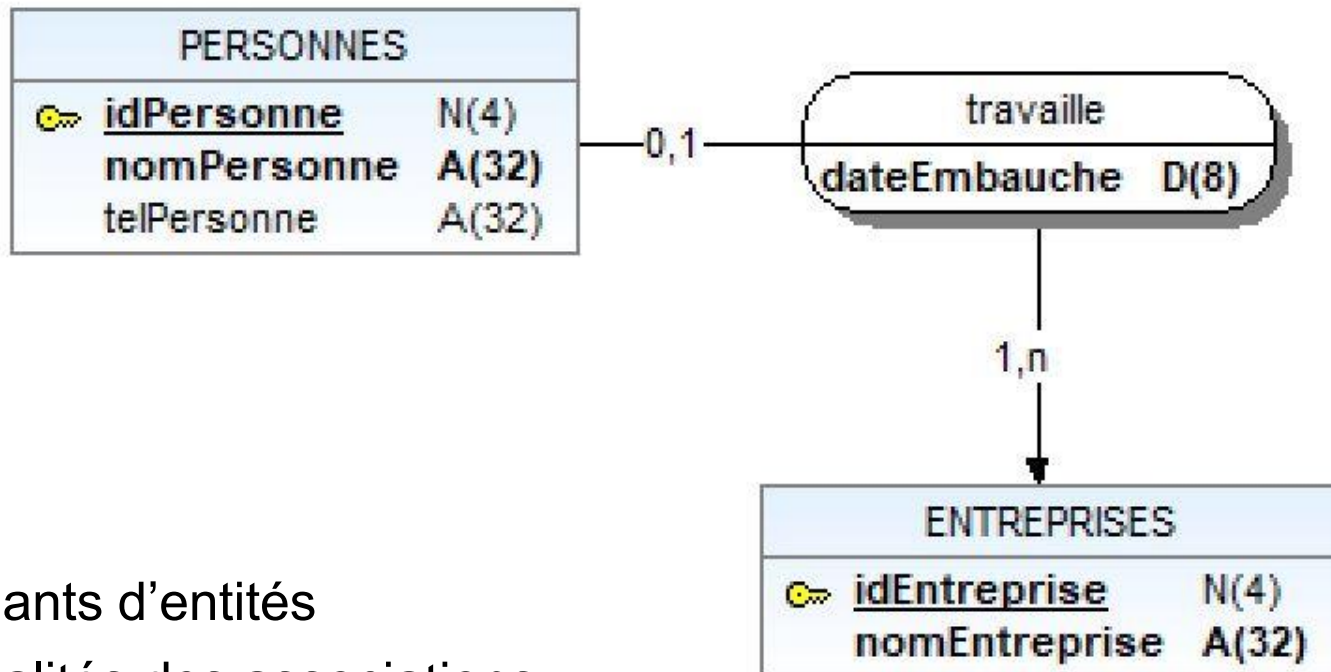


# **Quelques rappels du semestre 1...**

# Schéma entités-associations

Il s'agit du **modèle conceptuel de données** :

