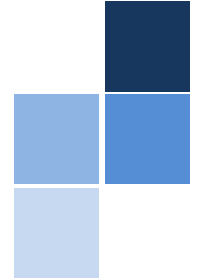




Chapitre 6.

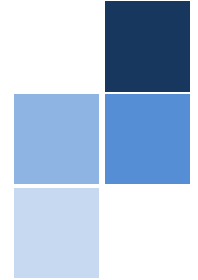
CONCEPTION DES BASES DE DONNÉES ET MODÈLE ENTITÉ-ASSOCIATION

Conception des bases de données



- Créer une base de données est une tâche complexe qui inclut la conception du schéma de la base de données, la conception des programmes qui accèdent et mettent à jour cette base de données et la prise en compte des autorisations d'accès.
- Le processus de création d'une base de données passe par quatre phases :
 - Spécification des besoins
 - Traduction des besoins en concepts dans un modèle conceptuel (type entité-association)
 - Traduction du modèle conceptuel en modèle logique (typiquement le modèle relationnel)
 - Conception physique (organisation des fichiers et index)

Conception des bases de données

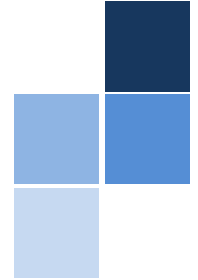


En conception, il faut éviter deux failles :

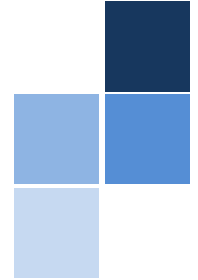
- La redondance : espace, problème de mise à jour.
Idéalement une information doit apparaître une seule fois
- L'incomplétude

Le modèle entité-association

- Le modèle entité-association se base sur trois concepts :
 - Entité
 - Association
 - Attribut



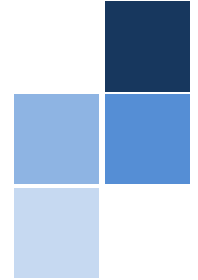
Le modèle entité-association



Les entités

- Une entité est une « chose » ou un « objet » du monde réel qui est clairement distinguable des autres objets.
- Par exemple, un employé dans une entreprise est une entité.
- Les entités de même type c'est-à-dire qui partagent le même ensemble d'attributs ou de propriétés sont regroupées sous le même concept.

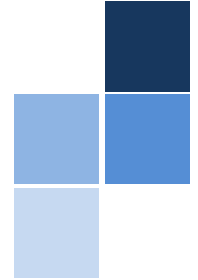
Le modèle entité-association



Les associations

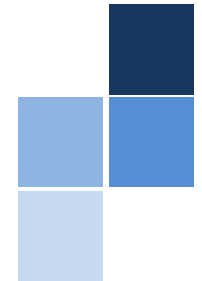
- Une association est une liaison entre plusieurs entités. Par exemple, un employé est associé aux projets sur lesquels il travaille
- Une association est une relation mathématique sur n ($n \geq 2$) entités
- Une entité a un rôle dans une association. Ce rôle n'est pas toujours défini explicitement
- Une association peut posséder un ou plusieurs attributs descriptifs. Par exemple l'attribut *heures* décrit l'association entre un employé et un projet et indique le nombre d'heures travaillées par cet employé sur le projet.

Le modèle entité-association



Les associations

- Une instance d'association doit être identifiée uniquement à partir des entités qui y participent.
- Si on souhaite modéliser les différentes dates de prise de fonction des chefs de département, il faut faire appel à une entité date.
- Il est possible d'avoir plusieurs associations qui font appel au même ensemble d'entités. Un employé est associé à un département en tant que simple employé ou en tant que chef de département
- Une association peut également être récursive, c'est-à-dire portant sur une seule entité



