203 Les courses de chevaux

# Récolte des besoins

Un parieur assidu des champs de courses et des bases de données, voudrait mémoriser, dans une base de données

relationnelle, les courses de chevaux, les paris qu'il a faits et les résultats.

Plus précisément, il veut enregistrer les informations suivantes pour chaque course :

- Le nom et la date (ex : Prix d'Amérique, 21-07-92)

- Le numéro, le nom chevaux partants

- Ses paris, avec pour chacun :

o Le type de pari (couplé, tiercé, quarté, quinté...).

▪ Couplé = Pari sur 2 chevaux (1er et 2nd)

▪ Tiercé = Pari sur 3 chevaux (1er, 2ème et 3ème)

▪ Etc...

o La somme jouée et les numéros de chevaux dans l'ordre du pari.

Une fois la course jouée, on enregistre aussi :

- Le résultat : l'ordre d'arrivée des chevaux.

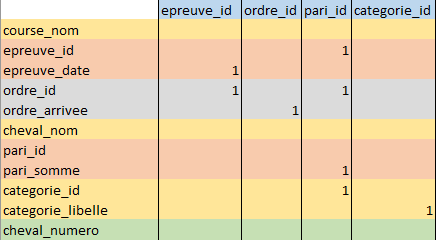
- Le gain total du parieur pour la course (couplé = somme jouée x 2, tiercé = somme jouée \* 3, etc...).

Les noms de chevaux sont uniques et les noms de courses sont uniques (à un instant donné)

# Dictionnaire des données

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entité | Mnémonique | Signification | Type(longueur) | Contraintes |
| Course | course\_nom | nom de la course | VARCHAR(50) | identifiant |
| Epreuve | epreuve\_id | identifiant de l’épreuve | INT(11) | identifiant, ai |
|  | epreuve\_date | date de l’épreuve | DATE | obligatoire |
| Ordre | ordre\_id | identifiant de l’ordre d’arrivée | INT(11) | identifiant,ai |
|  | ordre\_arrivee | ordre d’arrivée des chevaux | VARCHAR(150) | obligatoire |
| Cheval | cheval\_nom | nom du cheval | VARCHAR(50) | identifiant |
| Pari | pari\_id | identifiant du pari | INT(11) | obligatoire, ai |
|  | pari\_somme | somme jouée pour le pari | DECIMAL(10,2) | obligatoire |
| Categorie | categorie\_id | identifiant de la catégorie de pari | INT(11) | identifiant |
|  | categorie\_libelle | libellé de la catégorie de pari (couplé,tiercé…) | VARCHAR(50) | obligatoire |
|  |  |  |  |  |
|  | cheval\_numero | numéro du cheval | TINYINT | obligatoire, unique |

# Matrice



# Dépendances fonctionnelles

epreuve\_id -> epreuve\_date, #ordre\_id

ordre\_id -> ordre\_arrivee

pari\_id -> pari\_somme, ordre\_id, #epreuve\_id, #categorie\_id

categorie \_id -> categorie \_libelle

epreuve\_id, cheval\_nom -> cheval\_numero

# Règles de gestion

Une course est composée de 1 à plusieurs épreuves  
Une épreuve compose 1 seule course

Un cheval participe à 1 ou plusieurs épreuves  
A une épreuve participent 1 ou plusieurs chevaux

Un pari est fait pour 1 seule épreuve

Pour une épreuve on fait 1 ou plusieurs paris

A un pari correspond 1 seule categorie de pari  
A une categorie de pari correspondent 0 ou plusieurs paris

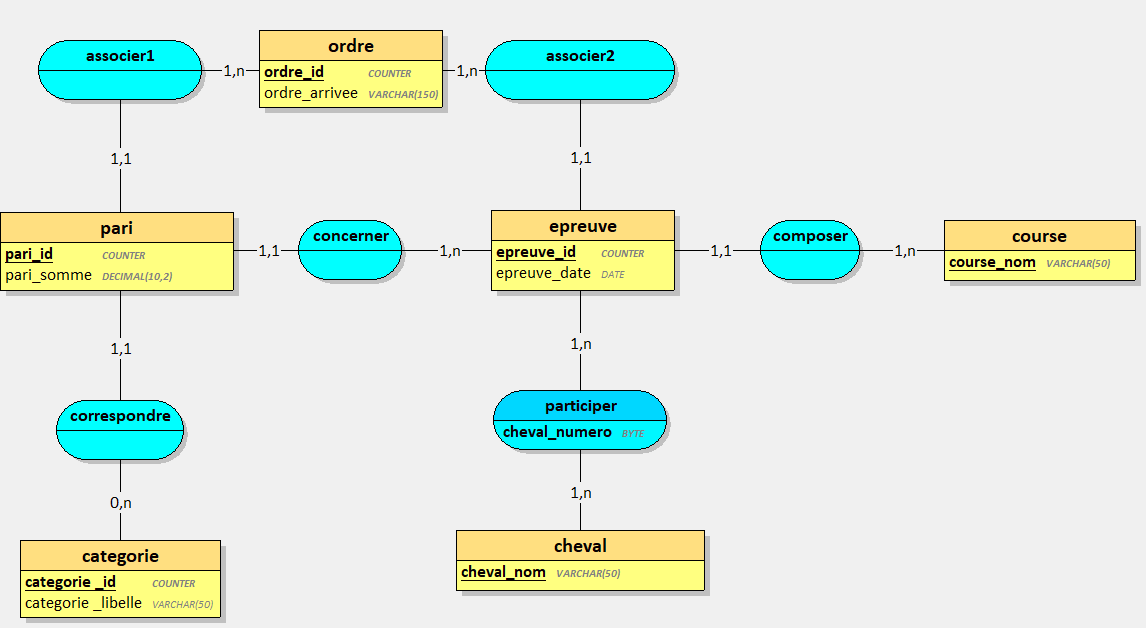
A un pari on associe 1 seul ordre d’arrivee

