères
- Recherches approximatives dans un intervalle
- Recherches approximatives dans des listes de valeurs

Combiner les informations de plusieurs tables

Pour finir

Retours sur le TD

Agencer les résultats

Rechercher, approximativement
- Recherches approximatives dans les chaînes de caractères
- Recherches approximatives dans un intervalle
- Recherches approximatives dans des listes de valeurs

Combiner les informations de plusieurs tables

Pour finir

LIKE

LIKE

permet de faire des recherches en utilisant deux "jokers", des caractères qui représentent n'importe quel caractère :

- % : représente n'importe quelle chaîne de caractères, quelle que soit sa longueur (y compris une chaîne de longueur 0)
- _ : qui représente un seul caractère

LIKE : exemple

Dans la BD Enquete

SELECT auteur FROM Livres WHERE auteur LIKE 'Pic%' (ou 'Pic_sso')

Commençant par "Pic"

```
SELECT auteur FROM Livres WHERE auteur LIKE 'Pic%'
```

Tout afficher | Nombre de lignes : 25

+ Options

←T→	auteur
Éditer Copier Supprimer	Picasso

Lettre manquante (a)

```
SELECT auteur FROM Livres WHERE auteur LIKE 'Pic_sso'
```

Tout afficher | Nombre de lignes : 25

+ Options

←T→	auteur
Éditer Copier Supprimer	Picasso

Recherche négative : NOT LIKE

Rechercher une chaîne de caractères qui ne contient pas. . .

SELECT auteur FROM Livres WHERE auteur NOT LIKE '%a%'

```
SELECT auteur FROM Livres WHERE auteur NOT LIKE '%a%'
```

Tout afficher | Nombre de lignes : 25

+ Options

←T→	auteur
Éditer Copier Supprimer	Molto
Éditer Copier Supprimer	Corneille
Éditer Copier Supprimer	Levi
Éditer Copier Supprimer	Godoc

Retours sur le TD

Agencer les résultats

Rechercher, approximativement
- Recherches approximatives dans les chaînes de caractères
- Recherches approximatives dans un intervalle
- Recherches approximatives dans des listes de valeurs

Combiner les informations de plusieurs tables

Pour finir

BETWEEN...AND

SELECT filiere, nb_heures FROM Programme WHERE nb_heures BETWEEN 1 AND 2

```
SELECT filiere, nb_heures FROM Programme WHERE nb_heures BETWEEN 1 and 2
```

Tout afficher | Nombre de lignes : 25 Filtrer les

+ Options

filiere	nb_heures
Informatique	2
Informatique	2
Informatique	2
Sciences pour l'ingenieur	1
Sciences pour l'ingenieur	2
Sciences du langage	2
Sciences du langage	2
Sciences de l'education	2
Sciences de l'education	1

Retours sur le TD

Agencer les résultats

Rechercher, approximativement
- Recherches approximatives dans les chaînes de caractères
- Recherches approximatives dans un intervalle
- Recherches approximatives dans des listes de valeurs

Combiner les informations de plusieurs tables

Pour finir

IN

SELECT * FROM Livres WHERE auteur IN ('Picasso', 'Anonyme')