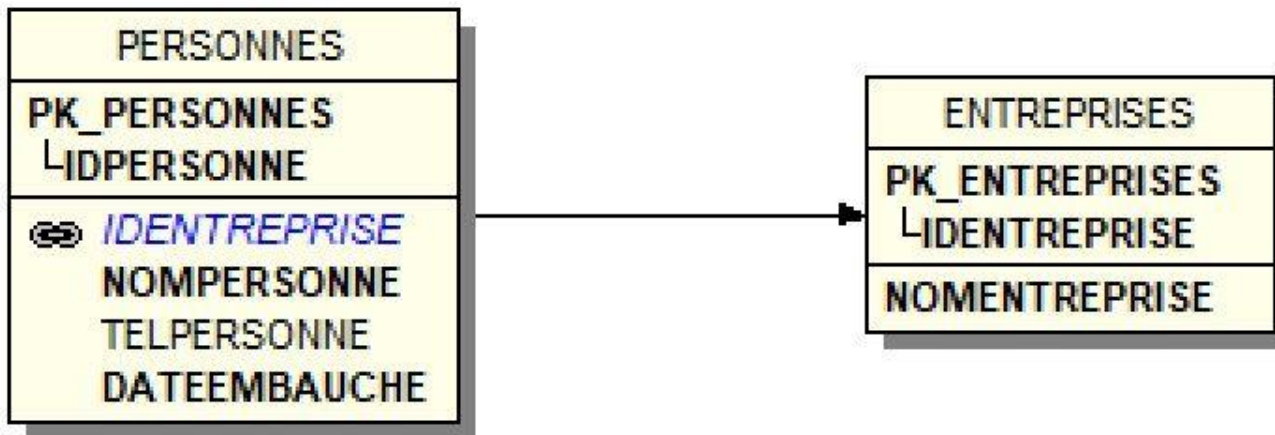


- Identifiants d'entités
- Cardinalités des associations
- Attributs obligatoires (non NULL)

# Schéma relationnel

---

Il s'agit cette fois du **modèle logique de données** :



- Clés primaires
- Clés étrangères
- Contraintes de référence

On utilise le **langage de définition de données (LDD)**.

Exemple précédent au format SQL Server :

```
use master /* pour être "au bon niveau" */
```

```
/* suppression de l'ancienne base */
```

```
drop database MABASE
```

```
go
```

```
/* création de la base */
```

```
create database MABASE
```

```
go
```

```
/* utilisation de la base */
```

```
use MABASE
```

```
go
```

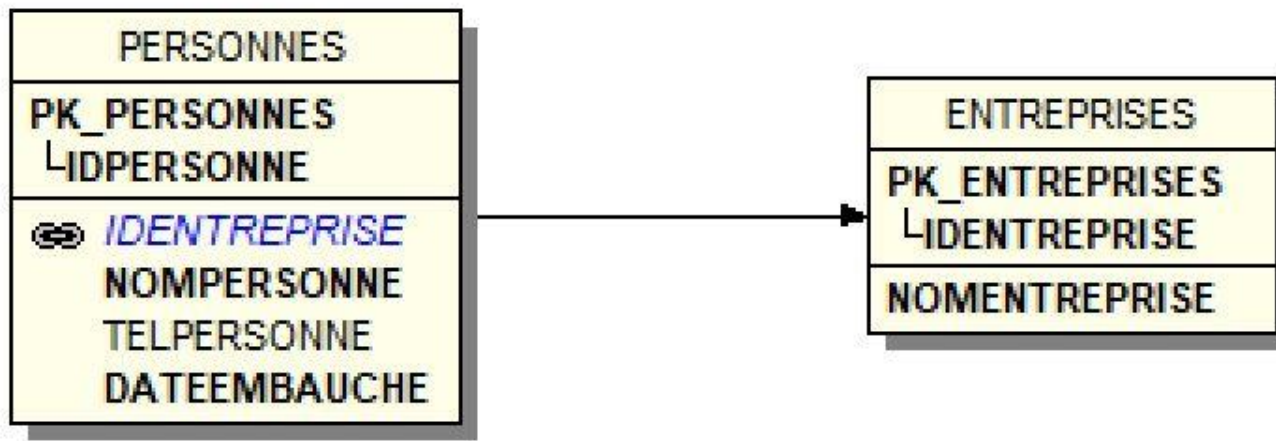
Définition des tables :

```
create table PERSONNES
(  IDPERSONNE int identity (1, 1),
  IDENTREPRISE int null,
  NOMPERSONNE char(32) not null,
  TELPERSONNE char(32) null,
  DATEEMBAUCHE datetime not null,
  constraint PK_PERSONNES primary key (IDPERSONNE)
)
go

create table ENTREPRISES
(  IDENTREPRISE int identity (1, 1),
  NOMENTREPRISE char(32) not null,
  constraint PK_ENTREPRISES primary key (IDENTREPRISE)
)
go
```

