
III. Modèle Relationnel

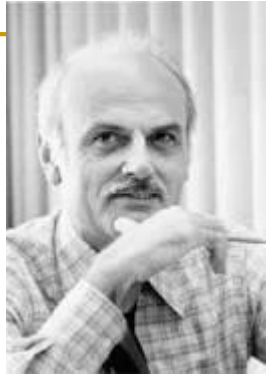
Objectifs

- ❑ Saisir le sens des concepts du modèle relationnel
- ❑ Transformer un schéma conceptuel de données en un schéma relationnel

Plan

1. Concepts de base
2. Règles d'intégrité structurelle
3. Règles de traduction ou passage du diagramme E/R au schéma relationnel

1. Concepts de base



- ❑ Le modèle relationnel a été introduit, en 1970, par le mathématicien Edgar Frank Codd (1923/2003, Turing Award 1981)
 - ❑ Il s'agit d'un modèle ensembliste simple :
 - ❑ Utilisant une terminologie propre (d'origine mathématique)
 - ❑ Se reposant sur des théories formelles (théorie ensembliste et théorie des prédicats)
 - Il présente deux aspects fondamentaux :
 - ❑ Une **démarche** de conception permettant de définir une collection de relations
-
- ❑ Une **algèbre** permettant de manipuler les tables ou relations

1. Concepts de base

Une **BD Relationnelle** est vue par ses utilisateurs comme une collection de **relations** ou **tables**.

Représentation **implicite** d'une table :

Nom_Relation (Clé, attrib1, attrib2, ..., attribn)

Représentation **explicite** d'une table :

| Clé | attrib1 | attrib2 | ... |
|-----|---------|---------|-----|
| | | | |
| | | | |
| | | | |

1. Concepts de base - Relation

- ❑ **RELATION / TABLE** : En colonne, on porte les **attributs** qui doivent être atomiques (élémentaires, non décomposables) et en ligne les **tuples** ou **n-uplets**
- ❑ **TUPLE** : Formé par une valeur de chacun des attributs. Tous les tuples différents, on n'autorise pas la duplication
- ❑ **DOMAINE** : Ensemble borné et fini de valeurs possibles ou existantes, que peut prendre un attribut d'une relation. A deux attributs différents correspondent deux domaines différents

1. Concepts de base - Clés

- ❑ **CLÉ PRIMAIRE** : Chaque table doit obligatoirement en posséder une. C'est un ensemble d'attributs dont la valeur identifie le tuple. Il ne peut y avoir deux tuples ayant la même valeur de la clé primaire, la clé doit être **unique**
- ❑ **CLÉ ÉTRANGÈRE** : Ensemble d'attributs dans une table qui est aussi une clé primaire dans une autre table

1. Concepts de base - Clés

- ❑ **CLÉ CANDIDATE** : Autre la clé primaire, on peut en avoir une ou plusieurs, c'est un ensemble d'attributs qui peut être retenu comme clé primaire
- ❑ **CLÉ SECONDAIRE** : C'est une clé candidate qui n'a pas été retenue comme clé primaire.

1. Concepts de base - Clés

Exemple :

Soit la table **Étudiant** (*NCE, NCIN, Nom, Prénom, DatNaiss, Adresse, Tél, Parcours, Niveau, Groupe*)

Les clés *candidates* sont : **NCE, NCIN, Tél, Nom+Prénom** (s'il n'y a pas d'homonymes)

Si on retient **NCE** comme clé *primaire*

Les attributs **NCIN, Tél, Nom+Prénom** forment alors des clés *secondaires*

1. Concepts de base - Divers

- ❑ **CARDINALITÉ** : Nombre de tuples dans une table
- ❑ **DEGRÉ** : Nombre d'attributs dans une table
- ❑ **VUE** : C'est une table (relation) virtuelle, sans existence physique, dérivée d'une ou plusieurs tables (relations) de base par la commande **Create View**

2. Règles d'Intégrité Structurelle

La cohérence et l'intégrité des données sont régies par les règles d'intégrité suivantes :

