

Modélisation Entité/Association

La phase de design d'une BD

- Analyse des besoins
- Design conceptuel
 - Modèle EA, UML
- Design logique
 - de EA vers des tables
 - normalisation de schéma

Modèle Entité/Association (ER)

- Entité

- similaire à la notion d'objet, représente une «chose» du monde réel

- Type d'entité

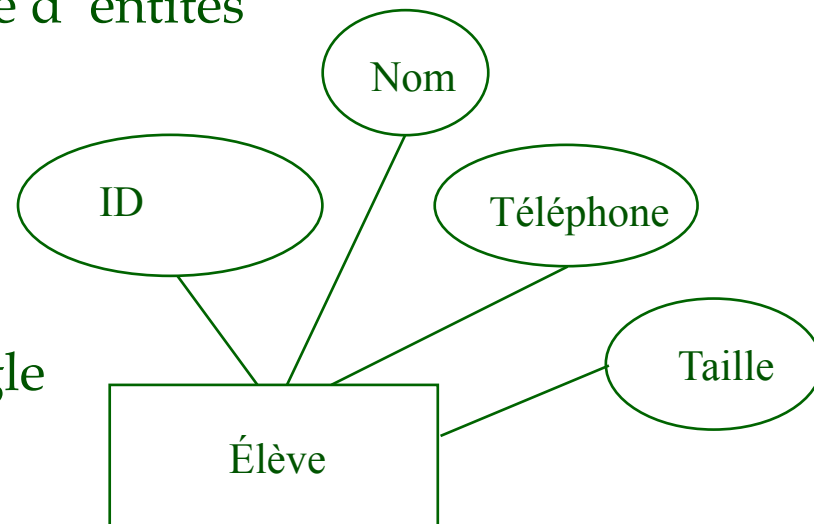
- similaire à la notion de classe, définit la structure commune d'un ensemble d'entités

- Propriété

atomiques

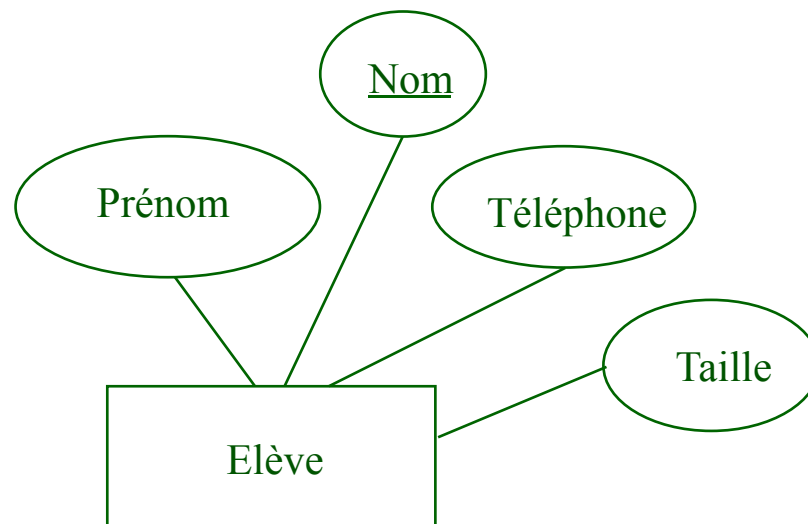
- Diagramme ER

- Type d'entité --> rectangle
- Propriété --> ovale



Propriétés identifiantes

- Ensemble de propriétés dont la valeur ne peut pas appartenir à plus qu'une entité
- Doit être fixé pour chaque type d'entité
- Soulignées dans le diagramme ER :



Associations

- Lien entre deux ou plusieurs types d'entité
- Diagramme ER : losange



- Type d'associations
 - définit des liens entre des types d'entité.
 - la valeur d'un type d'association peut être une table :
 - une colonne par type d'entité participant à l'association
 - une ligne pour chaque combinaison d'entités participant à une association

Elève	Cours
Jean	Macro Economie
Jean	Anglais
Jacques	E-Commerce

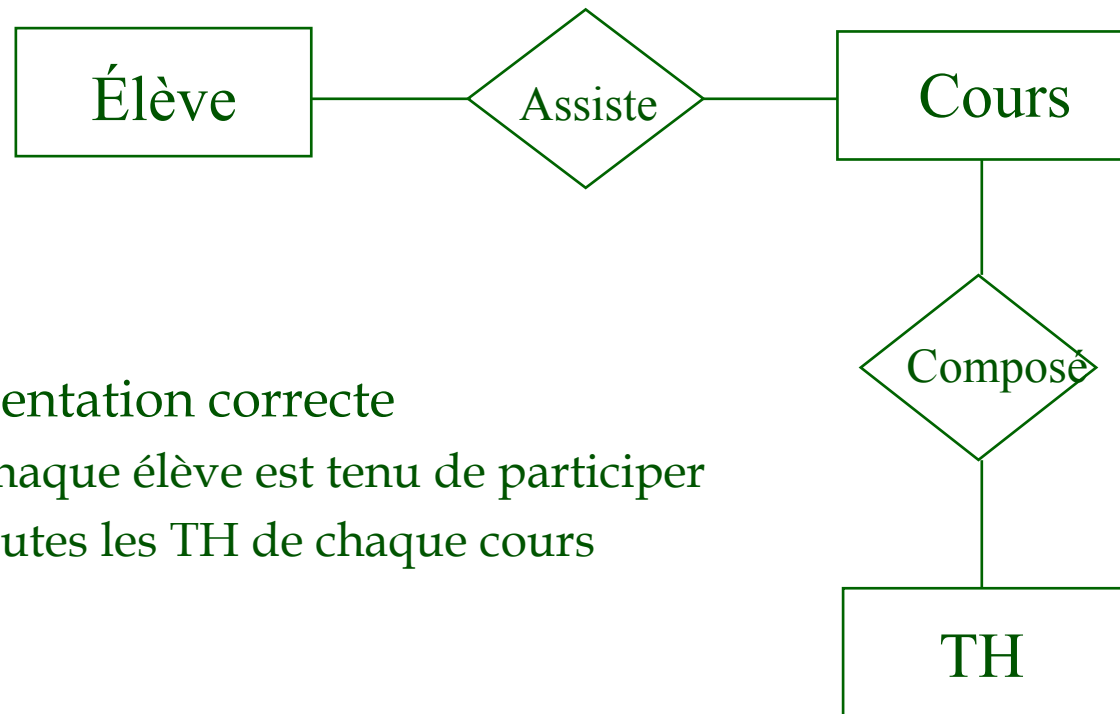
Cardinalité

- Min:Max
- Valeurs à considérer :
 - 0:1, 1:1
 - 0:N, 1:N,
 - M-N
 - Exemple :



