

BUT Informatique
1A - Semestre 1
Introduction aux bases de données
(R1.05)

R. Fleurquin

Chapitre 1

Le paradigme relationnel

Le support de cours « littéraire » est dans le sujet de TD1!

Le paradigme *relationnel*
a plus de 50 ans et
il est « bien fondé »

=>

Théorie des ensembles \cup Logique des prédicats

L'information est stockée dans des *Relations* ou *Tables*

Attribut (colonne)

Domaine (type)

Étudiant

numEtudiant (int)	nomEtudiant (String)	prénomEtudiant (String)	dateNaissance (Date)	bacEtudiant (String)	formationEtudiant (String)
11	Menez	Matthys	17/11/2004	STI2D	BUT INFO
8	Leborgne	Néo	30/01/2004	Général	BUT INFO
114	Veran	Olivier	NULL	STI2D	BUT STID

tuple (n-uplet, enregistrement)

Contrainte du paradigme : **toutes les lignes d'une relation doivent être deux à deux distinctes !**

Des « clés » pour repérer l'information : au moins une *clé candidate* dont une *clé primaire*

Un sous-ensemble d'attributs d'une relation est dit « *clé candidate* » lorsque :

1. *Propriété d'unicité atemporelle* : la valeur sur cet ensemble d'attribut est et sera toujours différente pour chaque n-uplet d'une relation
2. *Propriété de minimalité atemporelle* : il n'est pas possible de retirer l'un des attributs de la clé sans perdre la propriété d'unicité précédente.

Parmi les clés candidates (au moins une!), une clé est privilégiée : la « *clé primaire* ».

Étudiant

numEtudiant (int)	nomEtudiant (String)	bacEtudiant (String)	formationEtudiant (String)
11	Le moal	STI2D	BUT INFO
8	Le moal	Général	BUT STID
114	Veran	STI2D	BUT STID

Clé primaire
(donc candidate!)

Clés candidates?

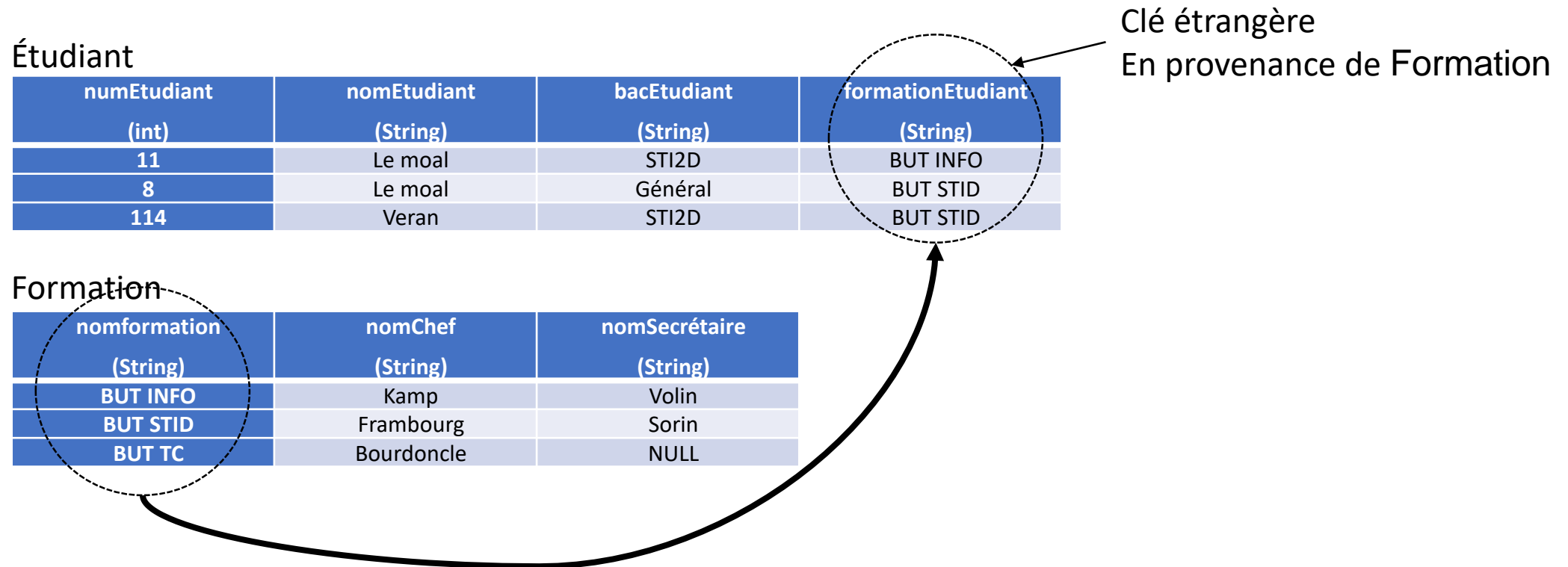
Non clé car pas d'unicité

Non clé car non minimale

Attention aux clés candidates de « circonstance » qui ne sont pas un invariant temporel dans l'univers modélisé.

Les clés étrangères pour « recoller » les morceaux!

L'importation d'une clé primaire d'une table dans une autre table s'appelle une « *clé étrangère* ».



L'attribut `formationEtudiant` de la relation `Etudiants` étant une clé étrangère, son domaine de valeurs doit nécessairement être inclus dans celui des valeurs actuellement présentes dans la relation `Formations`.

On parle de « *contrainte d'intégrité référentielle* ».

Par « jointure » on peut alors reconstruire de l'information dispersée dans plusieurs relations

Quel est le nom de la secrétaire qui s'occupe du dossier de l'étudiant 11 => Volin!

Étudiant

numEtudiant (int)	nomEtudiant (String)	bacEtudiant (String)	formationEtudiant (String)
11	Le moal	STI2D	BUT INFO
8	Le moal	Général	BUT STID
114	Veran	STI2D	BUT STID

BUT INFO **Kamp** **Volin**

=

Formation

formation (String)	nomChef (String)	nomSecrétaire (String)
BUT INFO	Kamp	Volin
BUT STID	Frambourg	Sorin
BUT TC	Bourdoncle	NULL

Jointure = outil pour « recoller les morceaux » de manière cohérente en utilisant des clés étrangères

Dualité Intension vs Compréhension d'une Relation

Notion de *schéma relationnel*

Les données dans les tables

Étudiant

numEtudiant (int)	nomEtudiant (String)	bacEtudiant (String)	formationEtudiant (String)
11	Le moal	STI2D	BUT INFO
8	Le moal	Général	BUT STID
114	Veran	STI2D	BUT STID

Formation

formation (String)	nomChef (String)	nomSecrétaire (String)
BUT INFO	Kamp	Volin
BUT STID	Frambourg	Sorin
BUT TC	Bourdoncle	NULL

Les données sur les données != métadonnées
= le schéma relationnel

```
Etudiants( numEtudiant : int (1),  
           nomEtudiant : String (NN)  
           bacEtudiant : String  
           @formationEtudiant : String REF Formations(formation) (NN)  
)
```

```
Formations( formation : String (1),  
            nomChef : String (2),  
            nomSecrétaire : String (UQ)  
)
```

(1) Attributs membres de la clé primaire

(2)(3)... Attributs membres de clés candidates

(NN) Attributs toujours valué (Not Null, interdiction de la valeur NULL)

(UQ) Attributs à valeurs uniques (mais attention le NULL est autorisé plusieurs fois !)

@Attribut clé étrangère : type REF Relation(Attributs clé primaire)