- ❑ L'intégrité du **domaine** : Contrôle de validité sur chaque valeur d'attribut
- ❑ L'intégrité de l'**entité** : Toute table possède une clé primaire et tout attribut participant à cette clé primaire doit être non vide
- ❑ L'intégrité **référentielle** : Règle régissant la clé étrangère et imposant que la valeur d'une clé étrangère doit être soit vide « NULL », soit une valeur existante de la clé primaire dans la table liée

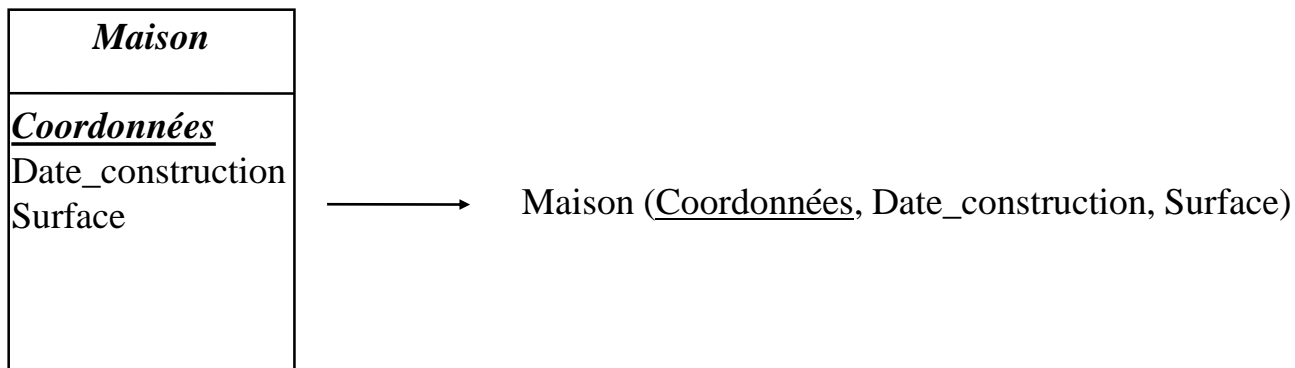
3. Transformation du formalisme Entité/Relation en formalisme relationnel

La Modélisation logique des données consiste en la transformation du schéma conceptuel, exprimé en formalisme « Entité/Relation », en un schéma exprimé dans un formalisme logique adapté au SGBD envisagé.

Le schéma logique sera ensuite transformé et adapté en fonction des spécificités du langage de définition de données (LDD) spécifique à l'outil retenu, pour devenir un schéma physique de données.

3. Règles de traduction

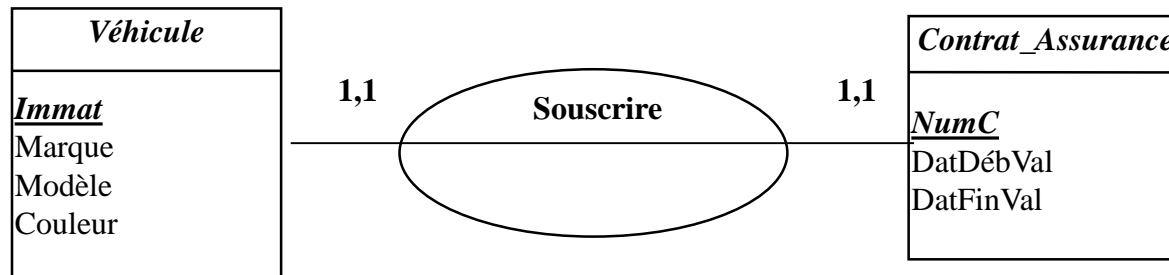
- ❑ Chaque entité qui figure dans le diagramme E/R est traduite par une table (relation)
- ❑ Ses propriétés deviennent des attributs de la table correspondante
- ❑ Son identifiant devient la clé primaire unique de la table correspondante