Associations ternaires

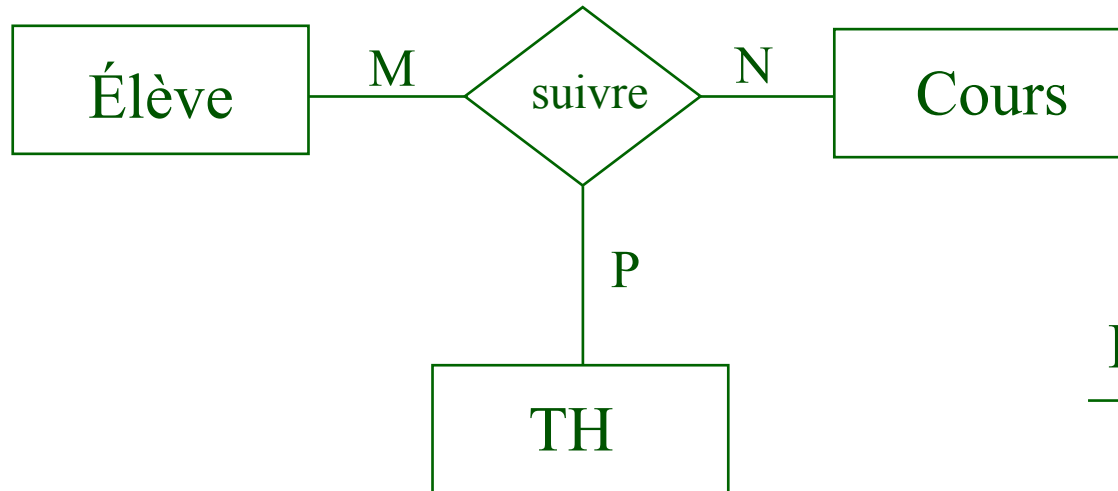
- Le plus souvent les associations binaires suffisent
 - Des associations entre trois types d'entité (ou plus) peuvent être nécessaires
 - Exemple : liens entre Elèves, Cours et TH



- Représentation correcte
 - si chaque élève est tenu de participer
 - a toutes les TH de chaque cours

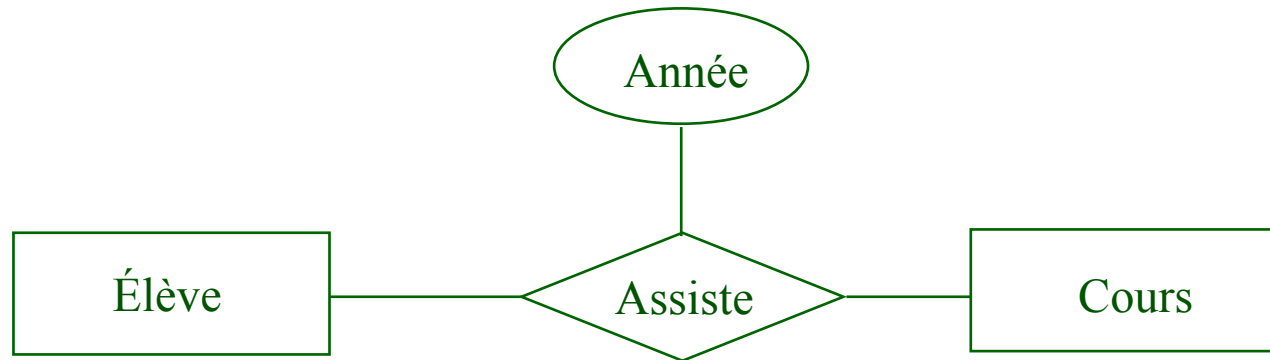
Exemple d'association ternaire

- Comment faire lorsque les élèves sont issus de différentes formations et ne suivent pas tous l'intégralité de chaque cours ?
 - Comment déterminer l'emploi du temps de chaque élève
 - ---> Besoin d'une association ternaire entre Élève, Cours et TH :

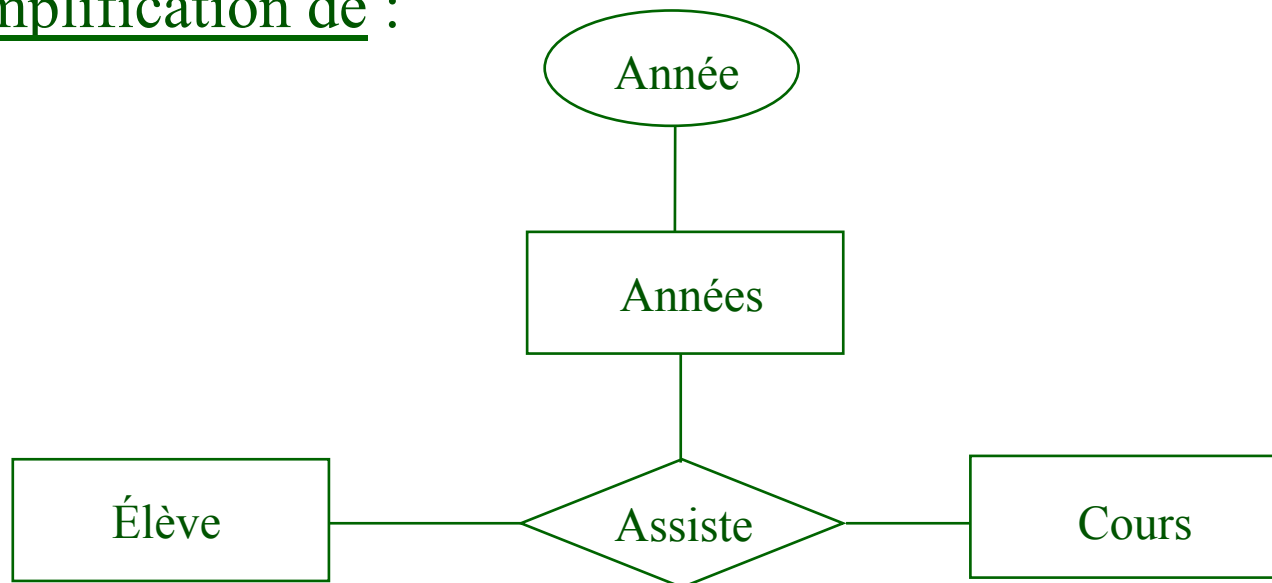


Élève	Cours	TH
Jean	Macro Economie	s40#1
Jean	Anglais	s40#2
Jacques	E-Commerce	s40#1
...		

