

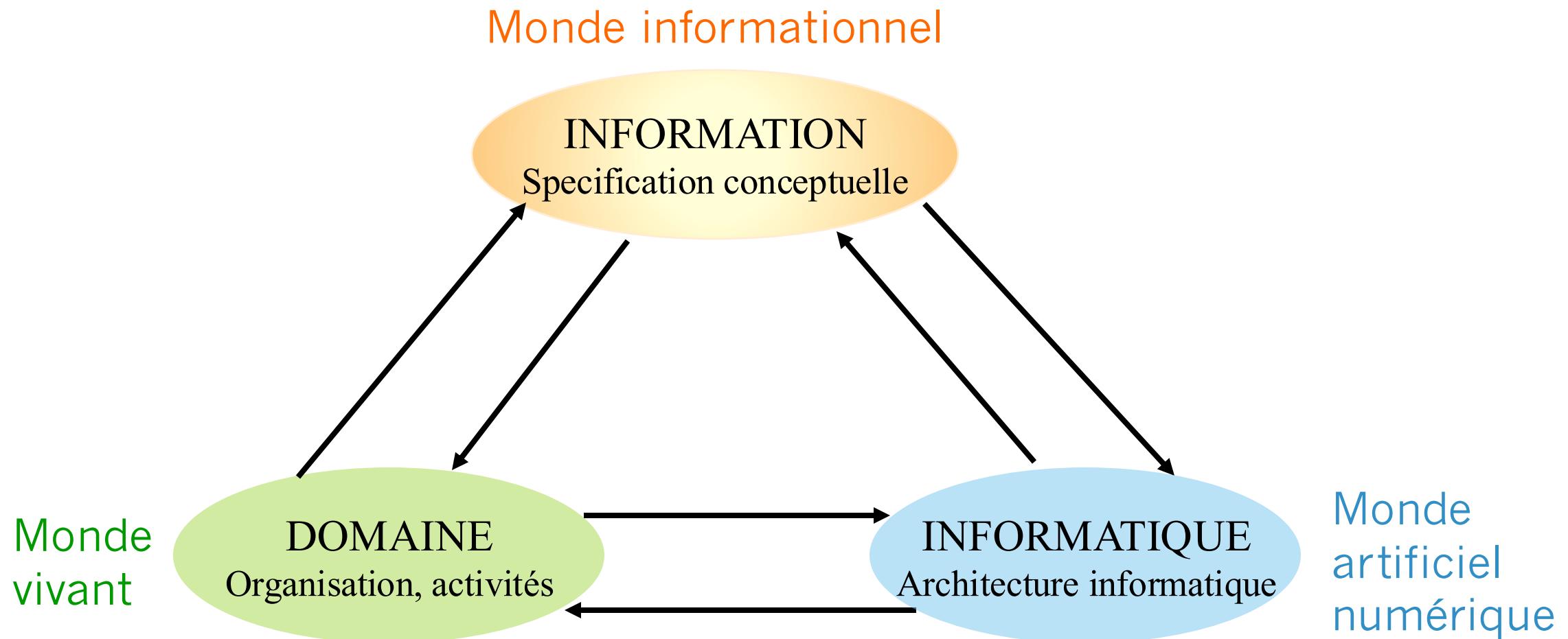
# MSIS Chapitre 6

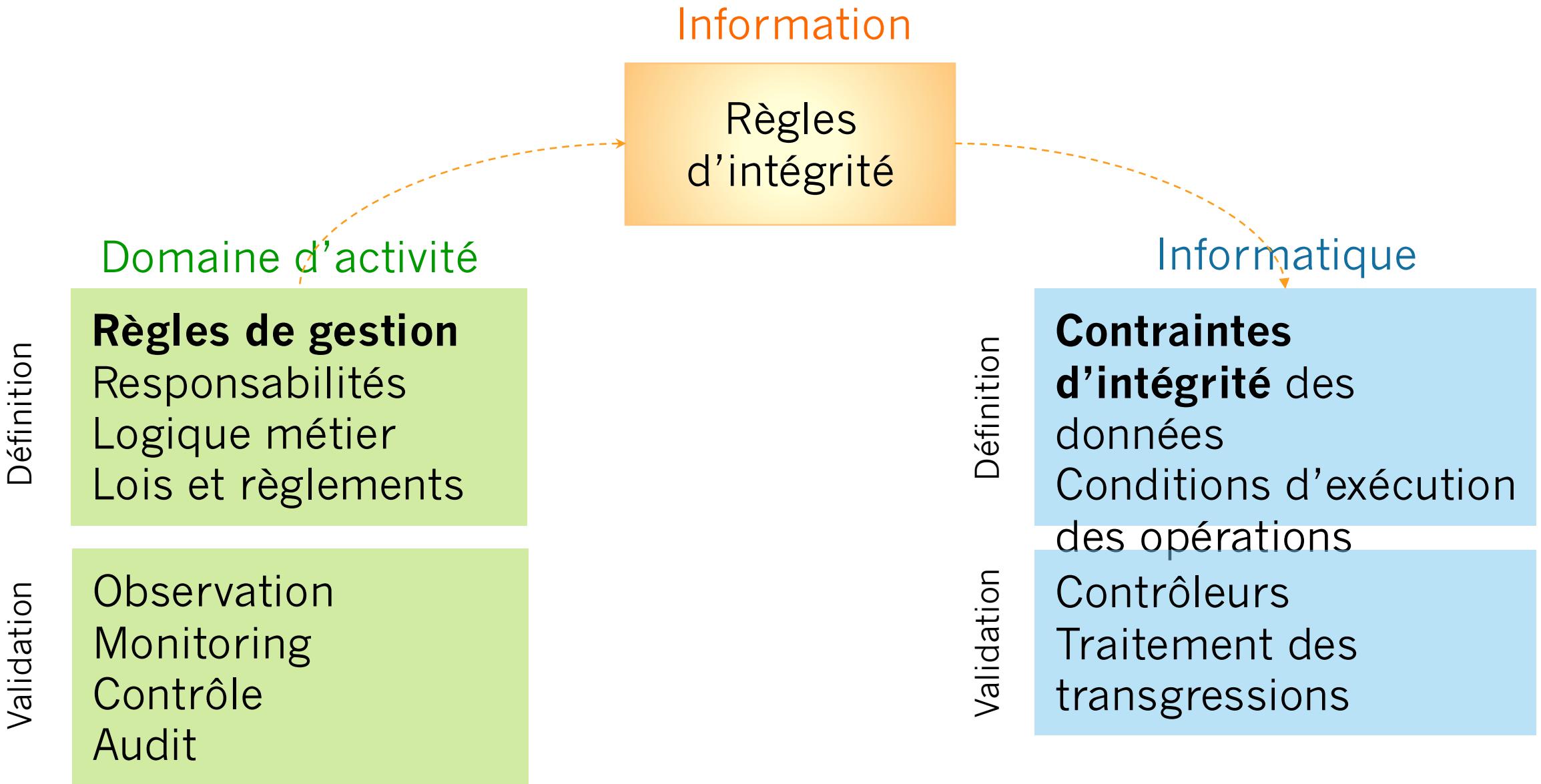
## Règles d'intégrité

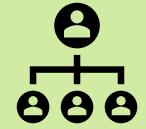
# Dans ce chapitre

- Règles de gestion vs règles d'intégrité
- Spécification des règles d'intégrité
- Typologie des règles d'intégrité
- Tableaux de portée des règles d'intégrité
- Exercice

# Spécification des règles







## Règle de gestion

Un concept métier qui sert à coordonner les activités de différents services et de différents acteurs de l'organisation.

Son objectif est d'assurer la cohérence logique des processus métier et leur conformité aux lois qui régissent le secteur d'activité.



## Contrainte d'intégrité

Un concept informatique qui désigne une condition logique qui doit toujours être vérifiée.

Son objectif est d'assurer la cohérence logique des données stockées dans une base de données.



## Règle d'intégrité (RI)

dans la modélisation conceptuelle de données

La conception d'une base de données comprend non seulement un modèle conceptuel de données (diagramme de classes, schéma relationnel) mais aussi un ensemble de règles d'intégrité.

