

# Méthodes et vues

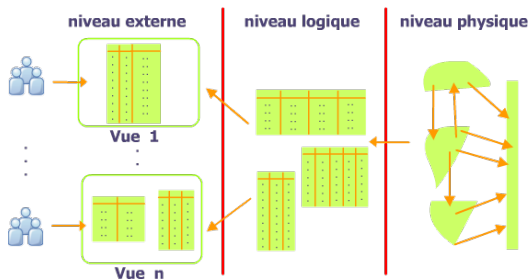
Jan Van den Bussche (Limburg)

–

Emmanuel Waller (Orsay)

BDA 98

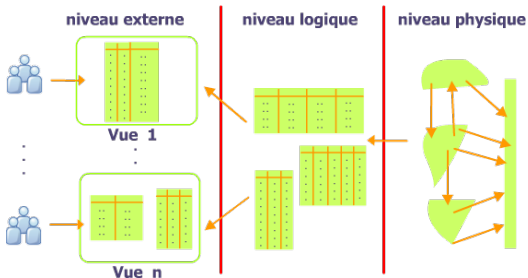
# Les bases de données et les vues



Base de données :

- ▶ Stockage d'une masse d'informations.
- ▶ Optimisation de la gestion (ajout, interrogation et mise à jour).

# Les bases de données et les vues



Vue : table virtuelle définie par une requête.

- ▶ Indépendance des niveaux
- ▶ Confidentialité des données
- ▶ Confort et optimisation des requêtes

Introduction

Le problème par  
l'exemple

Formalisation du  
problème

Travaux reliés

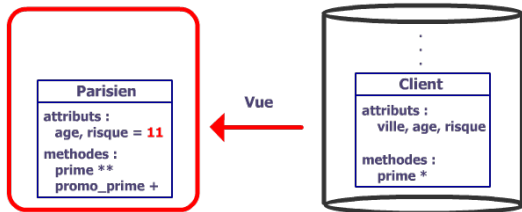
Mise en œuvre

Conclusion

# ... dans le modèle orienté objet

Les données sont représentées sous forme d'objets contenant

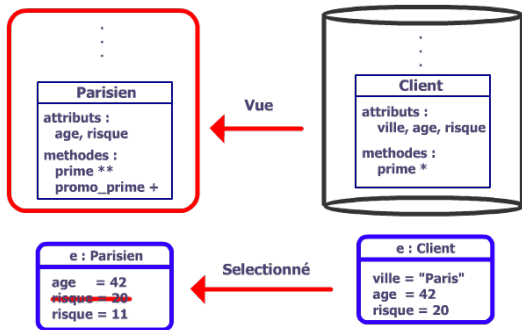
- ▶ Attributs
- ▶ Méthodes (héritage, redéfinition, liaison dynamique)
- ▶ ...



Base de données d'une compagnie d'assurance

- \* :  $prime = risque + age$
- \*\* :  $prime = risque \times age$
- + :  $promo\_prime = \frac{prime}{2}$

# Le contexte de l'exemple



Base de données d'une compagnie d'assurance

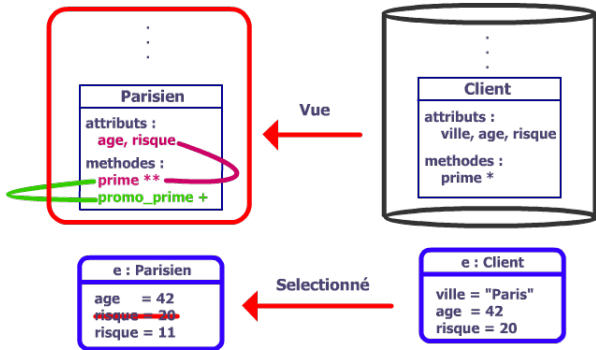
Calcul de la prime d'un parisien (*promo\_prime*)

$$promo\_prime(e) = \frac{prime(e)}{2}$$

Comment calculer *prime(e)* ?