Contraintes de référence (clés étrangères) :

```
alter table PERSONNES  
  add constraint FK_PERSONNES_ENTREPRISES  
    foreign key (IDENTREPRISE)  
    references ENTREPRISES (IDENTREPRISE)  
  
go
```



Format général de la **clause SELECT** :

```
SELECT [ALL | DISTINCT] <liste d'attributs> | *  
FROM <liste de tables>  
[ WHERE <condition>]  
[ GROUP BY <liste d'expressions> [HAVING <condition>]  
[ ORDER BY <liste ordre> ]
```

## Equi-jointure :

```
SELECT ...  
FROM T1 INNER JOIN T2 ON T1.attr1 = T2.attr2  
    [ INNER JOIN T3 ON ... ]
```

## Jointures externes :

```
SELECT ...  
FROM T1 ( LEFT | RIGHT | FULL ) OUTER JOIN T2  
    ON T1.attr1 = T2.attr2  
...
```



# Opérations ensemblistes (SQL)

---

Union, intersection ou différence (les deux `SELECT` doivent produire des tables « de même nature ») :

```
SELECT ...
```

```
...
```

```
[ UNION | INTERSECT | EXCEPT ]
```

```
SELECT ...
```

```
...
```

# Requêtes de mise à jour (SQL)

---

Ajout d'une ou plusieurs lignes :

```
INSERT INTO <table> ( <liste d'attributs> )  
VALUES ( <liste d'expressions> )
```

```
INSERT INTO <table> ( <liste d'attributs> )  
    ( SELECT ... )
```

Mise à jour d'une ou plusieurs lignes :

```
UPDATE <table> SET <attribut> = <expression>, ...  
[ WHERE ... ]
```

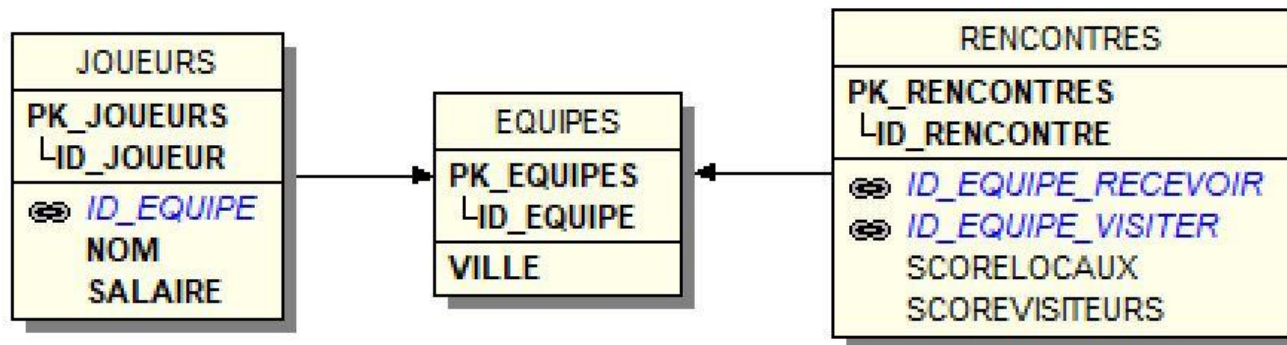
Suppression d'une ou plusieurs lignes :

```
DELETE FROM <table>  
[ WHERE ... ]
```

# Un petit exercice pour finir...

En utilisant **SQL Management Studio** :

1. Préparer un script `sql` permettant, sur `info-joyeux` (authentification `windows`), de créer une base `testVotrenom` contenant les trois tables suivantes :



2. Remplir votre base avec quelques données.
3. Afficher la masse salariale (somme des salaires) de chaque équipe.
4. Afficher le nombre de victoires à domicile, puis le nombre de victoires à l'extérieur, puis le nombre total de victoires, de chaque équipe (ce nombre peut être 0...).