**Règle d'intégrité (RI)** – désigne une condition logique qui est

- définie par rapport à une ou plusieurs classes,
- vérifiée pour tout état du SI ou pour tout changement d'état du SI,
- validée de manière algorithmique sur les objets des classes concernées.

# Une règle d'intégrité peut contrôler

- La cohérence de données lors de leur saisie

*La date d'obtention d'un certificat doit être postérieure à la date de naissance de la personne.*

- La cohérence de données liée à une règle de gestion

*Le stock d'un produit commandé doit être supérieur au nombre d'exemplaires commandés.*

- Une règle de gestion

*Toute personne pour être affectée à un projet doit avoir au moins deux compétences requises pour participer à ce projet.*

# Implémentation de règles d'intégrité

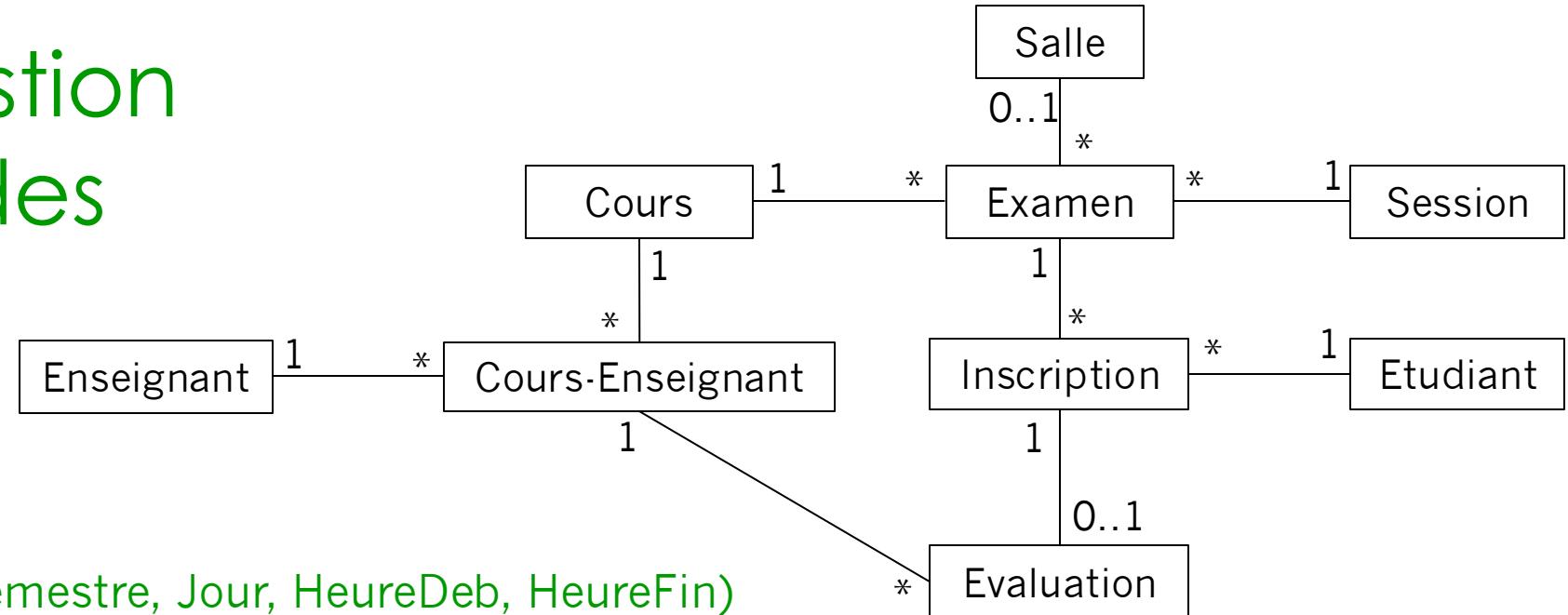
Implémentation de règles d'intégrité nécessite :

- la mise au point des **contrôleurs**,
- la détermination des situations qui doivent les déclencher,
- la conception de leur réponse à tout cas de transgression.