# Trois choix d'exécution

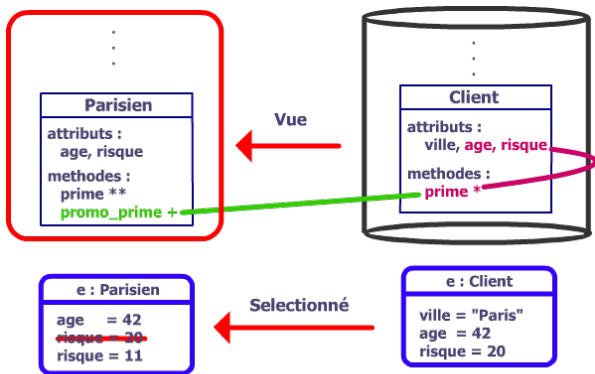
## Premier cas : Calcul dans la vue



Base de données d'une compagnie d'assurance

# Trois choix d'exécution

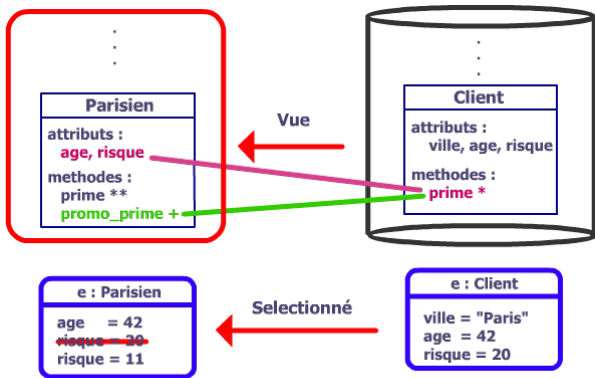
## Second cas : Calcul dans la base racine



Base de données d'une compagnie d'assurance



**Troisième cas :** Calcul dans la vue avec le code de la base racine



Base de données d'une compagnie d'assurance

## Introduction

### Le problème par l'exemple

### Formalisation du problème

### Travaux reliés

### Mise en œuvre

### Conclusion

- ▶ Permettre au développeur de choisir le mode d'exécution :
  - ▶ calcul dans la vue
  - ▶ calcul dans la base racine
  - ▶ calcul dans la vue avec le code de la base racine
- ▶ Introduire un modèle abstrait de vues comportementales.
- ▶ ...

# Ingrédients

## Définition

*Vue ordinaire* =  
*classes*  
+ *attributs*

## Définition

*Vue comportementale* =  
*classes*  
+ *attributs*  
+ *comportement (methodes codees)*

# Travaux reliés

## O<sub>2</sub> Views :

1. thèse Souza
2. Souza, Abiteboul, Delobel, EDBT 94

(Programmation E. Waller : appli pour revue Esprit)

- ▶ aspect structurel.
- ▶ méthodes : appelées par les requêtes lors du calcul, attributs virtuels.
- ▶ sémantique : tourne parfaitement mais implicite, ad hoc, incofortable.

## A formal model of views : Guerrini et al, TAPOS 97

- ▶ aspect structurel.
- ▶ méthodes : possible de disposer dans la vue d'un code de la base racine.
- ▶ sémantique non fournie.

# L'idée (esquisse)

Introduction de deux primitives permettant de spécifier le comportement de la vue.

- ▶ `import value.`
- ▶ `import code.`

# L'importation de valeur

Il s'agit d'effectuer le calcul dans l'environnement de la base racine et de renvoyer le résultat (valeur) à la vue.

*promo\_prime@Parisien = import value prime*

**Correspond au calcul dans la base racine**

# L'importation de code

Il s'agit d'importer le code d'une méthode se trouvant dans la base racine. Deux cas de figure peuvent se distinguer :

```
promo_prime@Parisien = import code prime
```

**Correspond au calcul dans la vue (valeurs de la vue)  
avec code de la base racine**

```
promo_prime@Parisien = import code prime  
with(risque : import value)
```

**Correspond au calcul dans la vue avec code de la base  
racine + spécification de l'importation de certaines  
valeurs de la base racine**

# Spécification du comportement d'une vue

## Définition

*Vue comportementale* =  
    *classes*  
    + *attributs*  
    + *comportement (methodes codees)*

## Et donc...

*Comportement d'une vue* =  
    *specification globale d'import de code*  
    + *definition des methodes codees*

- ▶  $m@c = \text{code}$
- ▶  $m@c = \text{import value } m'$
- ▶  $m@c = \text{import code } m'$



# En bref...

## Résumé

- ▶ Introduction d'un modèle abstrait permettant de spécifier le mode d'exécution des appels de méthodes.
  - ▶ calcul dans la vue
  - ▶ calcul dans la base racine
  - ▶ calcul dans la vue avec le code de la base racine
- ▶ Correction (voir papier).

## Perspectives

- ▶ Variantes de import code et import value (Ex : surcharge de import value, ie. tenter de charger une méthode dans vue, si échec, l'importer de la base).
- ▶ masquage de certaines parties de la vue (Ex : méthodes auxiliaires).