3. Règles de traduction

Pour toute association binaire un à un (cardinalité 0,1 ou 1,1 d'un côté et de l'autre), on a le choix entre l'échange des clés primaires des deux tables issues des entités ou bien leur fusion en une seule table

Exemple 1 : Echange des clés primaires

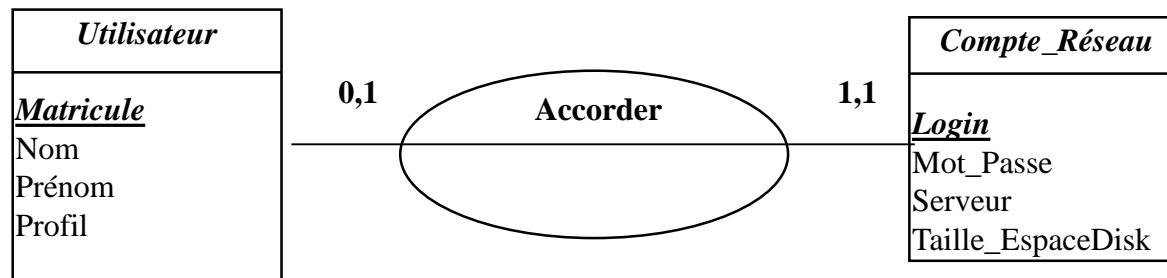


Véhicule (Immat, Marque, Modèle, Couleur, NumC#)

Contrat_Assurance (NumC, DatDébVal, DatFinVal, Immat#)

3. Règles de traduction

Exemple 2 : Fusion



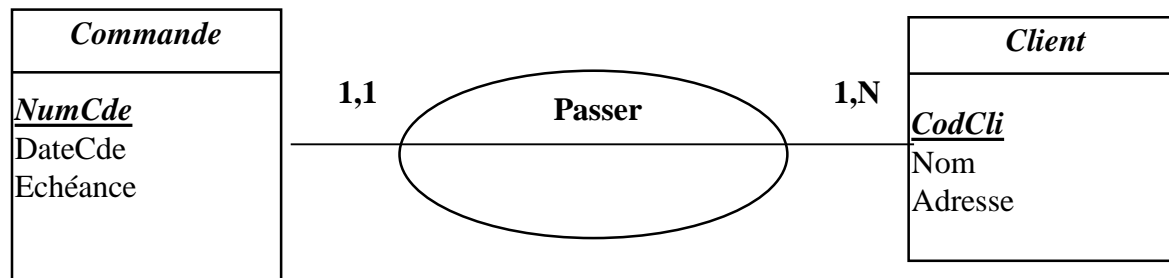
Cpte_User (Matricule, Nom, Prénom, Profil, Login, Mot_Passe, Serveur, Taille_EspaceDisk)

3. Règles de traduction

Pour toute association binaire un à plusieurs (cardinalité 0,1 ou 1,1 d'un côté et 0,N ou 1,N de l'autre), il faut mettre l'identifiant de la table issue de l'entité mère (à cardinalité 0,N ou 1,N) dans la table issue de l'entité fille (à cardinalité 0,1 ou 1,1) là où il devient une clé étrangère.

3. Règles de traduction

Exemple :



Client (CodCli, Nom, Adresse)

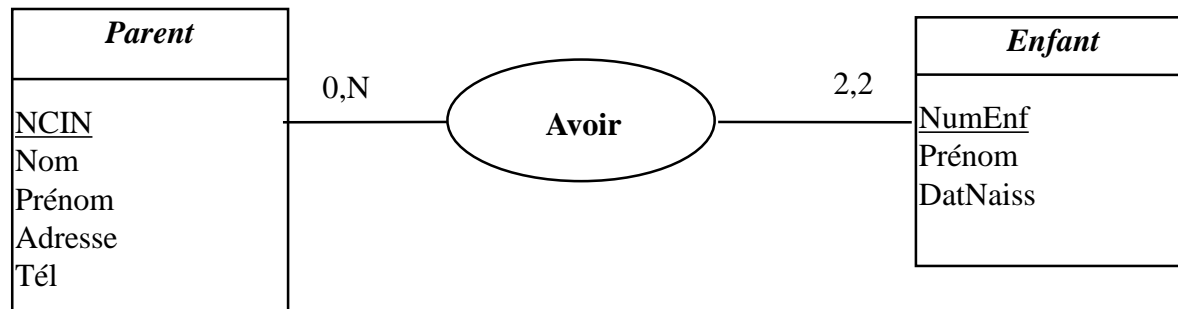
Commande (NumCde, DateCde, Echéance, CodCli#)