Les contrôleurs sont des programmes particuliers :

- ils se déclenchent pour contrôler tout traitement susceptible de transgresser la règle,
- dans le cas de transgression, ils arrêtent, suspendent ou remplacent les traitements fautifs.

# Exemple : Gestion des cours et des examens



**Cours** (NoCours // Titre, Semestre, Jour, HeureDeb, HeureFin)

**Enseignant** (NoEns // NomEns, PrénomEns)

**Cours-Enseignant** (NoCours, NoEns //)

**Session** (NomSession // DateDebSes, DateFinSes, DateLimitInscription)

**Salle** (NomSalle // Adresse, NbPlaces)

**Examen** (NoCours, NomSession // DateExam, NomSalle, HeureDebExam, Durée)

**Etudiant** (NoEtu // NomEtu, PrénomEtu, e-mailEtu)

**Inscription** (NoCours, NomSession, NoEtu // DateInscription)

**Evaluation** (NoCours, NomSession, NoEtu // NoEns, Note, DateNote)

# Exemples des RI

*DuréeSession* : Une session d'examens doit durer minimum trois semaines.

*CoursEns* : Un enseignant ne peut pas avoir plus de 3 cours par semestre.

*DateExam* : La date d'un examen doit être dans la période de la session concernée.

*+Note* : Si on modifie une note, elle ne peut qu'augmenter.

*+MoyNote* : Si on modifie une note, la nouvelle note doit être supérieure ou égale à la moyenne de toutes les notes du même examen.

# Spécification des règles d'intégrité

Contexte

Portée

Réponse

Expression formelle

# Contexte d'une RI

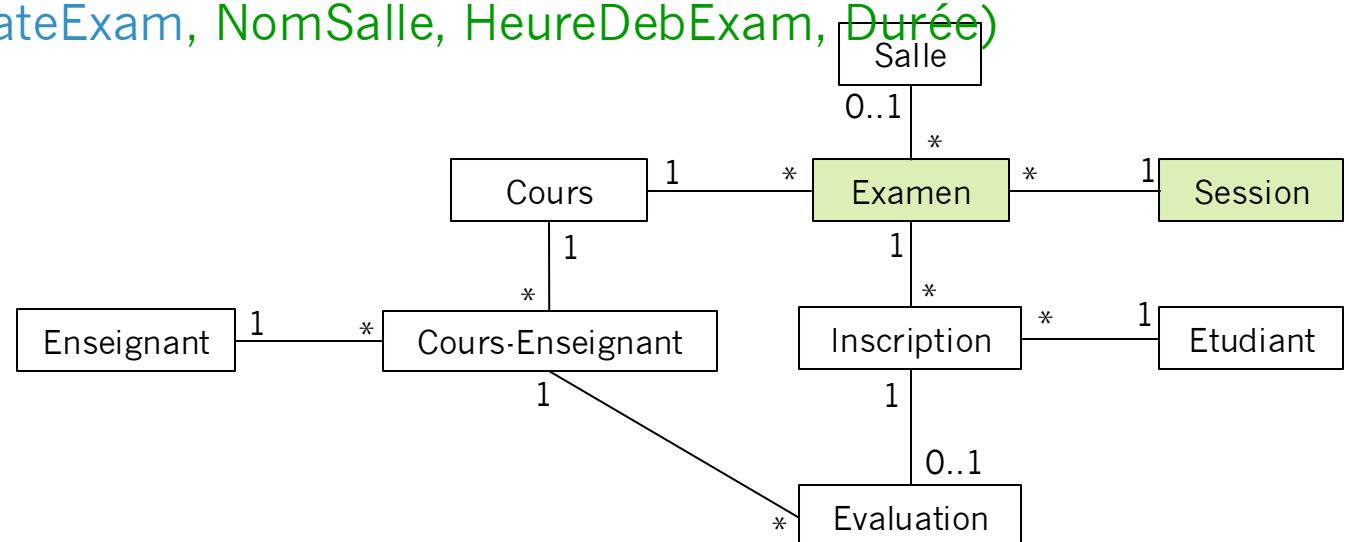
L'ensemble de classes concernées par la définition et la validation de la règle d'intégrité.

