

Bases de Données 2

#1 - SQL

Matthieu Nicolas
Polytech S5 - II

Slides réalisées à partir de celles de Claude Godart et Malika Smaïl

Plan

- Organisation des cours
- Présentation du module
- SQL
- Interrogation d'une BD avec SQL
- Outils

Organisation des cours

Base de Données 2
#1

Organisation des TDs

- Dorénavant, à **distance**
 - Vous encourage à participer aux cours en synchrone...
 - ... mais mode asynchrone possible
 - Prévenir à l'avance et à minimiser

Déroulement des TDs - 1

- **Commence par une partie cours puis enchaîne avec le TD correspondant**
- **Pour ceux qui font le TD en synchrone**
 - **Vous vous répartissez par groupe de 3-4 dans les différents channels vocaux du Discord**
 - **Pour travailler ensemble et discuter**

Déroulement des TDs - 2

- Je suis à votre **dispo** durant le **créneau** du TD
- **Si vous avez une question** : envoyez moi un **DM** et je viens quand je suis dispo
- En attendant vos questions, je fais ma “ronde”

Déroulement des TDs - 3

- Pour ceux qui font **le TD en asynchrone**
 - Je mets à dispo les ressources (slides, TDs, VOD)
 - Je suis toujours **joignable par mail/Discord pour vos questions**
 - Possible de convenir d'un créneau pour en discuter en live sur Discord

Présentation du module

Base de Données 2
#1

Organisation

- **Volume horaire : 18h**
- **Enseignement intégré : CM et TD fusionnés**

Objectifs / Syllabus

- **Apprendre à manipuler** une BD directement
 - Interrogation, Création, Modification, Suppression
- **Découvrir les concepts avancés** de BD
 - Vues, Tables Temporaires, Triggers, Transactions
- **Apprendre à manipuler** une BD **via un langage** de programmation
 - PHP

Évaluation

- **2 notes** pour l'évaluation
 - 1 TP noté (probablement la séance du 20/11)
 - 1 projet
- À ce sujet
 - C'est quoi votre **planning en fin d'année** ?
 - Histoire de convenir d'une **date de rendu** du projet

SQL

Base de Données 2 #1

Rappel - Modèle relationnel

- Données organisées sous forme de tables à 2 dimensions, appelées **relations**
- Les lignes de ces tables sont appelées **tuple** ou **n-uplet**
 - Une **ligne** représente un **objet**, une **colonne** un **attribut**
- Données manipulées par des opérateurs de l'algèbre relationnelle
- État cohérent de la base défini par ensemble de contraintes d'intégrités

Rappel - Algèbre relationnelle

- Ensemble d'**opérateurs** sur des relations **produisant** en résultat **de nouvelles relations**
- Objectif : **élaborer de nouvelles informations** à partir des **relations de départ** et d'une **composition séquentielle d'opérateurs**

SQL - 1

- **Structured Query Language**
- **Langage** pour les SGBD relationnels **basé sur l'algèbre relationnelle**
- **Standard ANSI et ISO**
 - Mais quelques différences entre les implémentations des SGBDs...

SQL - 2

- Comporte 3 parties :
 - **Langage de Définition de Données (LDD)** pour créer, modifier et supprimer des définitions de tables (*create table, alter table, drop table*)
 - **Langage de Manipulation de Données (LMD)** pour rechercher, ajouter, modifier et supprimer des données (*select, insert, update, delete*)
 - **Langage de Contrôle de Données (LCD)** pour gérer les droits d'accès aux données (*grant, revoke*)

Interrogation d'une BD avec SQL

Base de Données 2
#1

Syntaxe de base d'une requête SQL - 1

SELECT <liste d'attributs projetés>

FROM <liste de relations>

WHERE <liste de critères de sélection et de jointure>

Syntaxe de base d'une requête SQL - 2

- La partie **SELECT** indique les attributs qui doivent apparaître dans la réponse
 - Doivent appartenir aux relations dans le **FROM**
- La partie **FROM** décrit les relations qui sont utilisées dans la requête
 - Et donc l'ensemble des attributs disponibles
- La partie **WHERE** exprime la ou les conditions que doivent respecter les attributs d'un tuple pour être dans la réponse
 - Partie optionnelle

Exemple

- On considère les relations suivantes pour les exemples suivants :

Produits(prod_id, nom, pu)

prod_id	nom	pu
p1	A3	10.0
p2	crayon	9
p3	stylo	15
p4	A4	10.0

Depots(depot_id, adr, volume)

depot_id	adr	volume
1	Nancy	100
2	Laxou	200
3	Vandoeuvre	115
4	Nancy	220
5	Nancy	1000