3. Règles de traduction

Pour toute association binaire (cardinalité 0,N ou 1,N d'un côté et 2,2 de l'autre), il faut mettre l'identifiant de la table issue de l'entité mère (à cardinalité 0,N ou 1,N) dans la table issue de l'entité fille (à cardinalité 2,2) **deux fois** et on procède à un changement d'appellation de l'attribut dupliqué pour éviter l'ambiguïté. Ces deux attributs sont des clés étrangères.

3. Règles de traduction

Exemple 1 :

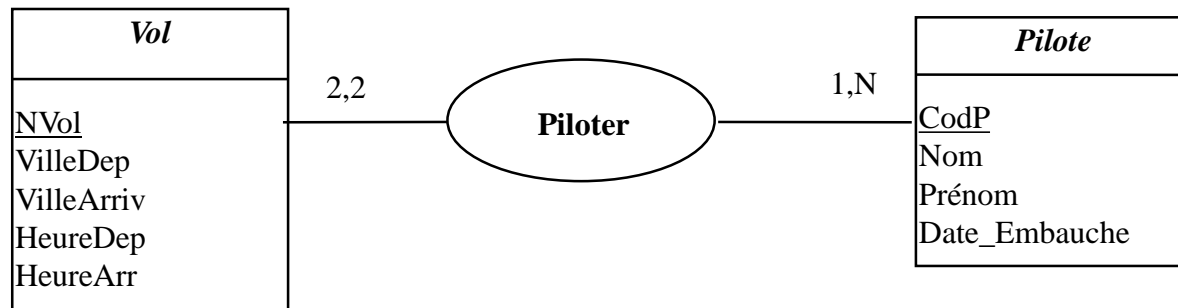


Parent (NCIN, Nom, Prénom, Adresse, Tél)

Enfant (NumEnf, Prénom, DatNaiss, NCINPère#, NCINMère#)

3. Règles de traduction

Exemple 2 :



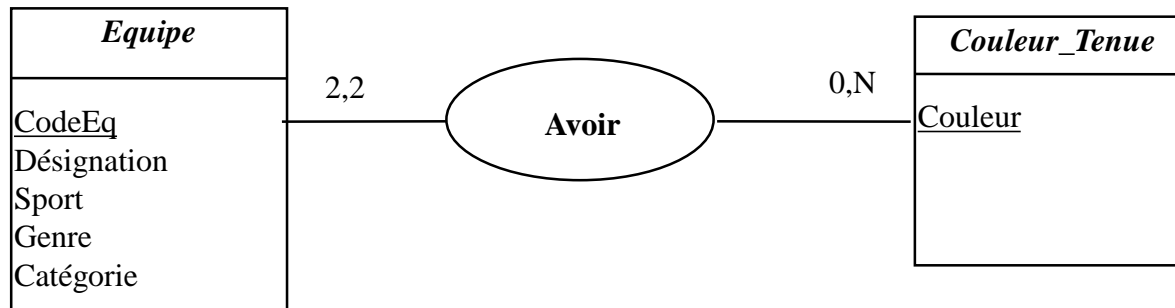
Pilote (CodP, Nom, Prénom, Date_Embauche)

Vol (NVol, VilleDep, VilleArriv, HeureDep, HeureArr,
CodPilote#, CodCoPilote#)

3. Règles de traduction

Exemple 3 :

Hypothèse : Chaque équipe possède 2 couleurs dans sa tenue



Equipe (CodeEq, Désignation, Sport, Genre, Catégorie, Couleur1#, Couleur2#)