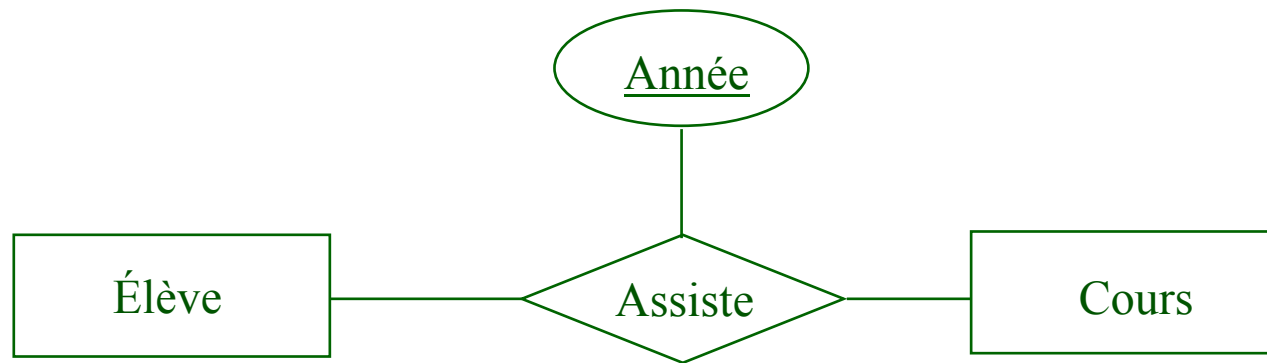
Propriétés des associations



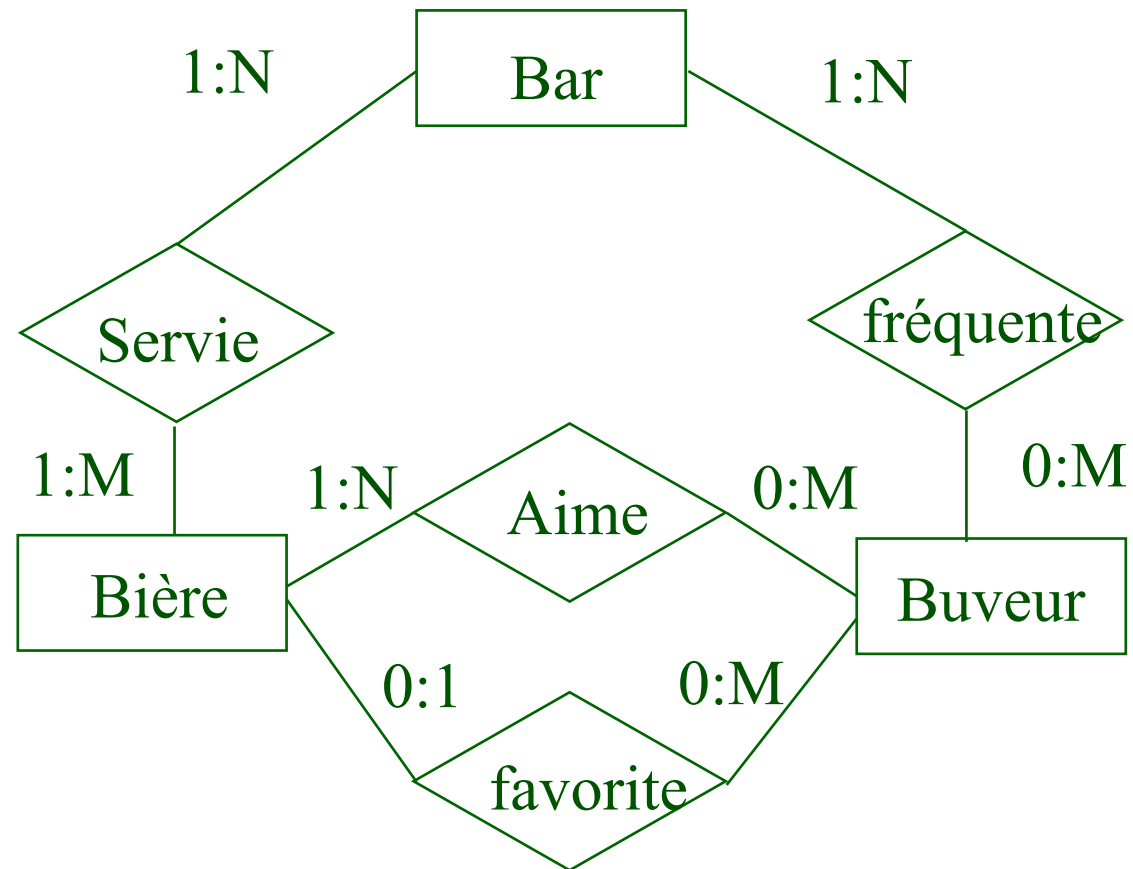
Simplification de :



Propriété multivaluée

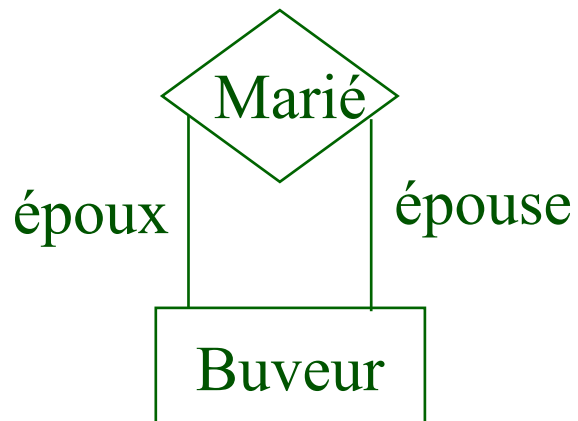


Exemple : Bars à Bière



Rôles

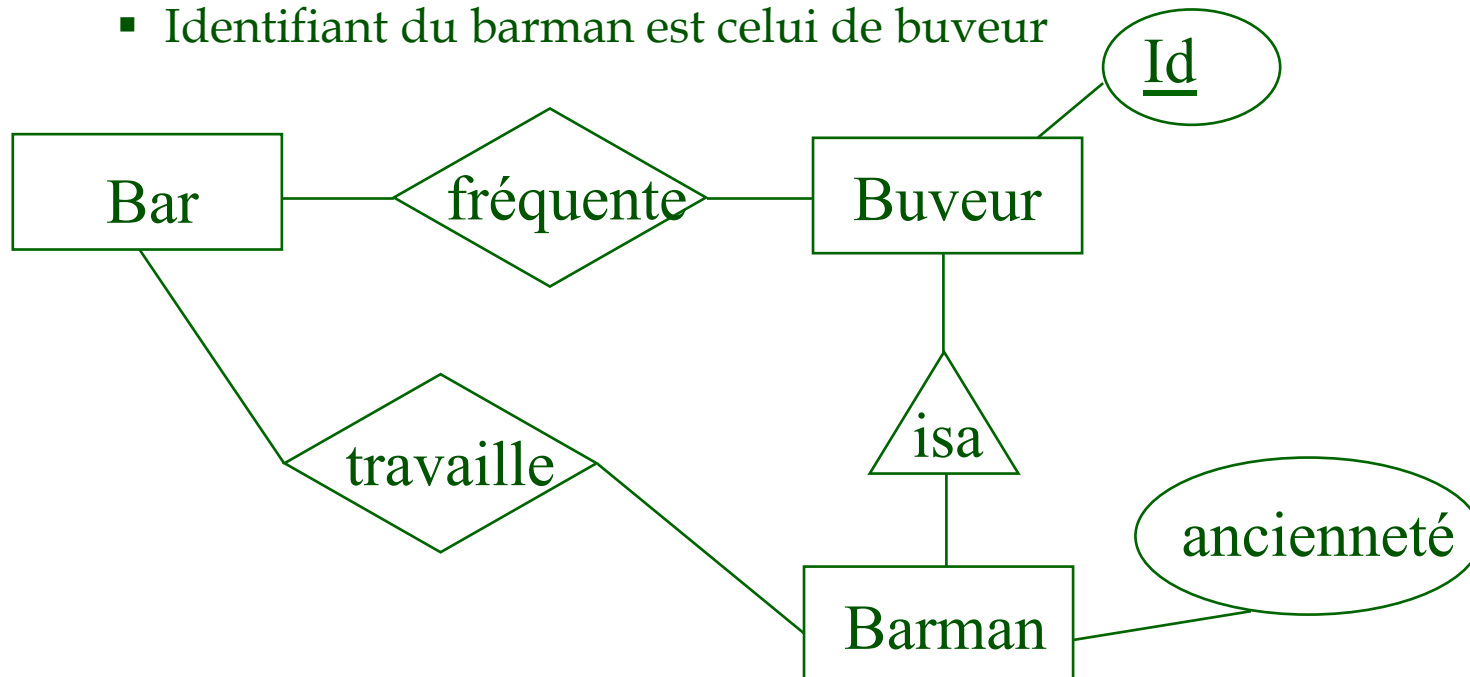
- Une entité peut participer plus qu'une fois à la même association
 - besoin d'associer des rôles différents pour une entité