Un ordre d’arrivee est associé a 1 ou plusieurs paris

A une epreuve on associe 1 seul ordre d’arrivee

Un ordre d’arrivee est associé à 1 ou plusieurs épreuves

# Modèle conceptuel des données



# Modèle relationnel des données

ordre(ordre\_id, ordre\_arrivee)

course(course\_nom)

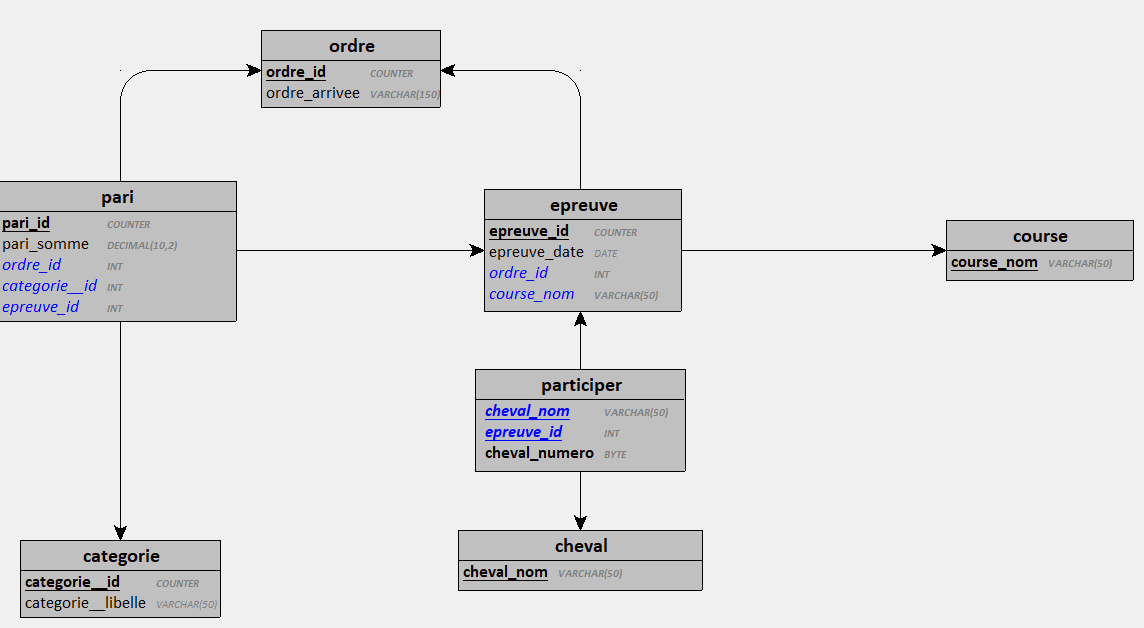
epreuve(epreuve\_id, epreuve\_date, #ordre\_id, #course\_nom)

categorie(categorie \_id, categorie \_libelle)

pari(pari\_id, pari\_somme, #ordre\_id, # categorie \_id, #epreuve\_id)

cheval(cheval\_nom)

epreuve\_cheval(#epreuve\_id, #cheval\_nom, cheval\_numero)



# Création de la base de données (DDL) SQL server

drop table if exists epreuve\_cheval;

drop table if exists cheval;

drop table if exists pari;

drop table if exists categorie;

drop table if exists course;

drop table if exists ordre;

create table ordre

(

ordre\_id int primary key identity(1,1)

,ordre\_arrivee varchar(150) not null

)

;

create table course

(

course\_nom varchar(50) primary key not null

)

;

create table epreuve

(

epreuve\_id int primary key identity(1,1)

)

;

create table categorie

(

categorie\_id int primary key identity(1,1)

,categorie\_libelle varchar(50) not null

)

;

create table pari

(

pari\_id int primary key identity(1,1)

,pari\_somme decimal(10,2) not null

,ordre\_id int

,categorie\_id int

,epreuve\_id int

,foreign key (ordre\_id) references ordre(ordre\_id)

,foreign key (categorie\_id) references categorie(categorie\_id)

,foreign key (epreuve\_id) references epreuve(epreuve\_id)

)

;

create table cheval

(

cheval\_nom varchar(50) primary key not null

)

;

create table epreuve\_cheval

(

epreuve\_id int

,cheval\_nom varchar(50)

,cheval\_numero tinyint not null unique

,primary key (epreuve\_id, cheval\_nom)

,foreign key (epreuve\_id) references epreuve(epreuve\_id)

,foreign key (cheval\_nom) references cheval(cheval\_nom)

)

;