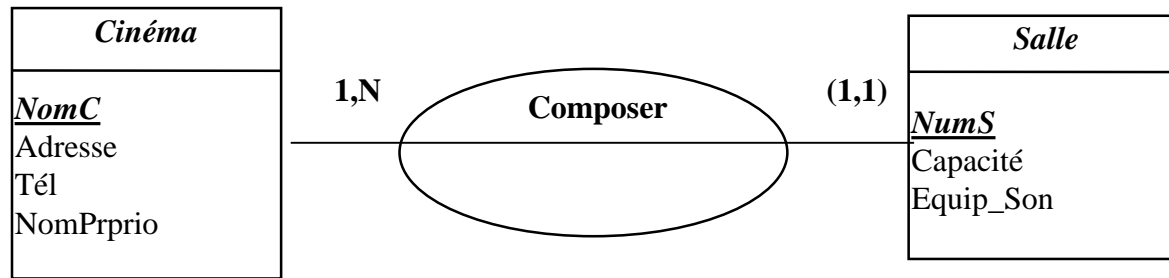
3. Règles de traduction

Pour toute association un à plusieurs faisant intervenir une **entité faible** (cardinalité 0,N ou 1,N d'un côté et (0,1) ou (1,1) de l'autre) : il faut mettre l'identifiant de la table issue de l'entité mère (à cardinalité 0,N ou 1,N) dans la table issue de l'entité fille (à cardinalité (0,1) ou (1,1)), cet identifiant participera à la clé de la table issue de l'entité fille.

Si cette association est porteuse de données, ces dernières migrent avec l'identifiant.

3. Règles de traduction

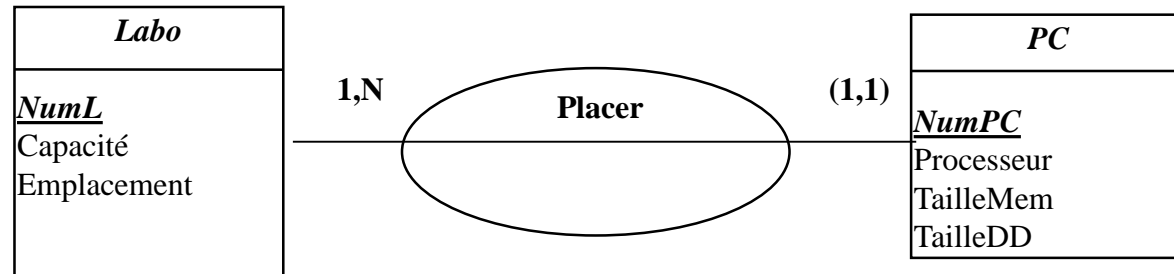
Exemple 1:



Cinéma (NomC, Adresse, Tél, NomProprio)

Salle (NumS, NomC#, Capacité, Equip_Son)

Exemple 2:



Labo (NumL, Capacité, Emplacement)

PC (NumPC, NumL#, Processeur, TailleMem, TailleDD)

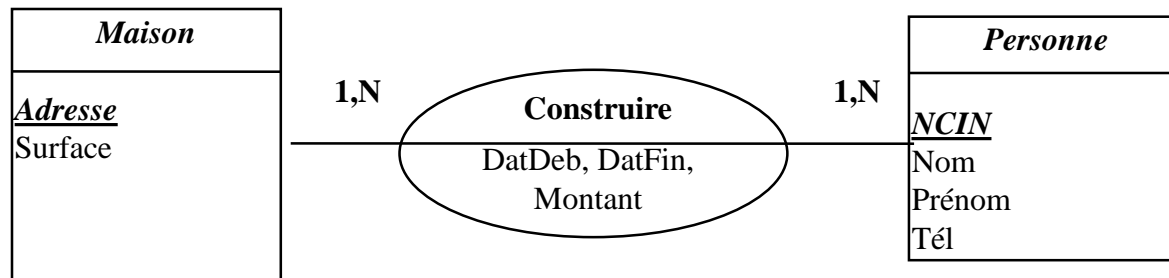
3. Règles de traduction

Pour toute association binaire plusieurs à plusieurs (cardinalité 0,N ou 1,N d'un côté et de l'autre), on crée une nouvelle table, sa clé sera composée des clés primaires des tables issus des entités qui participent à cette association.