Le modèle entité-association

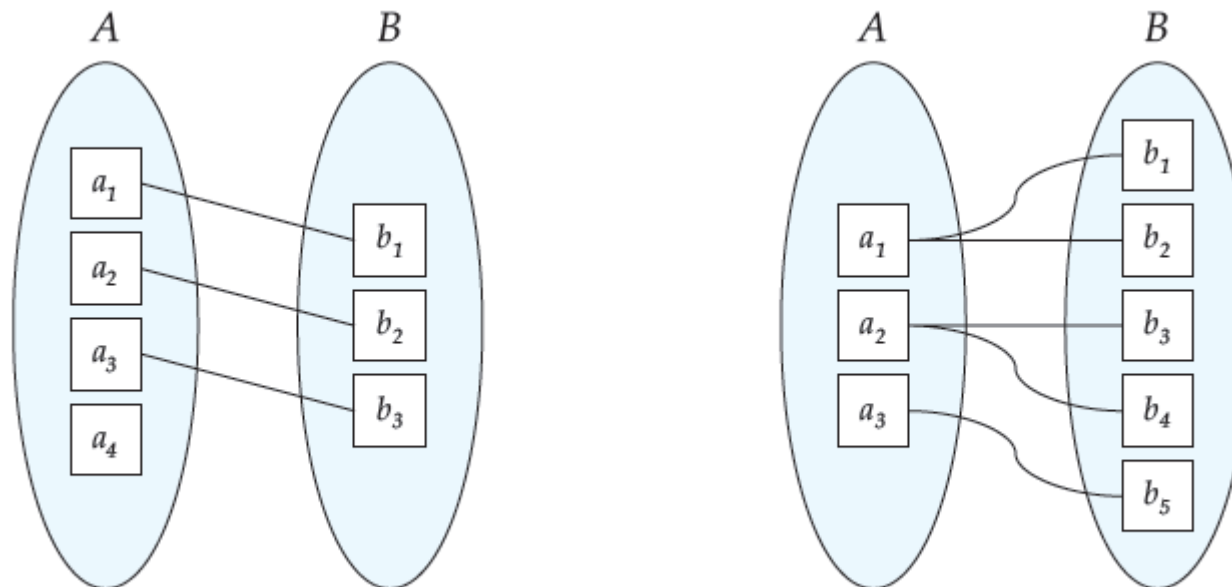
Les attributs

- Un attribut peut être :
 - Simple ou composé (adresse, nom-prénom, etc.)
 - Valeur unique ou « multivalué » (téléphones)
 - De base ou dérivé (âge)

Le modèle entité-association

Les cardinalités

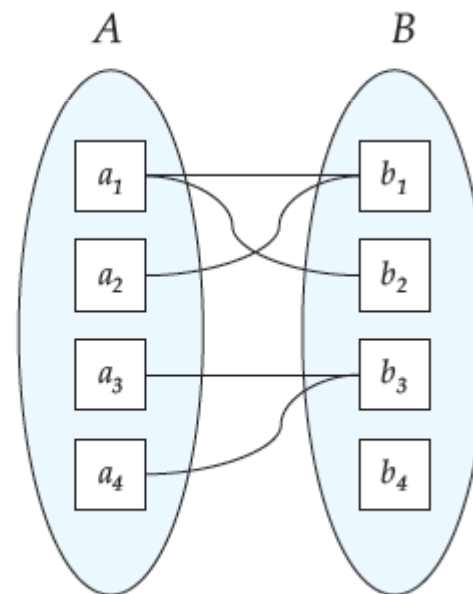
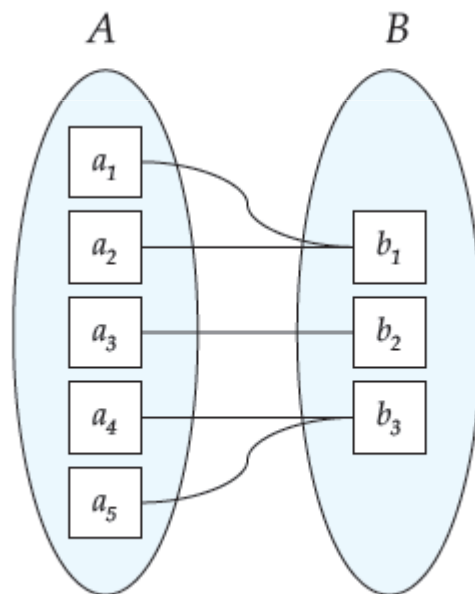
- le nombre d'entités auxquelles une autre entité peut être associé à travers une relation
- un-un
- un-plusieurs



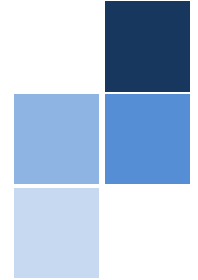
Le modèle entité-association

Les cardinalités

- plusieurs-un
- plusieurs-plusieurs



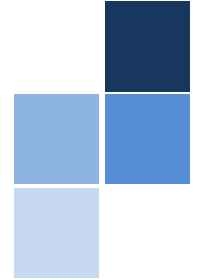
Le modèle entité-association



Les cardinalités

- On parle également de participation partielle et de participation totale dans une relation
- La participation d'une entité dans une association est totale si chaque instance de l'entité doit participer au moins une fois dans l'association