Retours sur le TD

Agencer les résultats

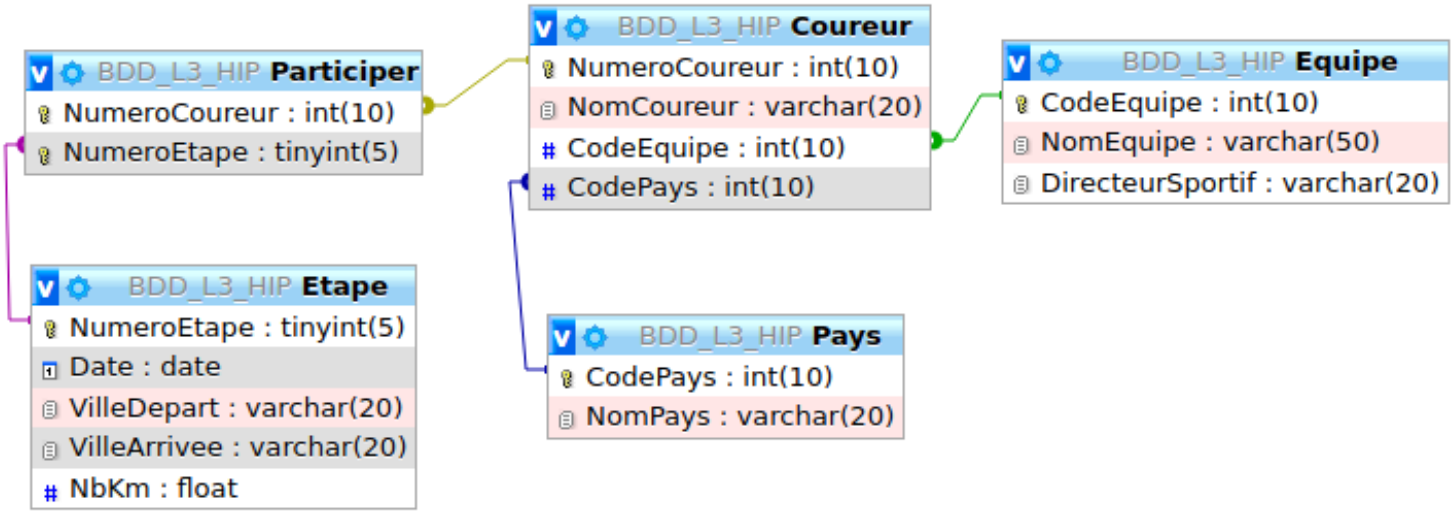
Rechercher, approximativement

Combiner les informations de plusieurs tables
- Jointures internes
- Jointures externes

Pour finir

BD Tour de France

Comment savoir de quel pays vient un coureur ?



→ dans la table Coureur, on a le code du pays, mais pas le nom du pays, qui lui est dans la table Pays, il faut donc lier les deux tables

Retours sur le TD

Agencer les résultats

Rechercher, approximativement

Combiner les informations de plusieurs tables
- Jointures internes
- Jointures externes

Pour finir

Définition

Jointures internes

ne sélectionnent que les données qui ont une correspondance entre les deux tables (ici, par CodePays)

Exemple (BDD_L3_HIP) :
SELECT NomCoureur, NomPays FROM Pays, Coureur WHERE Coureur.CodePays=Pays.CodePays

```
SELECT NomCoureur, NomPays from Pays,Coureur WHERE Coureur.CodePays=Pays.CodePays
```

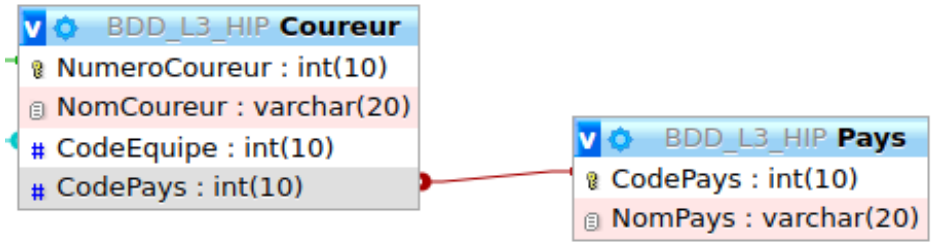
Tout afficher | Nombre de lignes : 25 Filtrer les lignes:

+ Options

NomCoureur	NomPays
Alaphilippe	France
Devenyns	Belgique
Lampaert	Belgique

Explications

SELECT NomCoureur, NomPays FROM Pays, Coureur WHERE Coureur.CodePays =Pays.CodePays



- CodePays apparaît dans deux tables (Pays et Coureur)
- c'est ce qui permet de lier les deux tables entre elles
- mais CodePays est ambigü (il y en a deux), il faut donc préfixer CodePays par le nom de la table correspondante : **Coureur**.CodePays pour le CodePays de **Coureur**, par exemple