RI : *DateExam* : La date d'un examen doit être dans la période de la session concernée.

Contexte (classes) :

Session (NomSession // DateDebSes, DateFinSes, DateLimitInscription)

Examen (NoCours, NomSession // DateExam, NomSalle, HeureDebExam, Durée)



# Portée d'une RI

L'ensemble des primitives de modification de classes, qui doivent valider la RI, pour conserver la cohérence du SI.

Primitives: créer, supprimer, modifier

Tout état du SI validant la RI doit être transformé par cette primitive en un autre état validant la RI.

Exemple : la portée de la RI *DateExam* :

- Créer Examen
- Modifier Session[DateDebSes]
- Modifier Session[DateFinSes]
- Modifier Examen[DateExam]

# Réponse d'une RI

Les actions à entreprendre en cas de transgression (non-validation) de la RI.

Le refus de l'exécution de la primitive, et, par conséquence, du traitement qui avait besoin de l'exécution de cette primitive.

Exemple : la réponse de la RI *DateExam* :

- Refuser Créer Examen
- Refuser Modifier Session[DateDebSes]
- Refuser Modifier Session[DateFinSes]
- Refuser Modifier Examen[DateExam]

# Expression d'une RI

Informelle : décrit la règle d'intégrité en langue naturelle, grâce aux prédictats des classes du contexte de la règle d'intégrité.

Exemple d'expression en langage naturel de la RI *DateExam* :

*La date d'un examen doit être dans la période de la session concernée.*

Formelle : une formulation rigoureuse et sans ambiguïté en utilisant un langage formel :

- une expression prédicative sur des classes du contexte de la RI (langage relationnel)
- une expression algébrique
- une expression algorithmique
- une expression mathématique

# Exemples d'expression formelle des RI

RI *DateExam* :

Expression en langage algébrique :

$$\begin{aligned} & \forall e \in \text{Examen}, s \in \text{Session} \mid e.\text{NomSession} = s.\text{NomSession} \\ \Rightarrow & (e.\text{DateExam} \geq s.\text{DateDebSes} \wedge e.\text{DateExam} \leq s.\text{DateFinSes}) \end{aligned}$$

Expression en langage relationnel :

$$\begin{aligned} & \forall es \in (\text{Examen} * \text{Session}) : es[\text{DateExam}] \geq es[\text{DateDebSes}] \text{ et } es[\text{DateExam}] \\ & \leq es[\text{DateFinSes}] \end{aligned}$$

# Typologie des RI

RI individuelle vs RI ensembliste

RI statique vs RI dynamique

## RI individuelle

Une règle d'intégrité qui porte sur un seul objet d'une seule classe.

Exemple RI

*DuréeSession* : Une session d'examens doit durer minimum trois semaines.

La règle doit être validé au moment de la création et modification d'un objet de la classe :

Session (NomSession // DateDebSes,  
DateFinSes, DateLimiteInscription)

## RI ensembliste

Une règle d'intégrité qui porte sur plusieurs objets d'une ou plusieurs classes.

Exemple RI

*CoursEns* : Un enseignant ne peut pas avoir plus de 3 cours par semestre.

La validation de la règle porte sur les classes :

Cours (NoCours // Titre, Semestre, Jour,  
HeureDeb, HeureFin)

Cours-Enseignant (NoCours, NoEns //)

## RI statique

Une règle d'intégrité qui s'applique à n'importe quel état de la base de données.

- À tout moment il est possible de constater si la base de données valide la règle d'intégrité ou non.
- Si un état de la base de données ne valide pas la règle d'intégrité, cela veut dire qu'il existe un traitement où la validation de la règle se fait de manière erronée.

## RI dynamique

Une règle d'intégrité qui s'applique aux changements d'états de la base de données.

- Elle exprime une condition logique entre l'état de la base avant la modification et l'état de la base après la modification.
- Sa validation se fait obligatoirement pendant l'exécution d'un changement d'états – durant l'opération de modification.
- Il est impossible de détecter une transgression simplement en observant un état de la base de données.