Le modèle entité-association

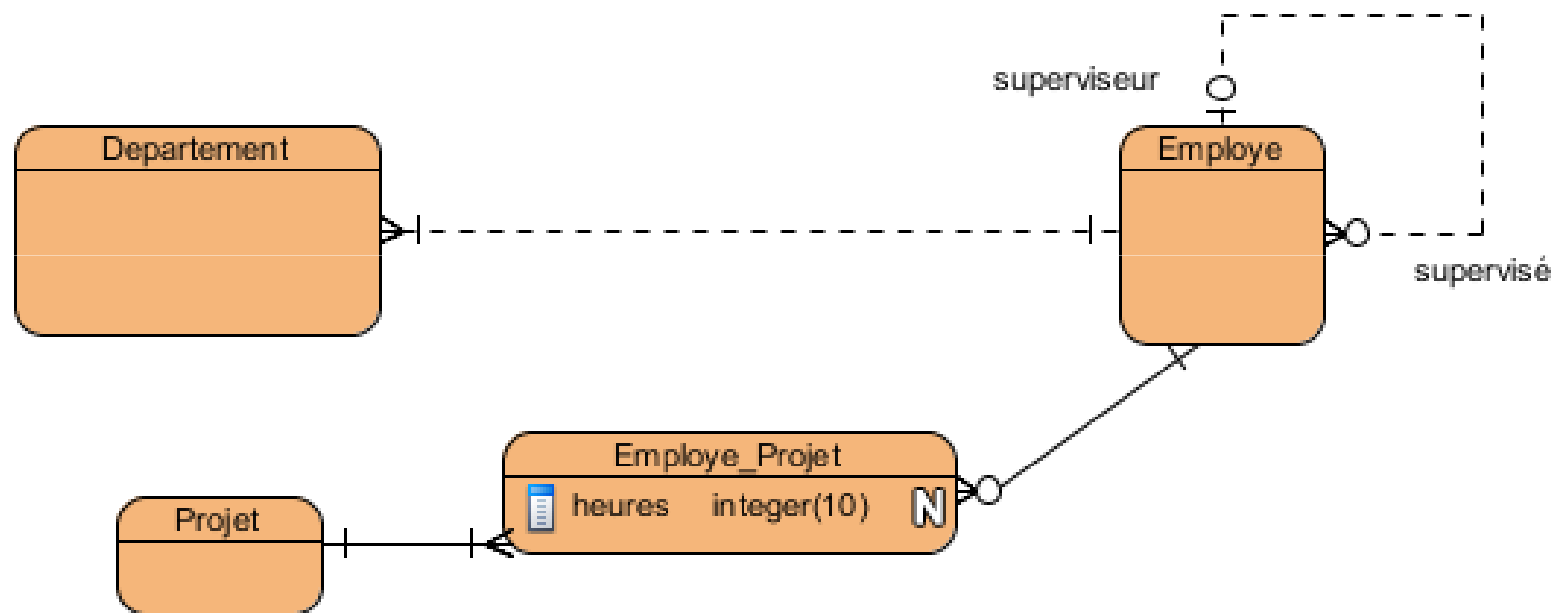


Les clés

- Comme en modèle relationnel, une clé d'entité est un ensemble d'attributs suffisant pour distinguer les entités les unes des autres.
- les clés permettent également d'identifier les relations qui existent entre les entités.
- L'union des clés des entités participant à la relation forment une super-clé pour la relation.
- Pour le cas des relations binaires, la clé d'une relation un-à-plusieurs est simplement la clé de l'entité du côté plusieurs.

Le modèle entité-association

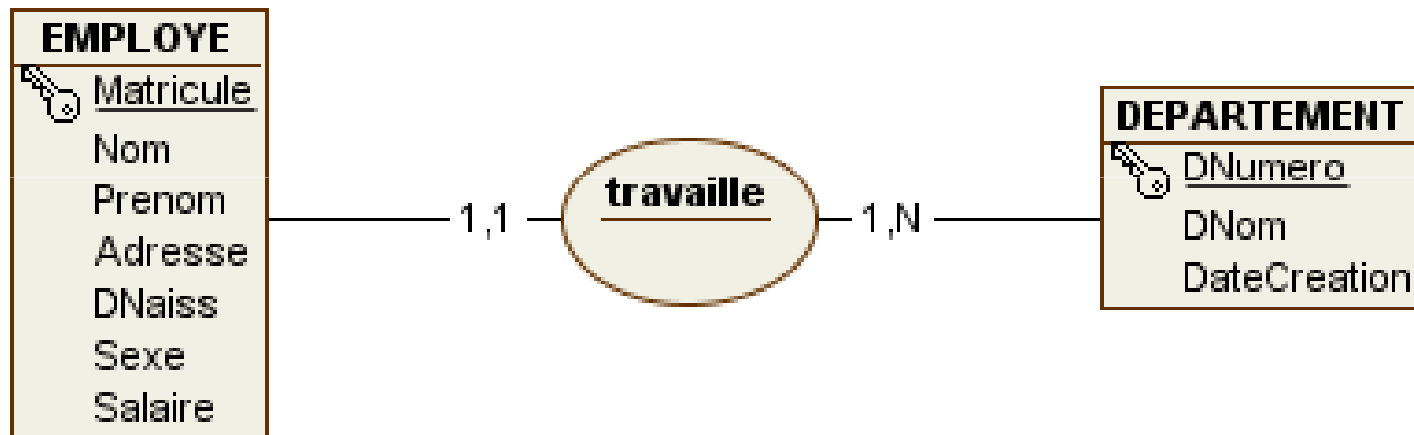
Crow's foot (patte d'oie)



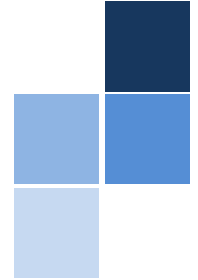
Le modèle entité-association

Merise

- Un employé travaille dans un département.