époux	épouse
Jean	Bernadette
Jules	Arlette

Sous-classe

- Sous-classe --> nombre d' entités restreint --> propriétés supplémentaires
 - Exemple : Barman est un buveur
 - + (1 propriété supplémentaire « ancienneté » et un lien « travaille »)
 - Identifiant du barman est celui de buveur



Sous-classe et héritage

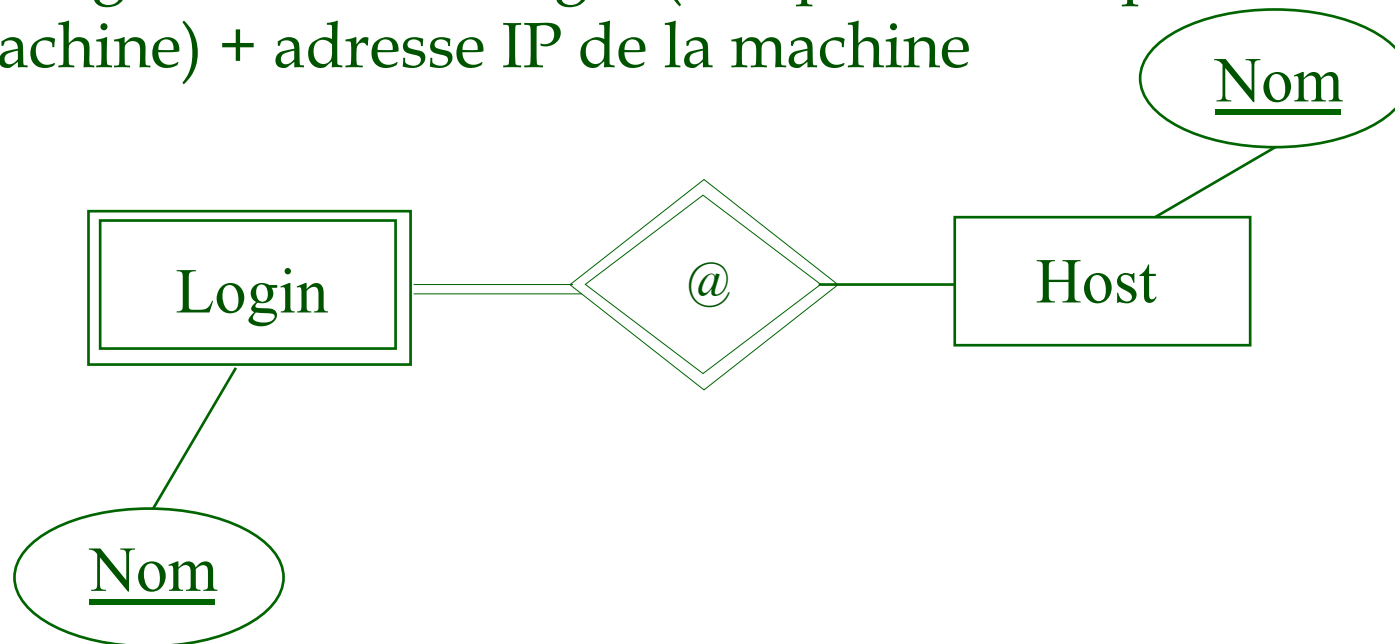
- Point de vue Objet :
 - 1 objet (entité) est associé à une seule classe
 - Il hérite les propriétés de sa super-classe
- Point de vue Entité/ Association :
 - 1 entité a une représentation dans toutes les sous-classe dont elle dépend
 - Ses propriétés sont obtenues par l'union des propriétés qu'elle a dans les différentes classes
- Héritage multiple :
 - Théoriquement possible
 - Problème résolution des conflits ?
- Contraintes COVER/OVERLAPS

Entité Faible (*weak entity*)

- L'identifiant d'une entité peut être composé à partir d'une (ou plusieurs) entité « forte » avec laquelle l'entité faible est liée

- Exemple :

Id Login = Nom de login (unique sur chaque machine) + adresse IP de la machine



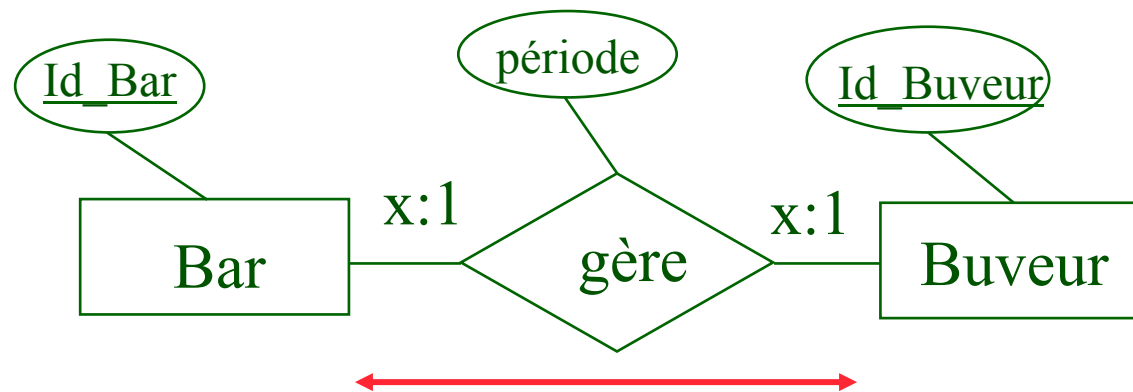
D' autres aspects...

- Choix attribut ou entité?
 - Exemple: plusieurs adresses
- Relation vs entité:
 - Exemple: **manager** (association avec un attribut **budget**)
 - budget global?