Dans le cas où cette association est porteuse de données, on les fait migrer dans la nouvelle table.

3. Règles de traduction

Exemple :



Maison (Adresse, Surface)

Personne (NCIN, Nom, Prénom, Tél)

Construire (NCIN#, Adresse#, DatDeb, DatFin, Montant)

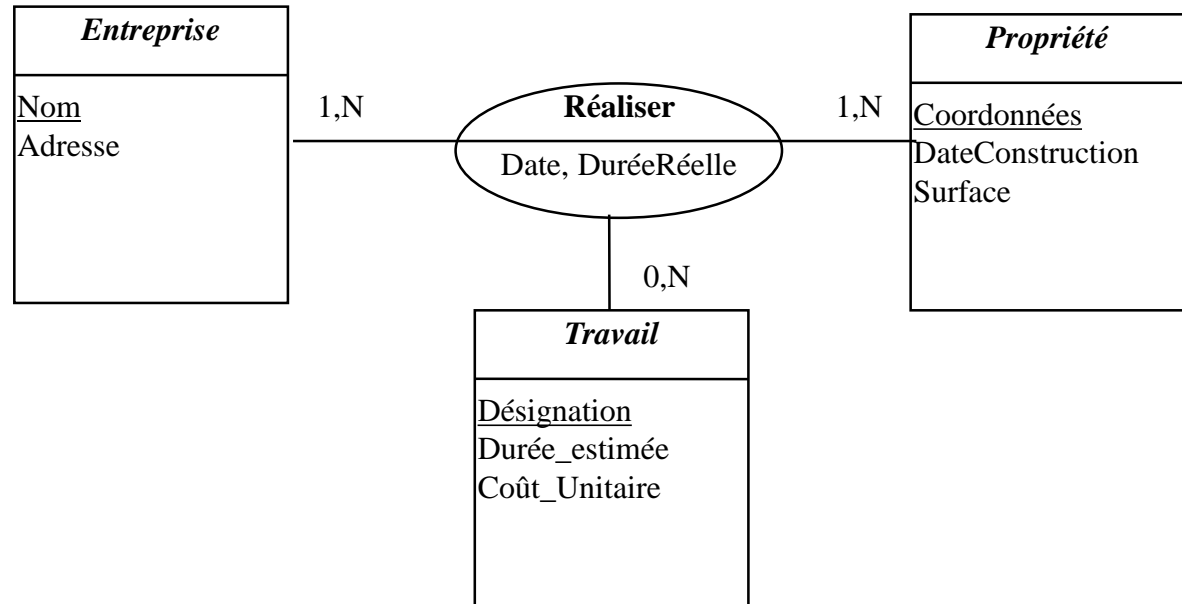
3. Règles de traduction

Pour toute association plus que binaire, dite n -aire (qui fait intervenir n entités avec $n > 2$), on crée une nouvelle table ayant comme clé une clé composée des identifiants des diverses entités reliées par l'association considérée.

Les éventuelles propriétés de cette association deviennent des attributs de la nouvelle table.

3. Règles de traduction

Exemple :



Entreprise (Nom, Adresse)

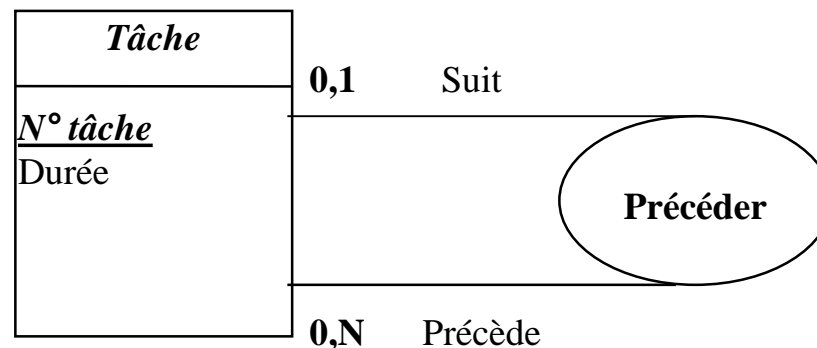
Propriété (Coordonnées, DateConstruction, Surface)