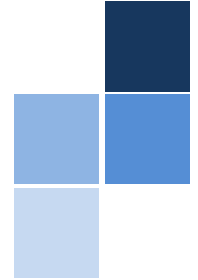
Le modèle entité-association



Généralisation/spécialisation

- **Exemple** : L'ensemble des employés peut être réparti en plusieurs sous-ensembles (secrétaires, techniciens, ingénieurs, etc.).
- Une classe fille d'une classe mère est une classe dont l'ensemble des instances est inclus dans l'ensemble des instances de la classe mère et qui **hérite** les attributs et les associations de cette dernière.
- Elle a en plus des attributs et des relations propres.
- Une instance de la classe fille est également instance de la classe mère.
- Une même instance peut faire partie de plusieurs classes filles de la même classe mère.

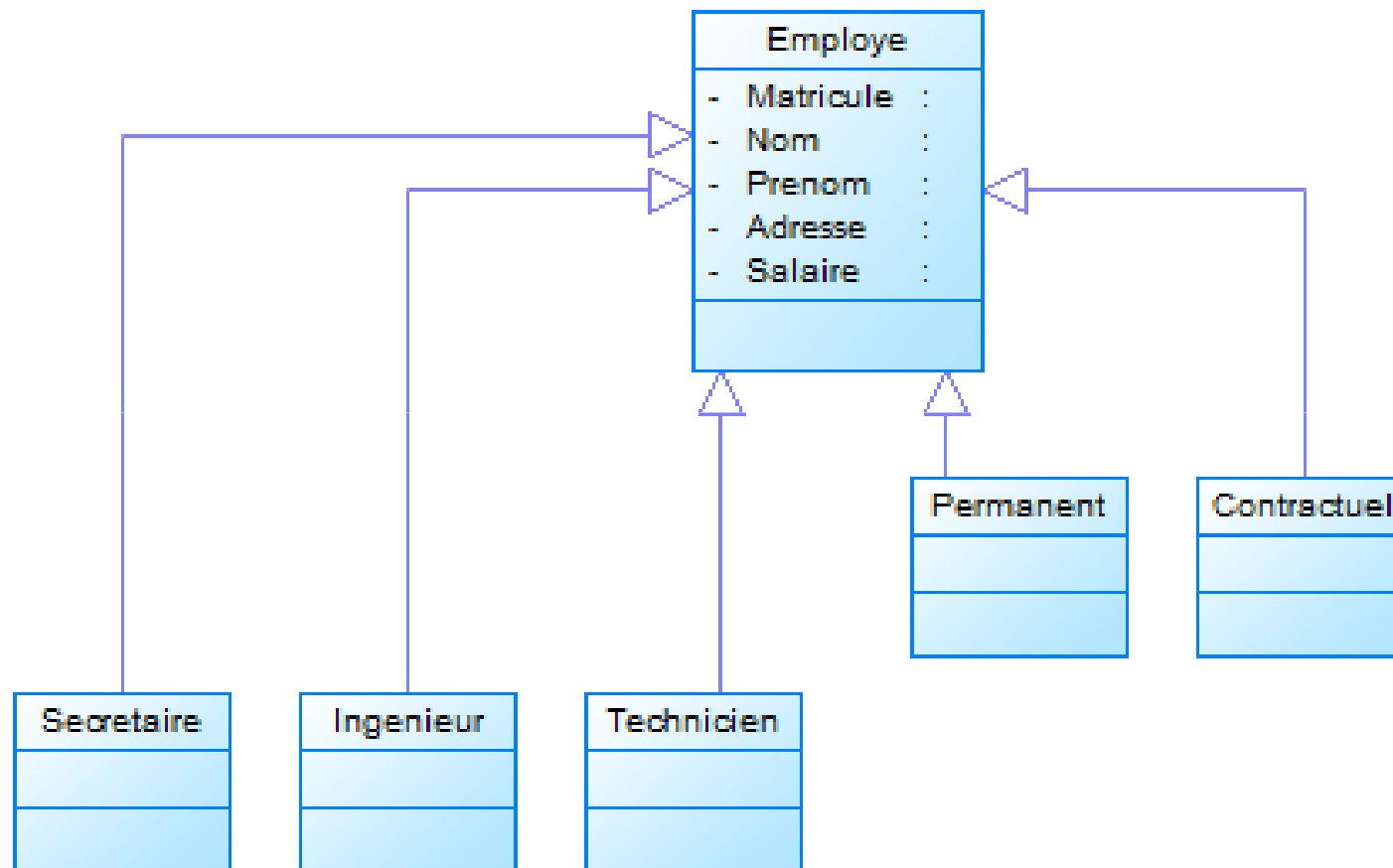
Le modèle entité-association



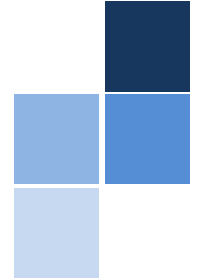
Généralisation/spécialisation

- La spécialisation est le processus de définition de classes filles pour une classe donnée
- On peut avoir plusieurs spécialisations pour une classe selon le critère de spécialisation
- La généralisation est le processus inverse de la spécialisation. Il consiste à créer une classe mère pour des classes qui ont des attributs en commun

Le modèle entité-association



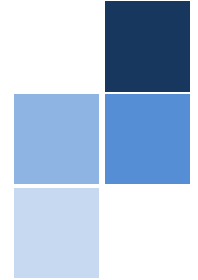
Transformation entité-association relationnel