# Exemple : RI statique individuelle

RI *DuréeSession* : Une session d'examens doit durer minimum trois semaines

Expression formelle :

$$\forall s \in \text{Session} : s[\text{DateFinSes}] - s[\text{DateDebSes}] \geq 21 \text{ jours.}$$

Contexte :

Session (NomSession // DateDebSes, DateFinSes, DateLimiteInscription)

Portée :

- Créer Session
- Modifier Session[DateDebSes]
- Modifier Session[DateFinSes]

# Exemple : RI statique ensembliste

RI *CoursEns* : Un enseignant ne peut pas avoir plus de 3 cours par semestre.

Expression formelle :

Si  $ce \in \text{Cours-Enseignant}$  et  $ce[\text{NoEns}] = noens$ ,  $c \in \text{Cours}$  et  $c[\text{Semestre}] = sem$  :  
cardinalité  $((\text{Cours} * \text{Cours-Enseignant})[\text{Semestre} = sem, \text{NoEns} = noens]) \leq 3$ .

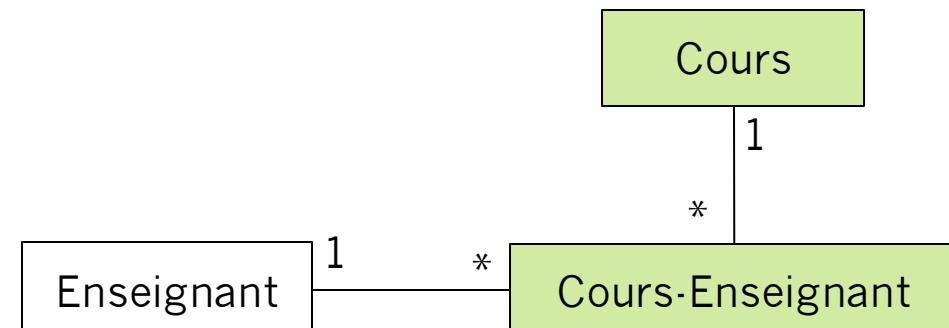
Contexte :

*Cours* (NoCours // Titre, Semestre, Jour, HeureDeb, HeureFin)

*Cours-Enseignant* (NoCours, NoEns //)

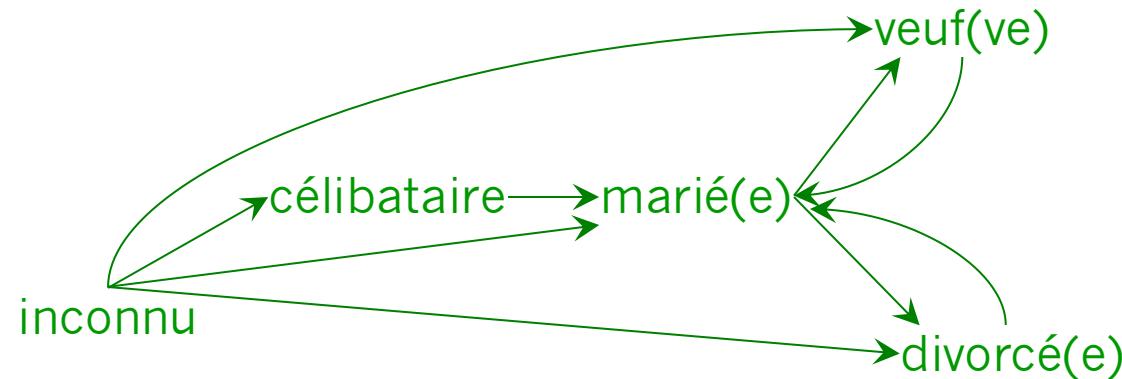
Portée :

- Créer *Cours-Enseignant*
- Modifier *Cours*[Semestre]



# Exemple : RI dynamique individuelle

RI : Les modifications possibles de l'état civil d'une personne :



Contexte :

Personne (NoPers // Nom, Prénom, DateNais, EtatCivil)

Portée :

Modifier Personne[EtatCivil]

# Exemple : RI dynamique individuelle

RI +*Note* : Si on modifie une note, elle ne peut qu'augmenter.