Travail (Désignation, Durée_estimée, Coût_Unitaire)

Réaliser (Coordonnées#, Nom#, Désignation#, Date, DuréeRéelle)

3. Règles de traduction

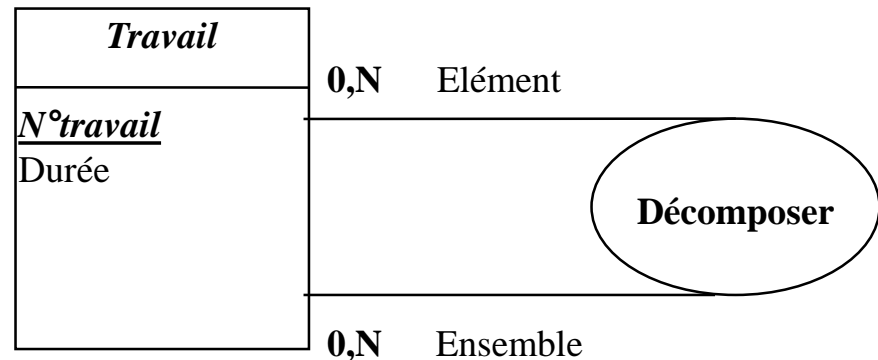
Association réflexive un à plusieurs (cardinalité 0,1 ou 1,1 d'un côté et 0,N ou 1,N de l'autre) : On duplique la clé de la table issue de l'entité, on procède à un changement d'appellation de l'attribut dupliqué. Les éventuelles propriétés de cette association deviennent des attributs de la table issue de l'entité.



Tâche (N°tâche, N°tâcheprécédente#, Durée)

3. Règles de traduction

Association réflexive plusieurs à plusieurs (cardinalité 0,N ou 1,N d'un côté et de l'autre) : On crée une nouvelle table, sa clé sera formée de deux fois la clé de la table issue de l'entité et on procède à un changement d'appellation de l'attribut dupliqué. Les éventuelles propriétés de cette association deviennent des attributs de la table issue de l'entité.