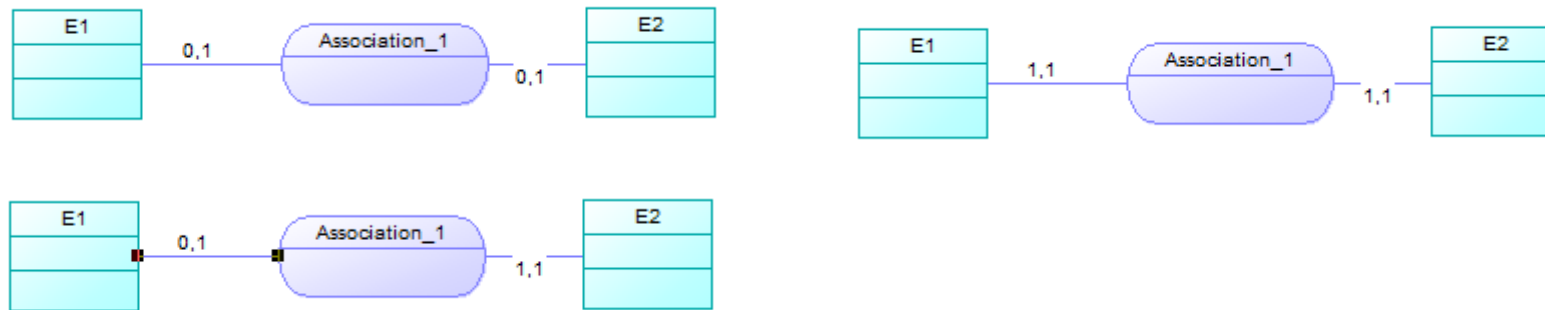
Les entités

- Une entité E du modèle entité-association est traduite en une relation R du modèle relationnel
- Une entité faible donne lieu à une relation avec comme clé étrangère la clé de l'entité dont elle dépend. Sa clé primaire sera la concaténation de cette clé étrangère et la clé « partielle » de l'entité faible

Transformation entité-association relationnel



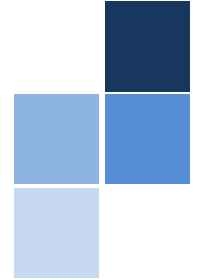
Les associations



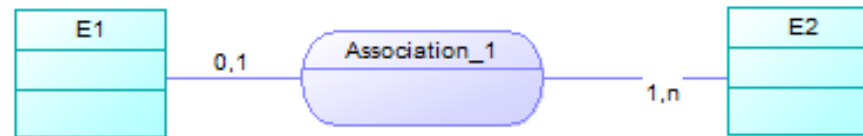
Solution 1 : Les entités E1 et E2 donnent lieu à la création de deux tables T1 et T2. L'une des deux tables ou chacune des deux est enrichie par la clé de l'autre.

Solution 2 : les entités E1 , E2 et Association1 donnent lieu à la création de trois tables T1, T2 et TAs. La table Tas contiendra les clés.

Transformation entité-association relationnel



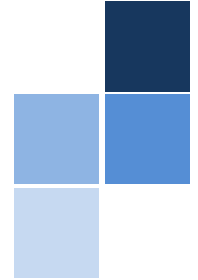
Les associations



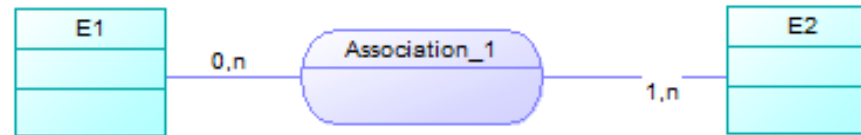
Solution 1 : Les entités E1 et E2 donnent lieu à la création de deux tables T1 et T2. seule la table T1 est enrichie par la clé de T2

Solution 2 : les entités E1 , E2 et Association1 donnent lieu à la création de trois tables T1, T2 et TAs. La table Tas contiendra les clés.

