

# BDW - Bilan

Fabien Duchateau

*fabien.duchateau [at] univ-lyon1.fr*

Université Claude Bernard Lyon 1

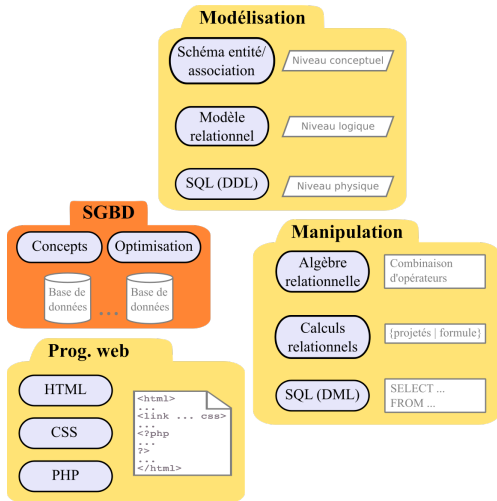
2023 - 2024



<https://perso.liris.cnrs.fr/fabien.duchateau/BDW/>

# Objectifs de BDW

- ▶ Concevoir une base de données (modélisation)
- ▶ Manipuler des données d'une base (langages AR, CR, SQL)
- ▶ Optimiser une requête
- ▶ Programmer des pages web dynamiques (langages PHP, CSS, HTML)



# Concevoir une base de données (méthode Merise)

## Diagramme entité/association (niveau conceptuel) :

- ▶ Entité, association, propriété, identifiant, cardinalité
- ▶ Entité faible, association réflexive, spécialisation, agrégation

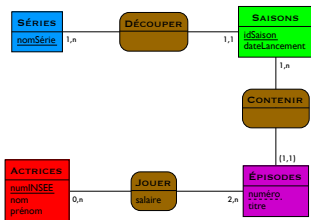
## Modèle relationnel (niveau logique) :

- ▶ Relation, attribut, clés primaire et étrangère
- ▶ Trois règles de transformation E/A vers relationnel
  - ▶ entité  $\Rightarrow$  relation
  - ▶ association avec cardinalité maximale en 1  $\Rightarrow$  clé étrangère
  - ▶ association sans cardinalité maximale en 1  $\Rightarrow$  relation dont la clé primaire se compose des identifiants des entités porteuses

## Concevoir une base de données (2)

### Langage de définition SQL (niveau physique) :

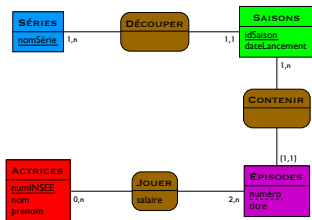
- ▶ Création, modification, suppression, renommage pour les tables, colonnes, contraintes
- ▶ Insertion, mise à jour, suppression pour les n-uplets



## Concevoir une base de données (2)

### Langage de définition SQL (niveau physique) :

- ▶ Création, modification, suppression, renommage pour les tables, colonnes, contraintes
- ▶ Insertion, mise à jour, suppression pour les n-uplets



SÉRIES (nomSérie)  
SAISONS (idSaison, dateLancement, #nomSérie)  
ÉPIISODES (numero, #idSaison, titre)  
ACTRICES (numINSEE, nom, prénom)  
JOUER (#numero, #idSaison, #numINSEE, salaire)

**Question 1** : pourquoi la clé primaire de la relation EPISODES est composée des deux attributs *numero* et *idSaison* ?

# Manipuler une base de données

## Algèbre relationnelle :

- ▶ Combinaisons d'opérateurs ("comment obtenir le résultat")
- ▶  $\Pi_{\text{nom, prénom}}(\sigma_{\text{salaire} > 5000}(\text{JOUER} \bowtie_{\text{numINSEE}} \text{ACTRICES}))$

## Calcul relationnel (à variable tuple) :

- ▶ Logique des prédicats ("décrire le résultat escompté")
- ▶  $\{a.\text{nom}, a.\text{prénom} \mid \text{ACTRICES}(a) \wedge \exists j (\text{JOUER}(j) \wedge j.\text{numINSEE}=a.\text{numINSEE} \wedge j.\text{salaire} > 5000)\}$

# Manipuler une base de données

## Algèbre relationnelle :

- ▶ Combinaisons d'opérateurs ("comment obtenir le résultat")
- ▶  $\Pi_{\text{nom, prénom}}(\sigma_{\text{salaire} > 5000}(\text{JOUER} \bowtie_{\text{numINSEE}} \text{ACTRICES}))$

## Calcul relationnel (à variable tuple) :

- ▶ Logique des prédicats ("décrire le résultat escompté")
- ▶  $\{a.\text{nom}, a.\text{prénom} \mid \text{ACTRICES}(a) \wedge \exists j (\text{JOUER}(j) \wedge j.\text{numINSEE}=a.\text{numINSEE} \wedge j.\text{salaire} > 5000)\}$