Stock(prod_id, depot_id, qte)

prod_id	depot_id	qte
p1	1	0
p3	2	9
p1	3	15
p2	4	20
p3	5	0
p1	4	5
p2	5	2
p3	3	30
p1	2	10

SELECT ... FROM ... - 1

- Équivalent de la **Projection** en algèbre relationnelle

Produits(prod_id, nom, pu)

prod_id	nom	pu
p1	A3	10.0
p2	crayon	9
p3	stylo	15
p4	A4	10.0

SELECT nom, pu
FROM Produits



Res(nom, pu)

nom	pu
A3	10.0
crayon	9
stylo	15
A4	10.0

SELECT ... FROM ... - 2

- Seule différence, pas d'élimination des doubles sauf mention explicite avec le mot clé **DISTINCT**

Produits(prod_id, nom, pu)

prod_id	nom	pu
p1	A3	10.0
p2	crayon	9
p3	stylo	15
p4	A4	10.0

SELECT DISTINCT pu
FROM Produits



Res(pu)

pu
10.0
9
15

WHERE - 1

- Équivalent de la **Restriction** en algèbre relationnelle

Produits(prod_id, nom, pu)

prod_id	nom	pu
p1	A3	10.0
p2	crayon	9
p3	stylo	15
p4	A4	10.0

SELECT nom, pu

FROM Produits

WHERE pu < 10



Res(nom, pu)

nom	pu
crayon	9

Opérateurs pour Restriction

- **Opérateurs de comparaison :**
 - =, !=, >, >=, <, <=, LIKE
- **Connecteurs logiques :**
 - AND, OR, NOT
- **Caractères génériques :**
 - % peut se substituer à toute chaîne de caractères
 - _ peut se substituer à tout caractère

WHERE - 2

- Infos sur les crayons et stylos ?

Produits(prod_id, nom, pu)

prod_id	nom	pu
p1	A3	10.0
p2	crayon	9
p3	stylo	15
p4	A4	10.0

SELECT nom, pu

FROM Produits

WHERE nom = "crayon"

OR nom = "stylo"



Res(nom, pu)

nom	pu
crayon	9
stylo	15

WHERE - 3

- Infos des produits dont le nom commence par un "A" ?

Produits(prod_id, nom, pu)

prod_id	nom	pu
p1	A3	10.0
p2	crayon	9
p3	stylo	15
p4	A4	10.0

SELECT nom, pu

FROM Produits

WHERE nom **LIKE** "A%"