Transposition en relationnel

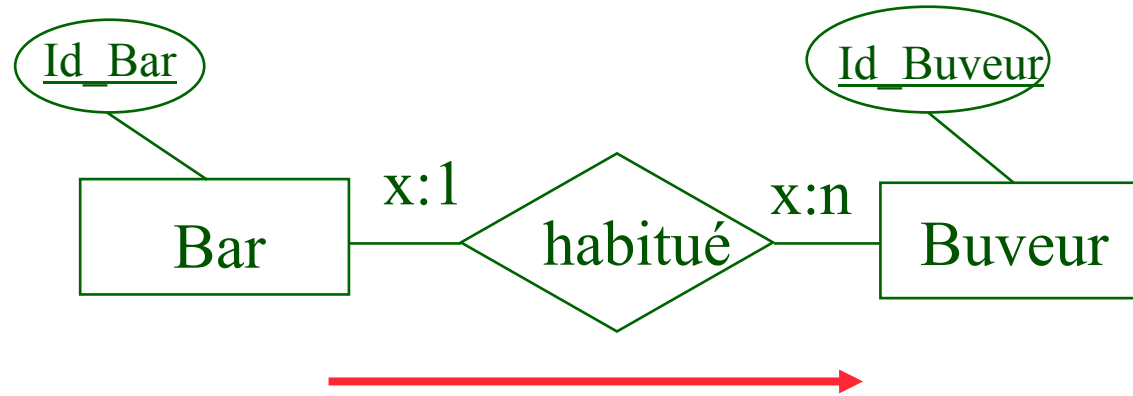
- Type d'entité ==> Relation
- Propriété ==> Attribut
- Propriétés identifiantes ==> Clé primaire
- Association ==> fonction des cardinalités ?

Transposition des associations



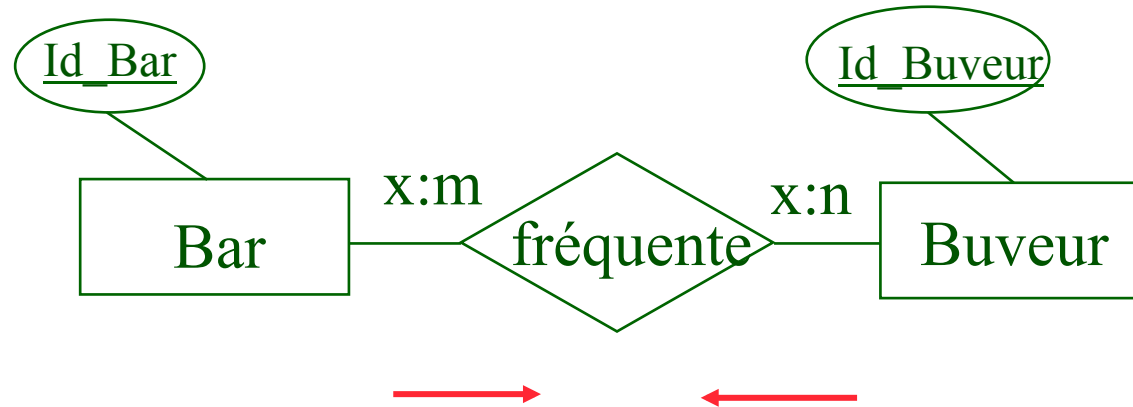
- L'identifiant de Buveur est utilisé comme clé étrangère dans la relation Bar, ou inversement.
- Les propriétés de l'association suivent la clé étrangère
Bar(Id_Bar, ...) et Buveur(Id_Buveur, ..., Id_Bar, période)
ou
Bar(Id_Bar, ..., Id_Gérant, période) et Buveur(Id_Buveur, ...)

Transposition des associations (suite)



- L'identifiant de Bar est utilisé comme clé étrangère dans la relation Buveur.
- Les propriétés de l'association suivent la clé étrangère
Bar(Id_Bar, ...) et Buveur(Id_Buveur, ..., Id_Bar)

Transposition des associations (suite)

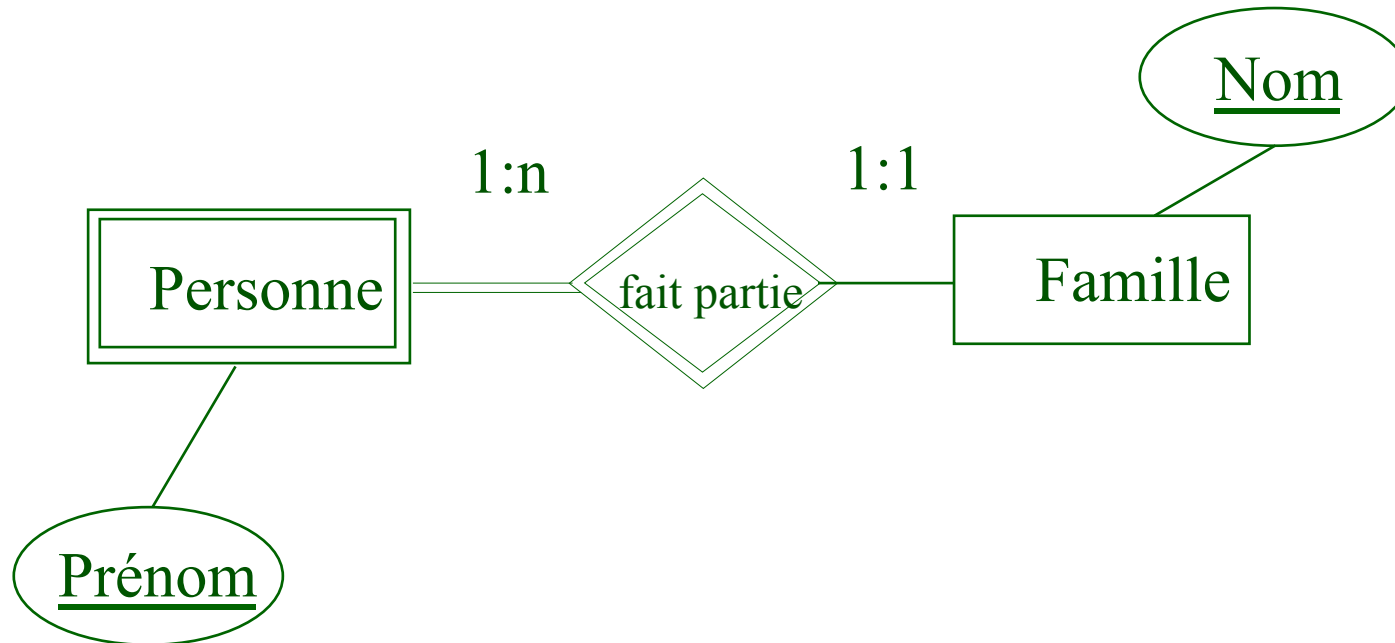


- L'association donne lieu à une relation.
- La clé primaire de la relation fréquente est composée des clés des relations Buveur et Bar.