Expression formelle :

Si  $e \in \text{Evaluation}$  et  $e[\text{Note}] = n$ ,  $\forall$  nouvelle note  $n'$   $e[\text{Note}] = n' : n' > n$ .

Contexte :

`Evaluation (NoCours, NomSession, NoEtu // NoEns, Note, DateNote)`

Portée :

`Modifier Evaluation[Note]`

# Exemple : RI dynamique ensembliste

RI +*MoyNote* : Si on modifie une note, la nouvelle note doit être supérieure ou égale à la moyenne de toutes les notes du même examen.

Expression formelle :

Si  $e \in \text{Evaluation}$ ,  $e[\text{Note}] = n$  et  $e[\text{NoCours}] = nc$ ,

$\forall$  nouvelle note  $n'$   $e[\text{Note}] = n' : n' > \text{moyenne}(\text{Evaluation}[\text{NoCours} = nc][\text{Note}])$

Contexte :

$\text{Evaluation}(\text{NoCours}, \text{NomSession}, \text{NoEtu} // \text{NoEns}, \text{Note}, \text{DateNote})$

Portée :

Modifier  $\text{Evaluation}[\text{Note}]$

# Tableaux de portée des RI

## Spécification des RI

# Tableau de portée des RI

Tableau de portée des RI – permet de spécifier l'ensemble de primitives de modification de classes, qui doivent valider les règles d'intégrité :

- **Créer** – le risque encouru lors de la création d'un objet
- **Supprimer** – le risque encouru lors de la suppression d'un objet
- **Modifier attribut** – le risque encouru lors de la mise à jour de la valeur prise par un objet pour un attribut

# Exemple : tableau de portée des RI

Classe	RI	DateExam	DuréeSession	CoursEns	+Note	+MoyNote
Cours				Maj Semestre		
Enseignant						
Cours-Enseignant				Créer		
Session	Maj DateDebSes Maj DateFinSes		Créer Maj DateDebSes Maj DateFinSes			
Examen		Créer Maj DateExam				
Evaluation					Maj Note	Maj Note

Remarque : Ce tableau se remplit en considérant que sont vérifiées toutes les dépendances référentielles et existentielles, ainsi que les règles concernant les identifiants primaires des classes et les domaines d'attributs.

# Exemple : tableau de portée d'une RI

RI : *DateExam* : La date d'un examen doit être dans la période de la session concernée.

Contexte

Session (NomSession // DateDebSes, DateFinSes, DateLimitInscription)

Examen (NoCours, NomSession // DateExam, NomSalle, HeureDebExam, Durée)

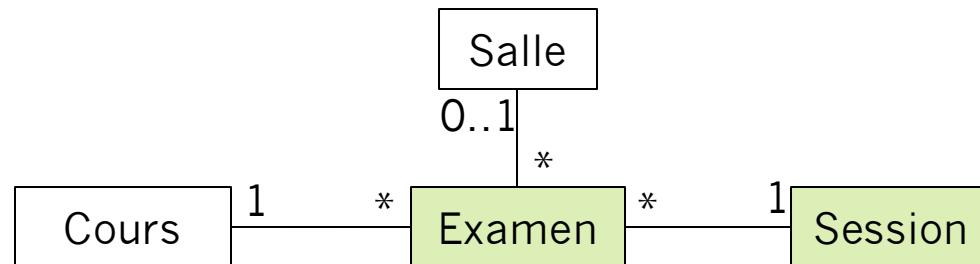


Tableau de portée

Classe	Créer	Supprimer	Modifier
Session	Non	Non	DateDebSes DateFinSes
Examen	Oui	Non	Maj DateExam

# Exemple : tableau de portée d'une RI

RI : *DuréeSession* : Une session d'examens doit durer minimum trois semaines.