Transformation entité-association relationnel

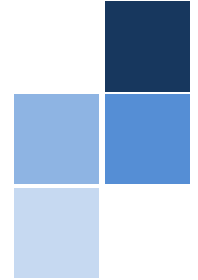


Les associations



Les entités E1 et E2 et Association vont donner lieu à trois tables T1, T2 et Tas. Si l'association est porteuse d'informations, elles seront mises dans la table Tas.

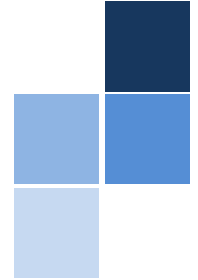
Transformation entité-association relationnel



Les spécialisations généralisations

- **Plusieurs relations : Classe mère et classes filles.**
 - On crée une relation pour la classe mère et on crée autant de relations que de classes filles. La relation représentant une classe fille est enrichie par la clé de la classe mère.
- **Plusieurs relations : classes filles**
 - On crée autant de relations que de classes filles. Une relation représentant une classe fille est enrichie par tous les attributs de la classe mère. (seulement si la spécialisation est totale et disjointe)

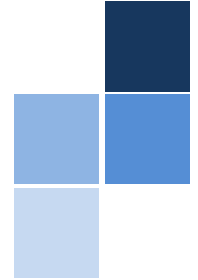
Transformation entité-association relationnel



Les spécialisations généralisations

- **Une seule relation avec un attribut type**
 - On crée une seule relation avec tous les attributs de la classe mère et des classes filles. On ajoute un attribut type qui indique la classe fille à laquelle appartient l'entité.
- **Une seule relation avec plusieurs attributs types**
 - On crée une seule relation avec tous les attributs de la classe mère et des classes filles. On ajoute des attributs booléens pour toutes les classes filles avec la valeur vraie si l'entité appartient à cette classe fille.

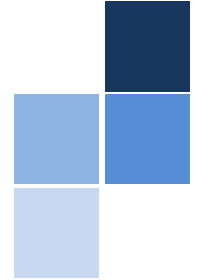
Le modèle entité-association



Exercice 1

La société *Nord Fournitures* reçoit des commandes de ses clients. Ces commandes précisent les produits et les quantités à livrer. La Société *Nord Fournitures* procède alors, selon les disponibilités, à la livraison aux différentes agences des clients. Les livraisons peuvent être partielles et donnent lieu à un bon de livraison. Une fois une commande totalement satisfaite, une facture est envoyée au client.

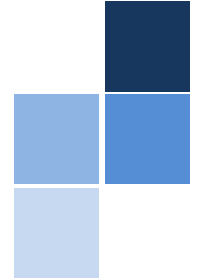
Le modèle entité-association



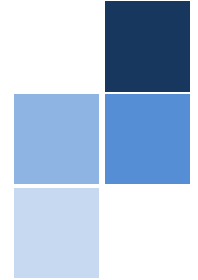
Exercice 2

Le parc informatique d'une entreprise est composé d'un ensemble de machines de types différents (ordinateurs, imprimantes, scanners, onduleurs, routeurs, ...). Chaque machine a des caractéristiques propres (vitesse processeur, capacité RAM, capacité disque dur, vitesse impression, autonomie, puissance, résolution,...). Ces machines sont affectés à des employés. Elles peuvent faire l'objet de maintenance préventive (nettoyage, réinstallation) ou curative (réparation ou remplacement composants).

Dépendances fonctionnelles et normalisation

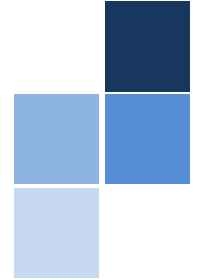


- Le regroupement des attributs d'un domaine dans des relations est souvent un choix subjectif du concepteur.
- Une bonne conception préserve les informations et minimise la redondance
- Il existe une théorie ; la **normalisation**, qui permet de mesurer la qualité d'un schéma de manière formelle. Elle se base sur le concept de **dépendance fonctionnelle**.



Règles non formelles de la modélisation

- S'assurer de la non ambiguïté de la sémantique des attributs dans le schéma relationnel
- Réduire la redondance au niveau des tuples
- Réduire les attributs NULL au niveau des tuples
- S'assurer de l'absence de faux tuples (suite aux jointures)

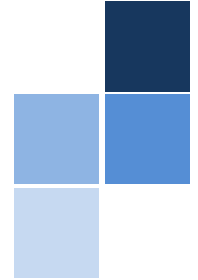


Redondance et anomalies : Insertion

Matricule	Nom_prenom	Adresse	Dept_Num	Dept_Nom	Dept_Chef
1201	Bekkali Ahmed	Tetouan	1	Informatique	1001
1540	Tijani Hassan	Tanger	2	Comptabilité	1200
1621	Alaoui Rachid	Tetouan	1	Informatique	1001
1842	Ahmadi Kamal	Tanger	2	Comptabilité	1200

- Insérer un nouvel employé implique l'insertion du département d'attache de cet employé (avec risque d'erreur au niveau du nom de département ou du chef de département)
- Les attributs *Dept_nom* et *Dept_Chef* doivent être mis à NULL pour un employé dont le département est inconnu
- il est difficile d'insérer les informations d'un département qui n'a pas d'employés.