Bar(Id_Bar, ...) et Buveur(Id_Buveur, ...)

Fréquente (Id_Bar, Id_Buveur, ...)

Transposition des associations (suite)

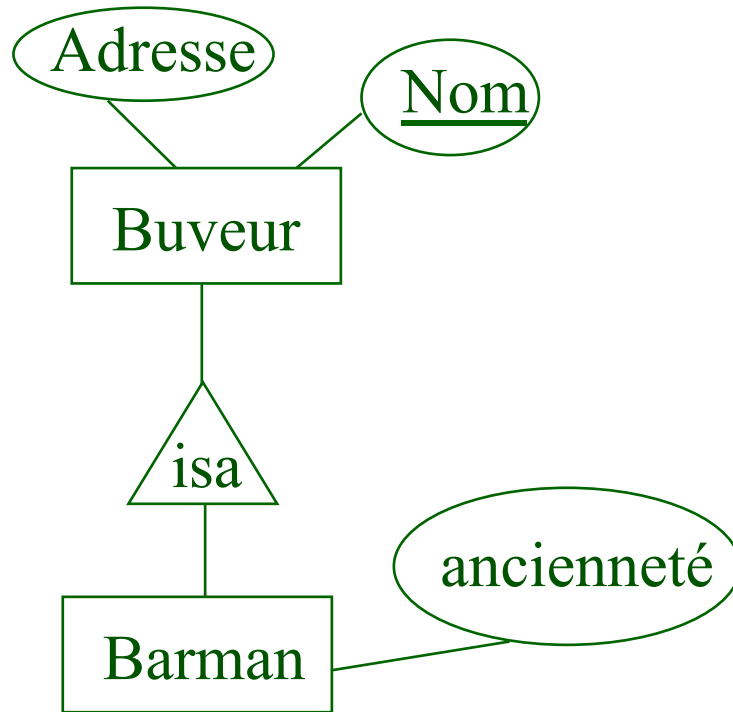


- La clé de la relation Personne est composée de celle de la relation Famille

Personne (Nom famille, Prénom, ...)

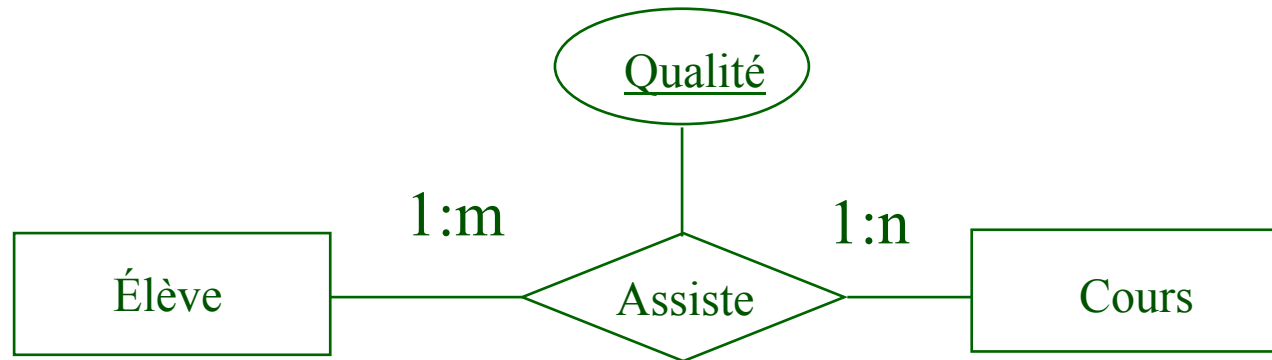
Famille (Nom, ...)

Transposition des associations (suite)



- Point de vue OO :
Buveur(Nom, Adresse) et Barman (Nom, Adresse, ancienneté)
- Point de vue ER
Buveur(Nom, Adresse) et Barman (Nom, ancienneté)

Transposition des associations (suite)

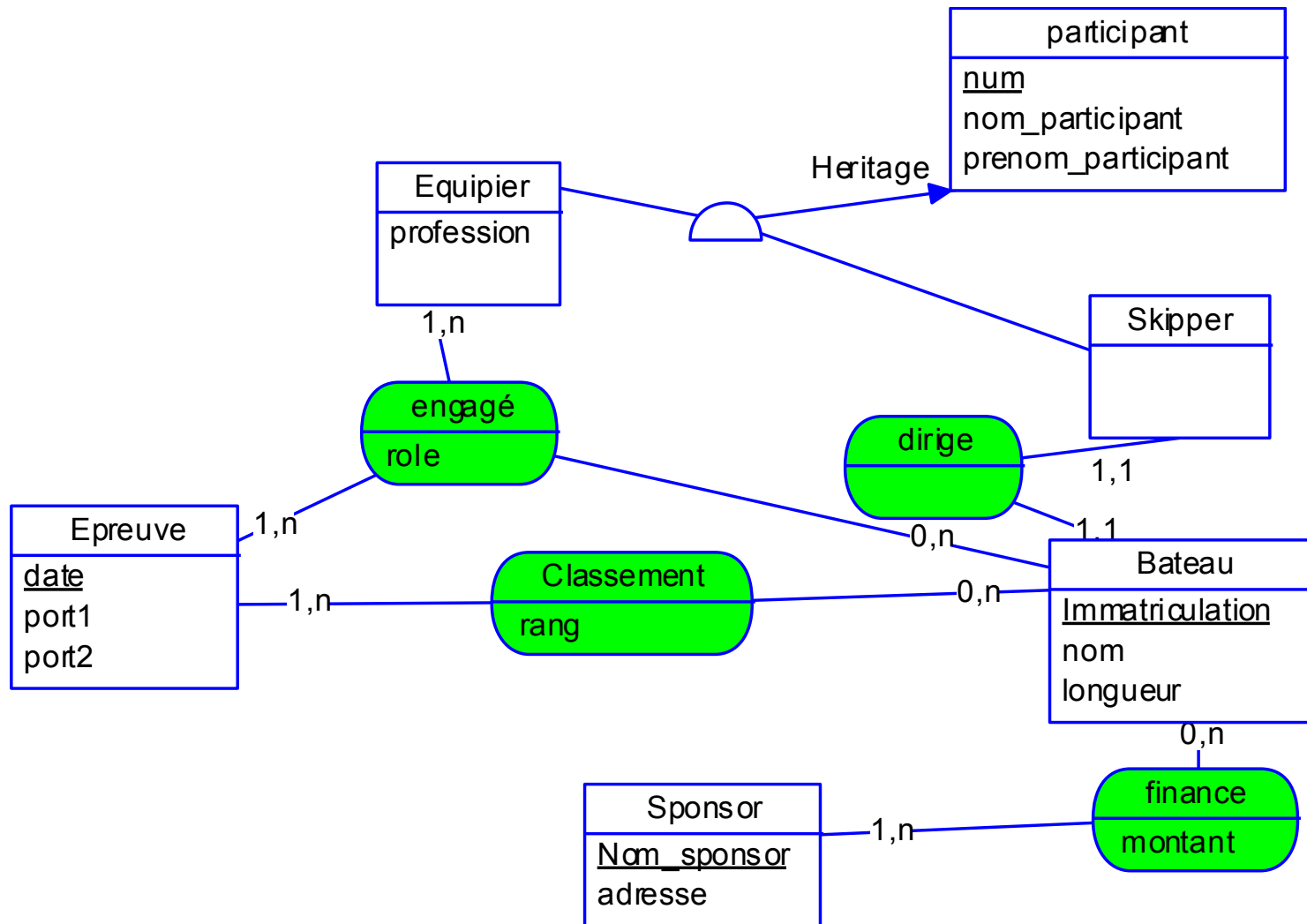


- L'attribut multivalué fait partie de la clé
Élève (Nom, ...)
Cours (Code, ...)
Assiste (Nom, Code cours, Qualité)

