

# TP 1 de projet BD : YOBA ROSTAND

## Exercices :

### I. Schéma relationnel

1. Donnons la clé primaire de chacune des tables de la base Zoo.

LesAnimaux => nomA

LesCages => noCage

LesResponsables => noAllée

LesMaladies => nomM

LesEmployés => nomE

LesGardiens => noCage

2. Donnez le schéma relationnel de la base Zoo.

LesAnimaux(nomA, sexe, type, pays, anNais, #noCage)

LesCages(noCage, fonction, #noAllée)

LesResponsables(noAllée, #nomE)

LesMaladies(#nomA, nomM)

LesEmployés(nomE, adresse)

LesGardiens(#noCage, #nomE)

### II. Compréhension des relations :

1. Indiquons les n-uplets que l'on doit insérer dans la ou les relation(s) concernée(s), pour représenter les informations contenues dans le texte ci-dessous :
  - Chloé, qui est au zoo depuis longtemps, vient de contracter une rage de dents en même temps que la grippe.
    - Insertion dans la table Maladie de deux tuples correspondants aux deux maladies
  - Cas de Brigitte
    - Insertion dans la table LesAnimaux
    - Insertion dans la table LesCages
  - Cas de Mme Bruandet
    - Insertion dans la table LesGardiens
    - Insertion dans la table LesCages
  - Dernier cas :
    - Insertion dans la table LesCages
- 2.

### III. Requêtes

Donnons l'expression SQL, ainsi que le résultat des requêtes données ci-dessous :

```

-- Requetes SQL
-- Utilisation de la Base de Données Zoo

-- 1. Nom des animaux du Zoo
select nomA
from LesAnimaux;

-- 2. Fonctionnalités disponibles dans le zoo.
select distinct fonction
from LesCages;

-- 3. les noms des léopards
select nomA
from LesAnimaux
where type='leopard';

-- 4. Les maladies contractées au moins une fois par des animaux du zoo.
select nomM, count(nomM)
from LesMaladies
GROUP BY (nomM)
HAVING count(nomM)>1;

-- 5. Les noms et Numéros de cage des animaux mâles qui sont originaires du
kenya et dont la date est antérieure à 1992
select nomA, noCage
from LesAnimaux
where (pays='Kenya'
and anNais='1992');

-- 6. requête produisant l'affichage suivant: Peyrin vit à Nouméa Burrut
vit à Sartène Sicard vit à Calvi Voiron vit à Pointe à Pitre Scholl vit à
ushuaia Adiba vit à Papeete
select concat (nomE, ' vit a ' ,adresse) as Location
from LesEmployes;

-- 7. Le nom et l'âge des animaux en 2012
select nomA, 2012-anNais as ageA
from LesAnimaux;

--8. Le nom des gardiens qui habitent Ushuaïas
select distinct nomE
from LesGardiens
where nomE IN
(select nomE
from LesEmployes
where adresse = 'Ushuaia');

-- 9. La fonctionnalité et le nom du gardien des cages gardées par un
employé habitant Calvi.
select distinct LesEmployes.nomE, LesCages.fonction
from LesCages, LesEmployes, LesGardiens
where LesEmployes.nomE = LesGardiens.nomE and LesGardiens.noCage =
LesCages.noCage
and LesEmployes.adresse = 'Calvi';

-- 10. Le nom des animaux ainsi que les employés qui en sont les gardiens
soit les responsables.
select LesEmployes.nomE, LesAnimaux.nomA
from LesAnimaux, LesEmployes