Dépendances fonctionnelles et normalisation



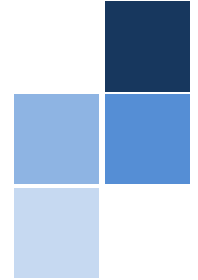
Redondance et anomalies : Suppression

- Supprimer le dernier employé rattaché à un département implique la suppression de ce département

Redondance et anomalies : Modification

- Quand on modifie la valeur d'un attribut du département, il faut le faire pour tous les employés rattachés à ce département.

Dépendances fonctionnelles et normalisation



- Une **dépendance fonctionnelle** entre un ensemble d'attributs X et un ensemble d'attributs Y d'une relation R exprime la contrainte qui stipule que si pour deux tuples t_1 et t_2 de R , on a $t_1[X] = t_2[X]$ alors $t_1[Y] = t_2[Y]$
- Autrement dit, les valeurs de attributs de Y dépendent de celles des attributs de X

Remarque :

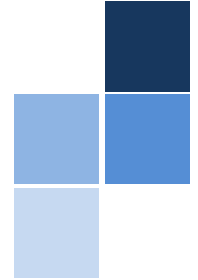
- Si X est une clé candidate de R alors $X \rightarrow Y$ pour tout sous-ensemble Y de R

Exemples :

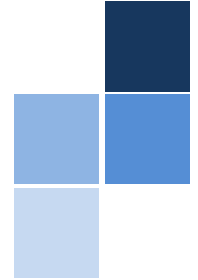
Matricule \rightarrow Nom_prenom

Dept_Num \rightarrow Dept_Nom

Dépendances fonctionnelles et normalisation



- Une relation est dite normalisée si elle vérifie un ensemble de tests bien définis.
- On parle de **forme normale**.
- Codd a défini en 1972 trois formes normales : 1FN, 2FN et 3FN
- Une définition plus stricte de la 3FN est appelée FN de Boyce-Codd (BCNF)
- La définition des formes normales repose sur les dépendances fonctionnelles

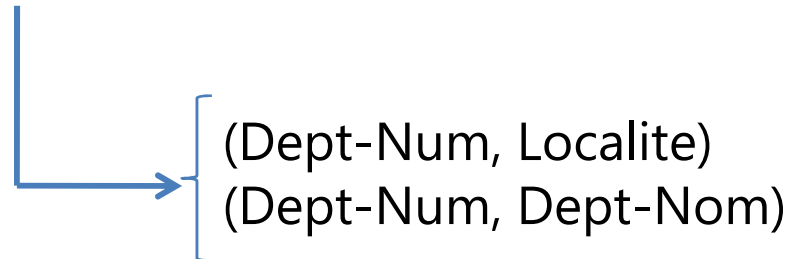


Première Forme Normale 1FN

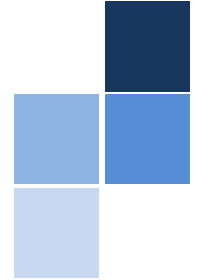
- La première forme normale fait maintenant partie de la définition du modèle relationnel.
- Une relation est en 1FN si tous ses attributs sont atomiques (listes et attributs composés sont interdits)

-Exemple normalisation 1FN :

(Dept-Num, Dept-Nom, {Localite})



Dépendances fonctionnelles et normalisation

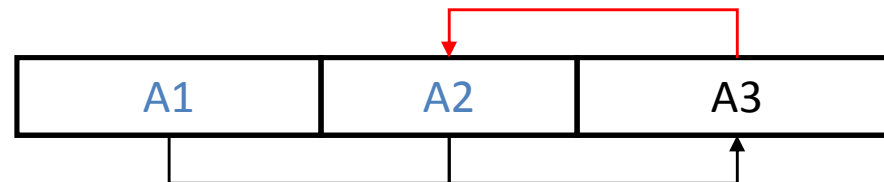


Forme Normale Boyce-Codd (BCNF)

-Une relation R est en **BCNF** si pour toute dépendance fonctionnelle $X \rightarrow A$ (X et A sous-ensembles des attributs de R), une des deux conditions suivantes est vraie :

- $X \rightarrow A$ est triviale ($A \subseteq X$)
- X est une super-clé.