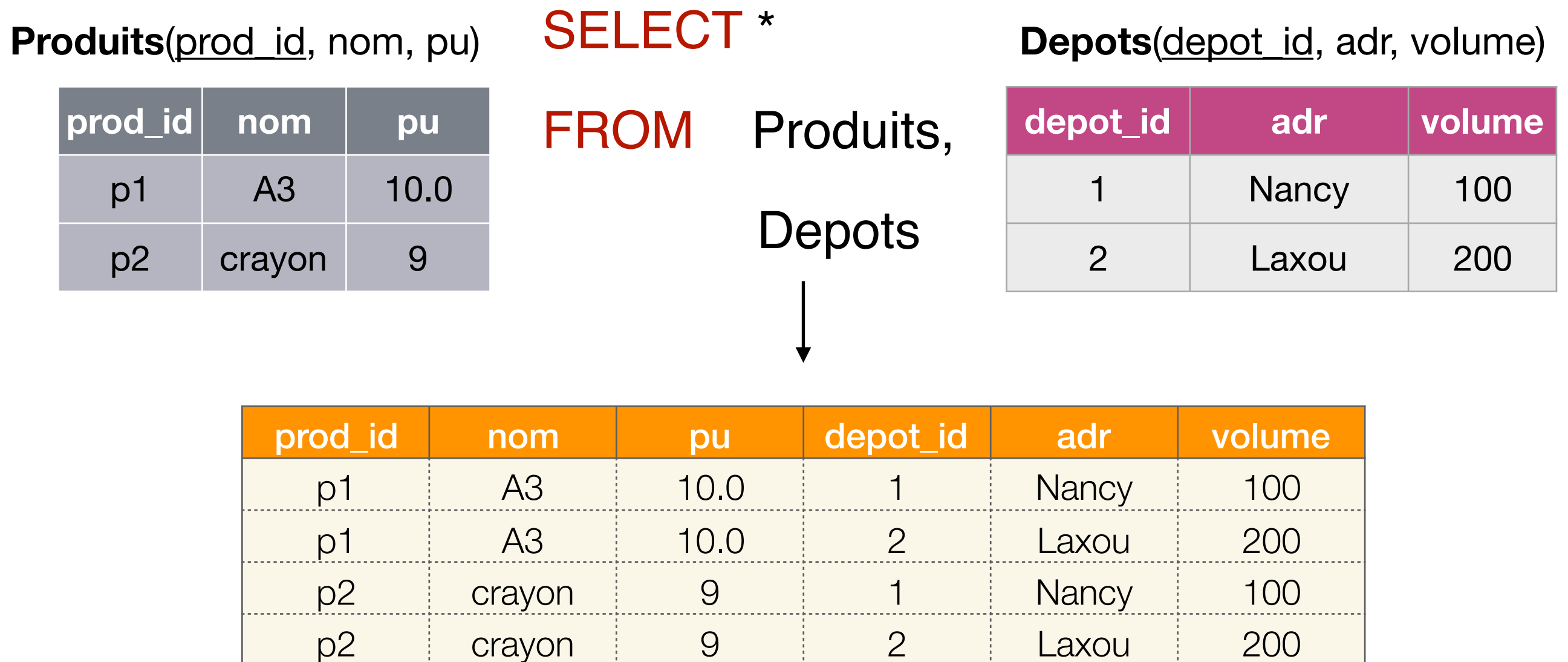


Res(nom, pu)

nom	pu
A3	10.0
A4	10.0

Produit cartésien

- Permet de sélectionner tous les attributs de la relation résultat



Jointure - 1

- S'effectue à l'aide du Produit cartésien et de la Restriction

SELECT *

FROM R1, R2

WHERE critère de jointure

Jointure - 2

- Nom des produits en stock dans le dépôt 2

```
SELECT nom
FROM   Produits, Stock
WHERE  Produits.prod_id
      = Stock.prod_id
AND    depot_id = 2
```



Res(nom)

nom
stylo
A3

Alias

- Possible d'utiliser des alias pour les relations pour simplifier l'écriture

```
SELECT nom  
FROM    Produits, Stock  
WHERE  Produits.prod_id  équivalent à  
        = Stock.prod_id  
AND    depot_id = 2
```

```
SELECT nom  
FROM    Produits p, Stock s  
WHERE  p.prod_id = s.prod_id  
AND    depot_id = 2
```

Jointure - 3

- Autre forme possible :

<code>SELECT *</code>		<code>SELECT *</code>
		<code>FROM R1</code>
<code>FROM R1, R2</code>	équivalent à	<code>WHERE attr1 IN</code>
<code>WHERE R1.attr1 = R2.attr2</code>		<code>(SELECT attr2</code>
		<code>FROM R2)</code>

- Avec *attr1* un attribut de *R1*, *attr2* de *R2*; *attr1* et *attr2* sont compatibles (de même domaine, de même type)
- Uniquement possible d'afficher des attributs de *R1*

Fonctions d'agrégation - 1

- **COUNT** : nombre de valeurs
- **SUM** : somme des valeurs
- **AVG** : moyenne des valeurs
- **MAX/MIN** : valeur maximale/minimale des valeurs

Fonctions d'agrégation - 2

- Ces fonctions retournent une seule valeur
 - et non pas un ensemble
- Peuvent être utilisées dans une clause **SELECT** ou **WHERE**
- Toutes les fonctions ignorent les valeurs NULL
 - sauf **COUNT**

Exemple COUNT

- Nombre de produits différents stockés dans le dépôt 2

Stock(prod_id, depot_id, qte)

prod_id	depot_id	qte
p1	1	0
p3	2	9
p1	3	15
p2	4	20
p3	5	0
p1	4	5
p2	5	2
p3	3	30
p1	2	10

SELECT COUNT(prod_id)

FROM Stocks

WHERE depot_id = 2



2

Outils

Base de Données 2
#1

Environnement de travail

- Utilise **MAMP** dans le cadre du module
 - Dispo ici : <https://www.mamp.info/fr>
- Boîte à outils fournissant **MySQL**, mais aussi un serveur **Apache** et l'interpréteur **PHP**
 - Des **alternatives existent** (XAMPP, uWamp)...
 - ... mais ne sont **pas aussi simples** d'utilisation
- Prévu pour le **développement d'applications web**

IDE

- Puisqu'on va écrire du SQL et plus tard du PHP, besoin d'un IDE
- Plusieurs IDEs disponibles
- Installer l'IDE de votre choix
 - Visual Studio Code, PhpStorm (payant mais devrait pouvoir récupérer une version éducation), Sublim Text, Vim...

phpMyAdmin

- Dans le cadre des premiers TDs, **manipule la BD via phpMyAdmin** (inclus dans **MAMP**)
- S'agit d'une **interface web pour créer, modifier et interroger simplement des bases de données** sur MySQL ou MariaDB

Démo

Arche

- Infos du Module
 - *2020 - Polytech 2i : Base de Données 2*
 - Identifiant : 43219
 - Clef d'inscription : P3AiiBD2