```

```

where LesAnimaux.noCage IN
(
select LesCages.noCage
from LesCages, LesGardiens
where (LesGardiens.noCage = LesAnimaux.noCage AND
LesGardiens.nomE = LesEmployes.nomE) OR (noAllee IN
(
select noAllee
from LesResponsables
where LesResponsables.nomE = LesEmployes.nomE
)
)
);

-- 11. Le nom des gardiens gardant tous les animaux
select distinct LesGardiens.nomE
from LesGardiens
left join LesAnimaux
on LesAnimaux.noCage = LesGardiens.noCage
GROUP BY LesGardiens.nomE
HAVING COUNT(*)=
(
select count(*)
from LesAnimaux
);

-- 12. Les noms et types des animaux qui n'ont jamais été malades.
select LesAnimaux.nomA, LesAnimaux.type
from LesAnimaux
where nomA NOT IN
(
select nomA
from LesMaladies
);

-- 13. Les noms des animaux originaire du Kenya ayant déjà contractés une
grippe.
select distinct LesAnimaux.nomA
from LesAnimaux
inner join LesMaladies
where LesAnimaux.pays='Kenya' and LesMaladies.nomM = 'grippe'
and LesAnimaux.nomA = LesMaladies.nomA;

-- 14. Les numéros et fonctionnalités des cages qui sont innocupées.
select LesCages.noCage, LesCages.fonction
from LesCages
where LesCages.noCage NOT IN
(
select LesAnimaux.noCage
from LesAnimaux
);

-- 15. Donner pour chaque animal mâle l'ensemble des maladies qu'il a
contractées (ensemble des couples nom d'animal, nom de maladie).
select LesAnimaux.nomA, LesMaladies.nomM
from LesAnimaux, LesMaladies
where LesAnimaux.sexe = 'male' AND LesAnimaux.nomA = LesMaladies.nomA;

-- 16. Les numéros et fonctionnalités des cages qui sont partagées par des
animaux de types différents. En d'autres termes, ce sont les cages qui
contiennent au moins deux animaux de types différents.

```

```

select distinct LesCages.noCage, LesCages.fonction
from LesCages
where LesCages.noCage IN
(
select LesAnimaux.noCage
from LesAnimaux
GROUP BY (LesAnimaux.noCage)
HAVING count(LesAnimaux.noCage)>1
);

```

-- 17. Les noms des responsables et les noms des gardiens de Charly.

```

select distinct LesEmployes.nomE
from LesResponsables, LesGardiens, LesCages, LesEmployes
where (LesResponsables.noAllee IN
(select LesCages.noAllee
from LesCages
where LesCages.noCage IN (
select LesAnimaux.noCage
from LesAnimaux
where LesAnimaux.nomA = 'Charly'
))) AND LesResponsables.nomE = LesEmployes.nomE OR LesGardiens.noCage IN
(
select LesAnimaux.noCage
from LesAnimaux
where LesAnimaux.nomA = 'Charly'
);

```

-- 18. Le nom et le pays d'origine de l'animal doyen du zoo (il peut y en avoir plusieurs).

```

select nomA, pays
from LesAnimaux
where anNais =
(
select MIN(anNais)
from LesAnimaux
);

```

-- 19. Le nom, le type et l'année de naissance des animaux qui ont contracté toutes les maladies (connues) du zoo.

```

select LesAnimaux.nomA, LesAnimaux.type, LesAnimaux.anNais
from LesAnimaux
inner join LesMaladies
on LesMaladies.nomA = LesAnimaux.nomA
GROUP BY LesMaladies.nomM
HAVING count(LesMaladies.nomM) =
(
select count(distinct LesMaladies.nomM)
from LesMaladies
);

```

-- 20. Le nom, le type et l'année de naissance des animaux qui partagent la cage de Charly.

```

select nomA, type, pays
from LesAnimaux
where LesAnimaux.noCage =
(
select noCage
from LesAnimaux
where LesAnimaux.nomA = 'Charly'
) AND LesAnimaux.nomA != 'Charly';

```

```
-- 21. Le nom et l'adresse des employés qui sont gardiens d'animaux de tous
types, on fait références aux types des animaux du zoo.
/* Cette requête n'est pas terminé */
select LesEmployes.nomE, LesEmployes.adresse
from LesEmployes
inner join LesGardiens
on LesEmployes.nomE = LesGardiens.nomE
inner join LesAnimaux
on LesAnimaux.noCage = LesGardiens.noCage
HAVING count(distinct LesAnimaux.nomA) =
(
select count(*)
from LesAnimaux
);
```