Exemple :



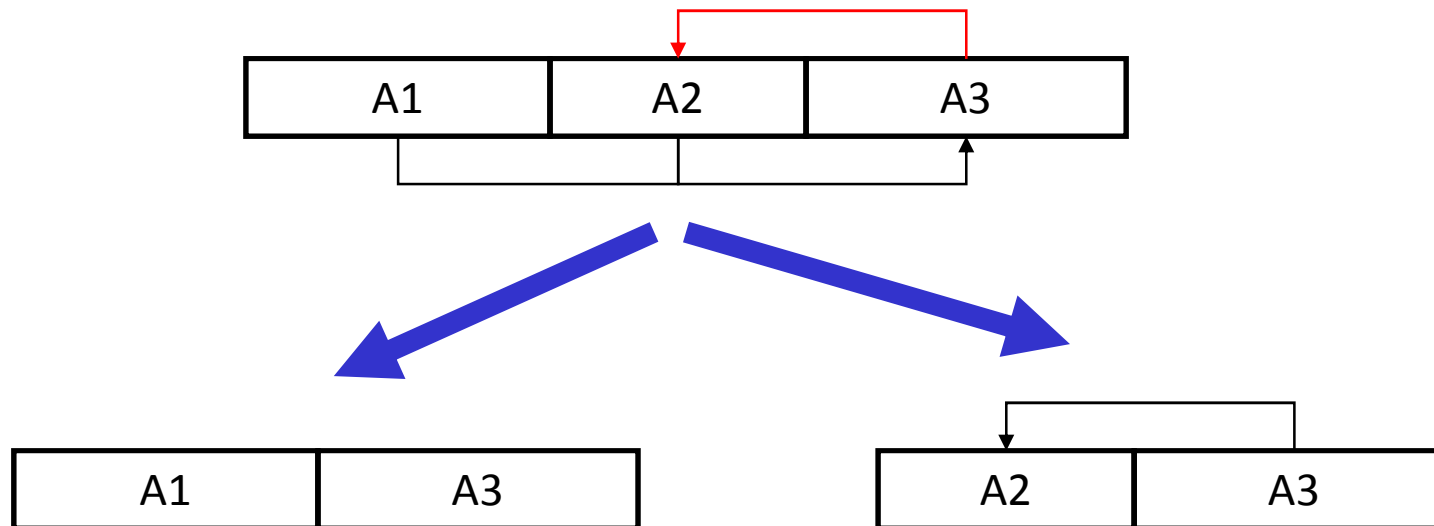
N'est pas en BCNF car $A3 \rightarrow A2$

Dépendances fonctionnelles et normalisation

Normalisation en BCNF

Dans une relation R qui n'est pas en BCNF, il existe au moins une dépendance fonctionnelle non triviale $X \rightarrow A$ où X n'est pas super-clé.

R est alors décomposée en $X \cup A$ et $R - (A - X)$



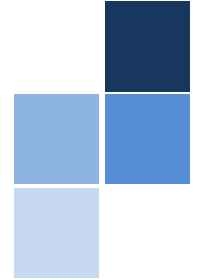
Dépendances fonctionnelles et normalisation



Normalisation en BCNF et préservation des dépendances

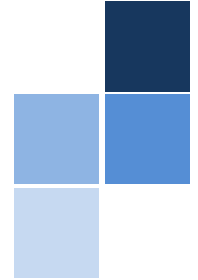
Soit la relation $R(A,B,C)$ où AB est une clé et avec les dépendances fonctionnelles suivantes :

- $B \rightarrow C$ et $AC \rightarrow B$
- Cette relation n'est pas en BCNF.
- Une transformation possible en BCNF sera $R_1(B,C)$ et $R_2(A,B)$
- Cette décomposition ne préserve pas la dépendance fonctionnelle $AC \rightarrow B$



Troisième forme Normale 3FN

- Une relation R est en **3FN** si pour toute dépendance fonctionnelle $X \rightarrow A$, (X et A sous-ensembles des attributs de R) une des conditions suivantes est vraie :
 - $X \rightarrow A$ est triviale ($A \subseteq X$)
 - X est une super-clé de R.
 - Tout attribut Y dans A-X appartient à une clé candidate de R



Exercice

Dans chacun des cas ci-dessous, dire si les relations sont en 3FN, proposer une décomposition en 3FN si nécessaire :

- a. $R(\text{ville}, \text{rue}, \text{codePostal})$ $F = \{\text{Ville}, \text{Rue} \rightarrow \text{CodePostal}, \text{CodePostal} \rightarrow \text{ville}\}$
- b. $R(A, B, C)$ $F = \{A \rightarrow B, B \rightarrow C\}$
- c. $R(A, B, C, D)$ $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$
- d. $R(A, B, C, D)$ $F = \{B \rightarrow C, B \rightarrow D\}$
- e. $R(A, B, C, D)$ $F = \{AB \rightarrow C, BC \rightarrow D, CD \rightarrow A, AD \rightarrow B\}$