Contexte

Session (NomSession // DateDebSes, DateFinSes, DateLimitInscription)

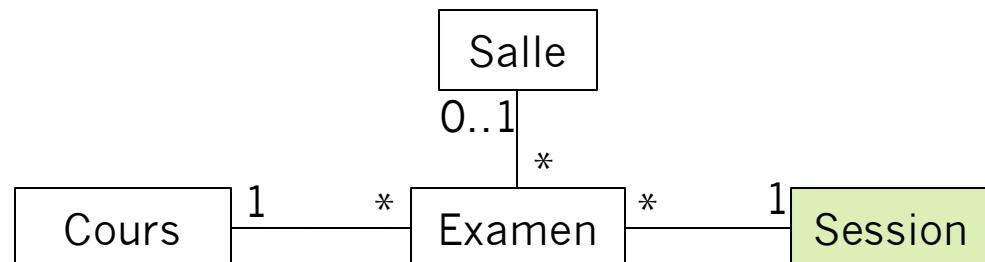


Tableau de portée

Classe	Créer	Supprimer	Modifier
Session	Oui	Non	DateDebSes DateFinSes

# Exemple : tableau de portée d'une RI

RI : *CoursEns* : Un enseignant ne peut pas avoir plus de 3 cours par semestre.

Contexte

Cours (NoCours // Titre, Semestre, Jour, HeureDeb, HeureFin)

Cours-Enseignant (NoCours, NoEns //)

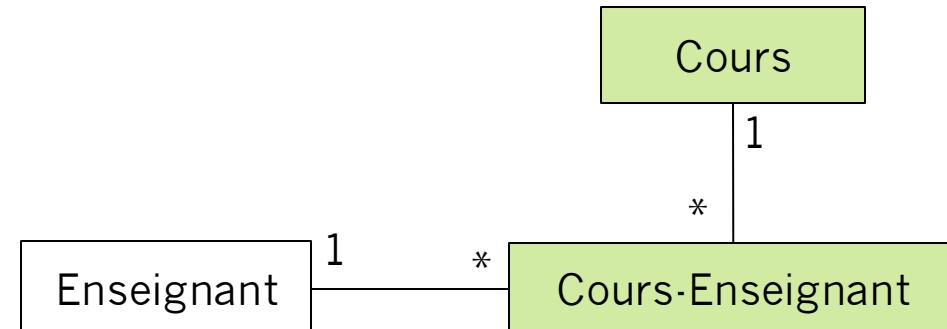


Tableau de portée

Classe	Créer	Supprimer	Modifier
Cours	Non	Non	Semestre
Cours-Enseignant	Oui	Non	-

# Exemple : tableau de portée d'une RI

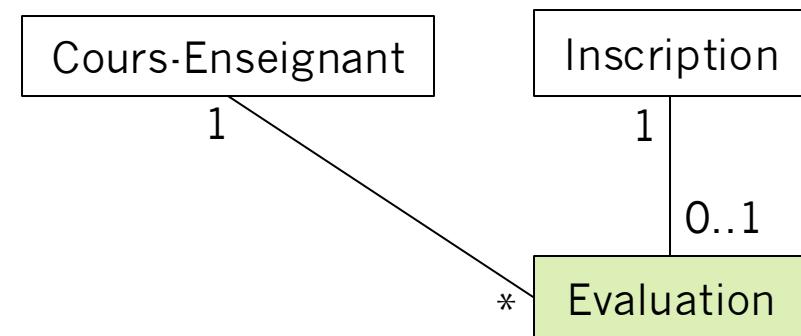
RI : RI : +Note : Si on modifie une note, elle ne peut qu'augmenter.

Contexte

Evaluation (NoCours, NomSession, NoEtu // NoEns, Note, DateNote)

Tableau de portée

Classe	Créer	Supprimer	Modifier
Evaluation	Non	Non	Note



# Défis d'implémentation des RI

La précision dans la spécification des RI

Langage naturel vs langage formel

Le choix des RI à implémenter

Règle de gestion vs Règle d'intégrité

Le coût (temps et ressources)

Développement

- Spécification: expression, portée, contrôleurs, réponses aux transgressions
- Implémentation et tests

Exécution des traitements