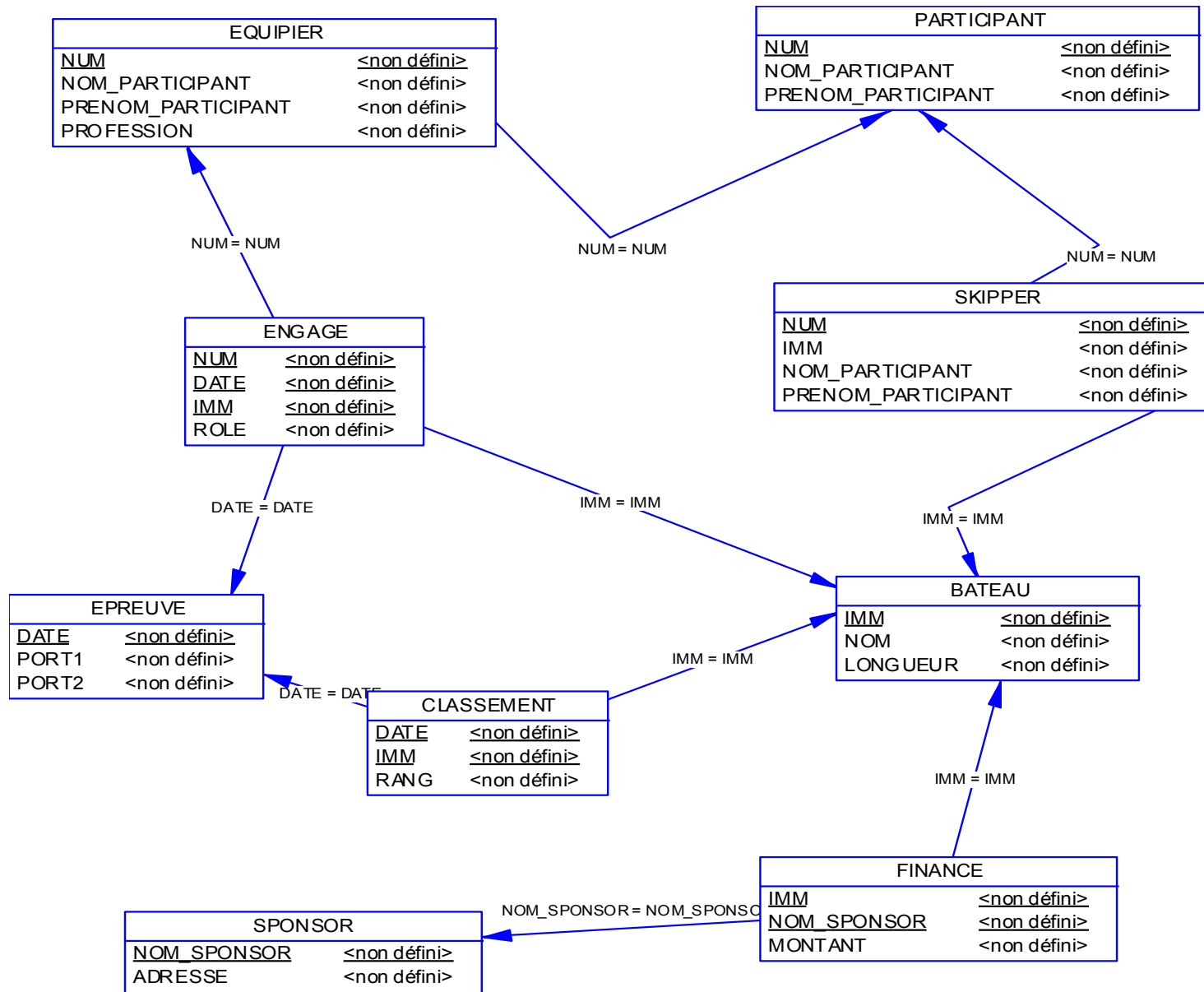
Conclusion

- Modélisation simple du monde réel
- Transposition relationnelle automatisable
- Outils d'aide à la conception E/A disponibles sur le marché

TD : base Speedo Finn



TD : base Speedo Finn, schéma relationnel



TD : base Speedo Finn, script SQL de création de la BD

```
=====
-- Nom de la base : MODELE_1
-- Nom de SGBD : ORACLE Version 8
-- Date de cr,ation : 07/11/2002 19:40
=====
-- Table : EPREUVE
=====
create table EPREUVE
(
  DATE          CHAR(10)      not null,
  PORT1         CHAR(10)      null ,
  PORT2         CHAR(10)      null ,
  constraint PK_EPREUVE primary key (DATE)
)
/
=====
-- Table : BATEAU
=====
create table BATEAU
(
  IMM           CHAR(10)      not null,
  NOM           CHAR(10)      null ,
  LONGUEUR     CHAR(10)      null ,
  constraint PK_BATEAU primary key (IMM)
)
/
=====
-- Table : SPONSOR
=====
create table SPONSOR
(
  NOM_SPONSOR  CHAR(10)      not null,
  ADRESSE      CHAR(10)      null ,
  constraint PK_SPONSOR primary key (NOM_SPONSOR)
)
/
```

```
=====
-- Table : PARTICIPANT
=====
create table PARTICIPANT
(
  NUM          CHAR(10)      not null,
  NOM_PARTICIPANT CHAR(10)    null ,
  PRENOM_PARTICIPANT CHAR(10) null ,
  constraint PK_PARTICIPANT primary key (NUM)
)
/
=====
-- Table : EQUIPIER
=====
create table EQUIPIER
(
  NUM          CHAR(10)      not null,
  NOM_PARTICIPANT CHAR(10)    null ,
  PRENOM_PARTICIPANT CHAR(10) null ,
  PROFESSION   CHAR(10)      null ,
  constraint PK_EQUIPIER primary key (NUM)
)
/
=====
-- Table : SKIPPER
=====
create table SKIPPER
(
  NUM          CHAR(10)      not null,
  IMM          CHAR(10)      not null,
  NOM_PARTICIPANT CHAR(10)    null ,
  PRENOM_PARTICIPANT CHAR(10) null ,
  constraint PK_SKIPPER primary key (NUM)
)
/
```

```

-- =====
-- Table : CLASSEMENT
-- =====
create table CLASSEMENT
(
  DATE          CHAR(10)      not null,
  IMM           CHAR(10)      not null,
  RANG          CHAR(10)      null   ,
  constraint PK_CLASSEMENT primary key (DATE, IMM)
)
/

-- =====
-- Table : FINANCE
-- =====
create table FINANCE
(
  IMM           CHAR(10)      not null,
  NOM_SPONSOR   CHAR(10)      not null,
  MONTANT       CHAR(10)      null   ,
  constraint PK_FINANCE primary key (IMM, NOM_SPONSOR)
)
/

-- =====
-- Table : ENGAGE
-- =====
create table ENGAGE
(
  NUM           CHAR(10)      not null,
  DATE          CHAR(10)      not null,
  IMM           CHAR(10)      not null,
  ROLE          CHAR(10)      null   ,
  constraint PK_ENGAGE primary key (NUM, DATE, IMM)
)
/