**Question 2** : en SQL, quel mot-clé faut-il ajouter pour obtenir le même résultat que les requêtes AR et CRVT ?

**SELECT** nom, prénom **FROM** Jouer **NATURAL JOIN** Actrices  
**WHERE** salaire > 5000 ;

## Manipuler une base de données (2)

**Langage de manipulation SQL (basé sur le CRVT) :**

```
SELECT att1 [, att2 [ AS att'2 ], ... ]  
FROM nom_table1 [, nom_table2 [ alias ]]  
[ WHERE condition ]  
[ GROUP BY attk, ... ]  
[ HAVING condition_groupe ]  
[ ORDER BY atti, ... ] ;
```



## Manipuler une base de données (2)

**Langage de manipulation SQL (basé sur le CRVT) :**

```
SELECT att1 [, att2 [ AS att'2 ], ... ]  
FROM nom_table1 [, nom_table2 [ alias ]]  
[ WHERE condition ]  
[ GROUP BY attk, ... ]  
[ HAVING condition_groupe ]  
[ ORDER BY atti, ... ] ;
```

**Question 3 :** traduire la requête suivante en SQL : "pour chaque actrice, donner la moyenne de ses salaires, le tout trié par nom"

```
ACTRICES(numINSEE, nom, prénom)  
JOUER(numéro, idSaison, numINSEE, salaire)
```

# Programmer des pages dynamiques

## Langage HTML :

- ▶ Affichage des données, du contenu
- ▶ Balises, attributs, formulaires
- ▶ Organisation des pages avec les balises structurantes

```
<HTML>
  <HEAD>
    <TITLE></TITLE>
  </HEAD>
  <BODY>

  </BODY>
</HTML>
```

## Langage CSS :

- ▶ Mise en page et mise en forme
- ▶ Feuille de style externe, sélecteurs, propriétés, valeurs
- ▶ Ordre du flux, choix du type de boîte, dimensions, flex, et positionnement pour placer un élément dans une page

```
.ninja {
  color: #000000;
  visibility: hidden;
}
```

# Programmer des pages dynamiques (2)

## Langage PHP :

- ▶ Traitements plus complexes (programmation)
- ▶ Variables, structures de contrôle, boucles
- ▶ Inclusion (factorisation du code) et architecture MVC
- ▶ Superglobales (dont session)
- ▶ Bases de données MySQL / MariaDB :
  - ▶ connexion et déconnexion
  - ▶ envoi d'une requête
  - ▶ récupération et utilisation du résultat
  - ▶ requêtes préparées (réutilisation)

## Programmer des pages dynamiques (3)

numero	idSaison	numINSEE	salaire
1	3	111	NULL
1	3	222	6000
2	3	111	5437.65
2	3	222	6000

Table JOUER

numINSEE	nom	prenom
111	Bean	Sean
222	Fairley	Michelle
333	Cuoco	Kaley
444	Parsons	Jim

Table ACTRICES

**Question 4** : qu'affiche ce script PHP ?

```
<?php
// on possède un lien de connexion $connexion
$req = "SELECT nom, prenom FROM Actrices NATURAL JOIN
      Jouer WHERE salaire > 5000 ORDER BY nom ";
$resultat = mysqli_query($connexion, $req) ;
if($resultat != FALSE) {
    while ($ligne = mysqli_fetch_array($resultat))
        echo $ligne[1] . ", " . $ligne[0] . "<br>";
}
?>
```

# Optimiser une requête

## Analyse, compilation, optimisation et exécution :

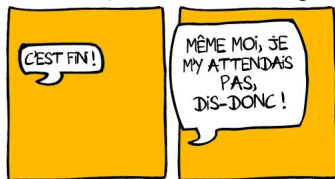
- ▶ Génération de plans d'exécution équivalents, optimisation des plans et sélection du plan optimal
- ▶ Optimisation par manipulation algébrique (i.e., réduire le nombre de n-uplets et le nombre de colonnes avant de réaliser les opérations coûteuses)
- ▶ Optimisation par choix d'un algorithme adapté

## Question 5 : parmi ces requêtes, quelle est la plus optimisée ?

1.  $\Pi_{nom,prenom}(\sigma_{salaire>1000}(\text{JOUÉ} \bowtie_{numINSEE} \text{ACTRICES}))$
2.  $\Pi_{nom,prenom}(\Pi_{numINSEE}(\sigma_{salaire>1000}(\text{JOUÉ}))) \bowtie_{numINSEE} (\text{ACTRICES})$

# Pour aller plus loin sur les SGBD

- ▶ Modélisation avancée (formes normales, dépendances fonctionnelles)
- ▶ Programmation de contraintes (langage PL/SQL)
- ▶ Nouveaux modèles de SGBD (post-relationnels)
- ▶ SGBD distribués, entrepôts, science des données, etc.



OH MON DIEU ! NON !!! MAIS QU'ALLONS NOUS FAIRE  
EN ATTENDANT ?!!! DEMANDE, ANXEUX, LE LECTEUR  
ABASOURDI PAR CETTE NOUVELLE ÉPOUSTOULANTE.

<http://www.labandepasdessinee.com/>