Changement de syntaxe

La nouvelle syntaxe est plus explicite

```
SELECT NomCoureur, NomPays FROM Pays, Coureur WHERE
Coureur.CodePays =Pays.CodePays
```

$\downarrow$

```
SELECT NomCoureur, NomPays FROM Pays INNER JOIN Coureur
ON Coureur.CodePays =Pays.CodePays
```

Retours sur le TD

Agencer les résultats

Rechercher, approximativement

Combiner les informations de plusieurs tables
- Jointures internes
- Jointures externes

Pour finir

Définition

Jointures externes

permettent de récupérer toutes les données, même celles qui n'ont pas de correspondance (par exemple, un coureur qui n'aurait pas de pays).

Exemple :
`SELECT` NomCoureur, NomPays `FROM` Pays `LEFT JOIN` Coureur `ON` Coureur.CodePays =Pays.CodePays

`LEFT JOIN` demande à récupérer tout le contenu de la table de gauche (ici, Pays), donc tous les pays, même si ces derniers n'ont pas d'équivalence dans la table Coureur (même si aucun coureur ne vient de ce pays).

LEFT JOIN

SELECT NomCoureur, NomPays FROM Pays LEFT JOIN Coureur ON Coureur.CodePays =Pays.CodePays

```
SELECT NomCoureur, NomPays FROM Pays LEFT JOIN Coureur ON Coureur.CodePays =Pays.CodePays
```

Profilage

Tout afficher | Nombre de lignes : 25 Filtrer les lignes: Cherche

+ Options

NomCoureur	NomPays
Alaphilippe	France
Devenyns	Belgique
Lampaert	Belgique

→ dans ce cas (BDD_L3_HIP), même résultat qu'avec une jointure interne

LEFT JOIN

SELECT NomCoureur, NomPays FROM Pays LEFT JOIN Coureur ON Coureur.CodePays =Pays.CodePays

```
SELECT NomCoureur, NomPays FROM Pays LEFT JOIN Coureur ON Coureur.CodePays =Pays.CodePays
```

Profilage

Tout afficher | Nombre de lignes : 25 Filtrer les lignes: Cherch

+ Options

NomCoureur	NomPays
Geraint Thomas	Royaume-Uni
Egan Bernal	Colombie
Peter Sagan	Slovaquie
Jonathan Castroviejo	Espagne
Kasper Asgreen	Danemark
Romain Bardet	France
Sébastien Reichenbach	Suisse
Stefan Küng	Suisse
NULL	Italie
NULL	Portugal
NULL	Luxembourg
NULL	Allemagne

→ dans ce cas (BDD_L3_FORT), résultat différent (certains pays n'ont pas de coureurs dans la BD)

RIGHT JOIN

Tout pareil, mais avec la table "de droite"

SELECT NomCoureur, NomPays FROM Pays RIGHT JOIN Coureur ON Coureur.CodePays =Pays.CodePays

SELECT NomCoureur, NomPays FROM Pays RIGHT JOIN Coureur ON Coureur.CodePays =Pays.CodePays

Profilage

Tout afficher | Nombre de lignes : 25 Filtrer les lignes: Cherche

+ Options

NomCoureur	NomPays
Sébastien Reichenbach	Suisse
Peter Sagan	Slovaquie
Romain Bardet	France
Stefan Küng	Suisse
Egan Bernal	Colombie
Kasper Asgreen	Danemark
Jonathan Castroviejo	Espagne
Geraint Thomas	Royaume-Uni

→ dans ce cas (BDD_L3_FORT), même résultat qu'avec une jointure interne (parce que tous les coureurs ont un pays)

Retours sur le TD

Agencer les résultats

Rechercher, approximativement

Combiner les informations de plusieurs tables

Pour finir
- CQFR : Ce Qu'il Faut Retenir
- TD

- maîtrise de : `ORDER BY`, `LIMIT`, `LIKE`, `BETWEEN`, `IN`
- jointures

TD de Marie Duflot (Loria) à finir

https://members.loria.fr/MDuflot/files/med/doc/BD/bdd.pdf

Finir le TD, en utilisant phpMyAdmin et la base BDD_L3_Enquete