```

```

alter table EQUIPIER
  add constraint FK_EQUIPIER_HERITAGE2_PARTICIP foreign key (NUM)
  references PARTICIPANT (NUM)
/

alter table SKIPPER
  add constraint FK_SKIPPER_DIRIGE2_BATEAU foreign key (IMM)
  references BATEAU (IMM)
/

alter table SKIPPER
  add constraint FK_SKIPPER_HERITAGE_PARTICIP foreign key (NUM)
  references PARTICIPANT (NUM)
/

alter table CLASSEMENT
  add constraint FK_CLASSEME_LIEN_15_EPREUVE foreign key (DATE)
  references EPREUVE (DATE)
/

alter table CLASSEMENT
  add constraint FK_CLASSEME_LIEN_16_BATEAU foreign key (IMM)
  references BATEAU (IMM)
/

alter table FINANCE
  add constraint FK_FINANCE_LIEN_24_BATEAU foreign key (IMM)
  references BATEAU (IMM)
/

alter table FINANCE
  add constraint FK_FINANCE_LIEN_25_SPONSOR foreign key (NOM_SPONSOR)
  references SPONSOR (NOM_SPONSOR)
/

alter table ENGAGE
  add constraint FK_ENGAGE_LIEN_47_EQUIPIER foreign key (NUM)
  references EQUIPIER (NUM)
/

alter table ENGAGE
  add constraint FK_ENGAGE_LIEN_48_EPREUVE foreign key (DATE)
  references EPREUVE (DATE)
/

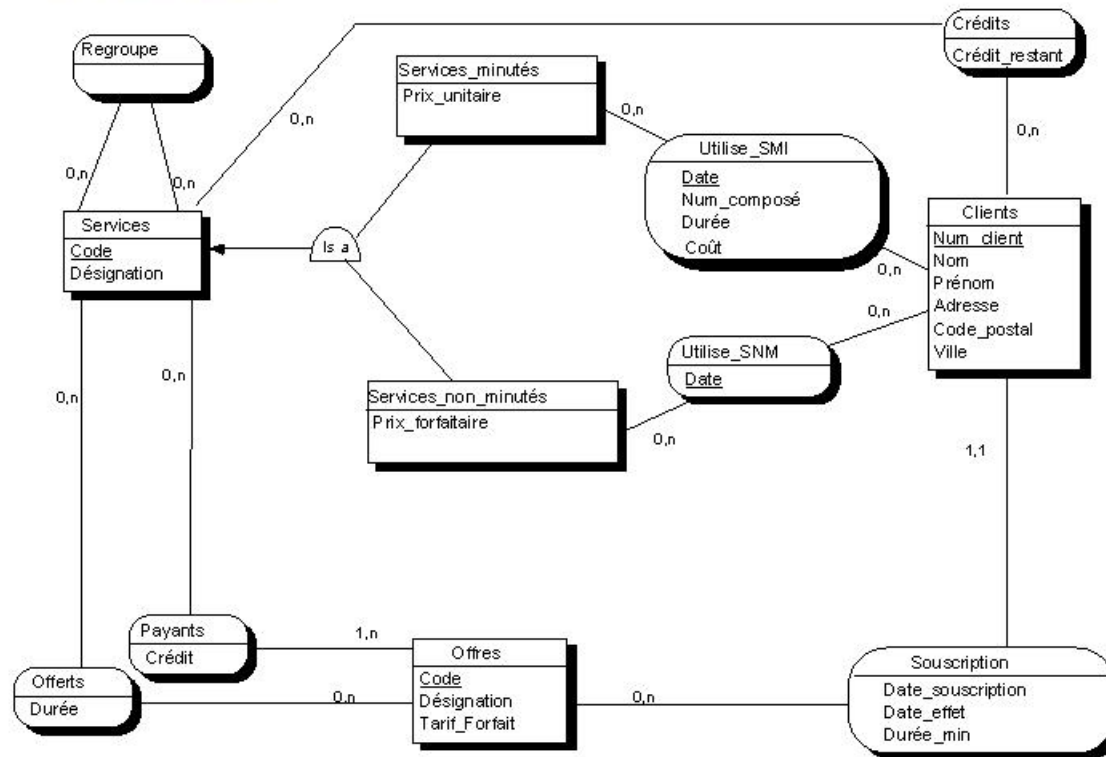
alter table ENGAGE
  add constraint FK_ENGAGE_LIEN_49_BATEAU foreign key (IMM)
  references BATEAU (IMM)
/

```

TD : base Xtélécom

Partie I : Conception de BD et SQL

Diagramme E/A :



TD : base Xtélécom, schéma relationnel

Partie I : Conception de BD et SQL

Schéma relationnel :

CLIENTS (NUM_CLIENT, NOM, PRENOM, ADRESSE, CODE_POSTAL, VILLE, CODE_OFFRE,
DATE_SOUSCRIPTION, DATE_EFFET, DUREE_MIN) - 2NF -

OFFRES (CODE, TARIF_FORFAIT, DESIGNATION)

OFFERTS (CODE_OFFRE, CODE_SERVICE, DUREE)

PAYANTS (CODE_OFFRE, CODE_SERVICE, CREDIT)

SERVICES (CODE, DESIGNATION)

REGROUPE (CODE_SG, CODE_SE)

SERVICES_MINUTES (CODE, PRIX_UNITAIRE)

SERVICES_NON_MINUTES (CODE, PRIX_FORFATAIRE)

UTILISE_SMI (CODE_SERVICE, NUM_CLIENT, DATE, NUM_COMPOSE, DUREE, COUT)

UTILISE_SNM (CODE_SERVICE, NUM_CLIENT, DATE)

CREDITS (CODE_SERVICE, NUM_CLIENT, CREDIT_RESTANT)

Ce schéma n'est pas en 3NF car la relation CLIENTS n'est pas en 3NF.

On a la DF suivante : Adresse, Ville ----> Code_Postal donc CLIENTS est 2NF et non 3NF

Ce schéma est donc en 2NF