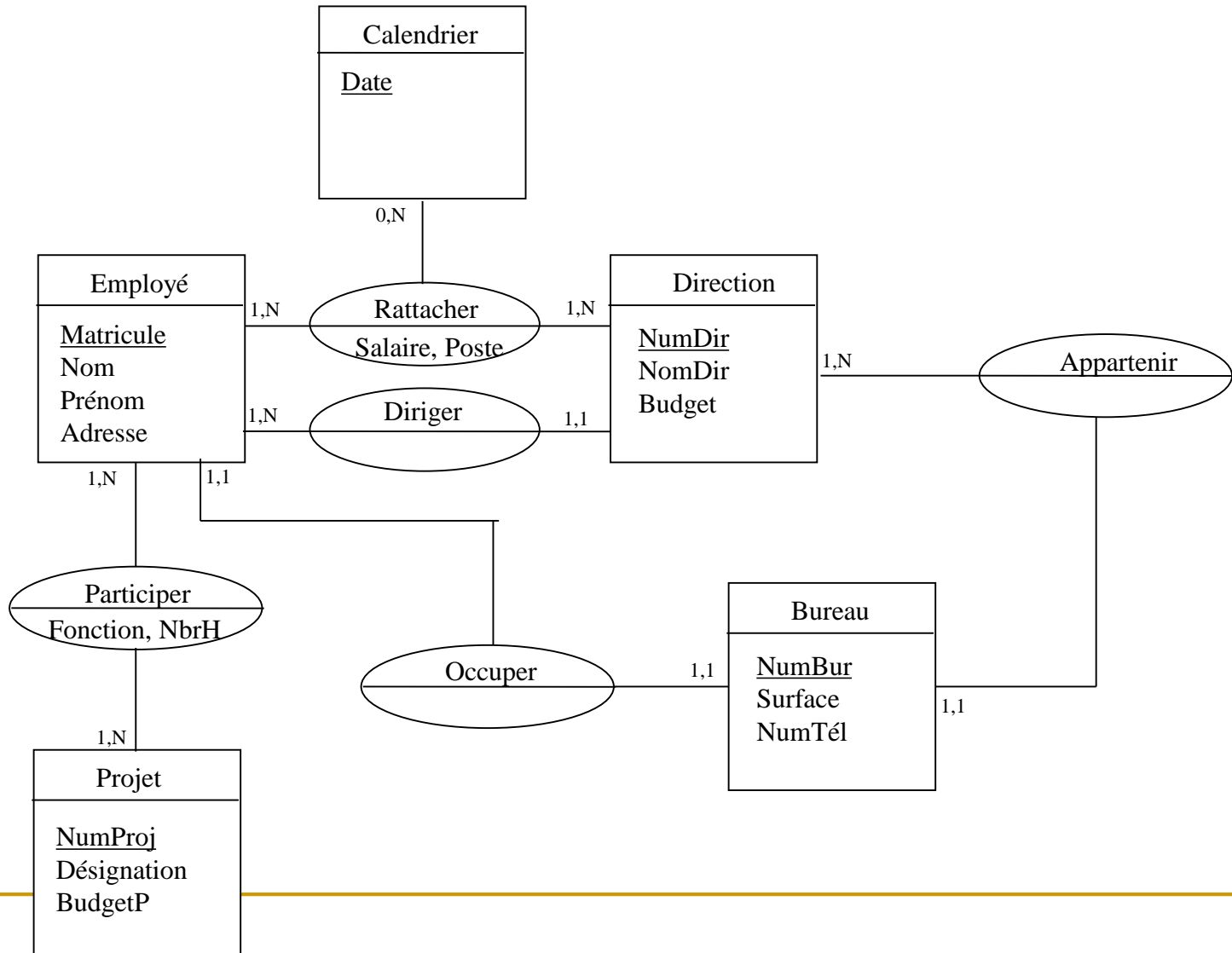
Travail (*N°travail*, *Durée*)

Décomposer (*N°Travailcomposé#*, *N°Travailcomposant#*)

Exemple récapitulatif – Énoncé

On veut mettre en place une base de données afin de gérer le personnel d'une entreprise. Chaque employé, identifié par un matricule, est rattaché à une direction, occupe un bureau personnel et participe à un ou plusieurs projets développés par cette entreprise. Pour chaque employé, on désire conserver son nom, son prénom, son adresse et la liste des salaires qu'il a touchés dans les différents postes qu'il a occupés au fil du temps. Chaque direction, identifiée par son numéro, est décrite par son nom et son budget et est dirigée par un directeur faisant partie du personnel. Chaque bureau, identifié par son numéro et caractérisé par une surface exprimée en mètres carrés, possède un numéro de téléphone et appartient à une direction. Chaque projet, identifié par son numéro, possède une désignation et un budget, il emploie plusieurs personnes. Relativement à un projet, chaque employé assure une fonction et effectue un certain nombre d'heures de travail.

Exemple récapitulatif – Diagramme E/R



Exemple récapitulatif – Traduction du schéma conceptuel en schéma logique (relationnel)

- **Employé** (Matricule, Nom, Prénom, Adresse, NumBur#)
- **Direction** (NumDir, NomDir, Budget, MatriculeDir#)
- **Projet** (NumProj, Désignation, BudgetP)
- **Bureau** (NumBur, Surface, NumTél, Matricule#, NumDir#)
- **Participer** (Matricule#, NumProj#, Fonction, NbrH)
- **Rattacher** (Matricule#, NumDir#, Date, Salaire, Poste)