

## Base de données

Soit la base de données CIRQUE<sup>\*</sup> dont l'extension est donnée ci-dessous.

### PERSONNEL

NOM(20)	ROLE(20)
Clovis	Jongleur
Reine de May	Ecuyer
Louche	Clown
Attention	Equilibriste
Partition	Musicien
Crinière	Dompteur
Jerry	Clown
Bal	Jongleur
Final	Musicien
Louis	Jongleur
Léo	Jongleur
Lulu	Ecuyer
Coloquinte	Equilibriste
Groistas	Jongleur
Sangtrèspur	Dompteur

### NUMERO

TITRE(30)	NATURE(20)	RESPONSABLE(20)
Les Zoupalas	Jonglerie	Clovis
Le coche infernal	Equitation	Reine de May
Les fauves	Clownerie	Louche
Les Smilers	Equilibre	Attention
La passoire magique	Lion	Crinière
Les Zozos	Clownerie	Jerry
Les Tartarins	Jonglerie	Bal

### ACCESSOIRE

NOM(30)	COULEUR(10)	VOLUME(5,2)	RATELIER(2)	CAMION(1)
Ballon	Rouge	0.3	15	5
Barre	Blanc	0.6	19	5
Fouet	Marron	0.2	11	3
Bicyclette à éléphant	Vert	0.4	27	8
Trompette	Rouge	0.2	2	1
Cercle magique	Orange	0.2	1	1
Boule	Cristal	0.2	88	8
Cage à lions	Noir	10.0	0	2
Chaise longue de lion	Bleu	0.9	11	5
Peigne de chimpanzé	Jaune	0.2	23	3
Etrier	NULL	NULL	NULL	NULL

### UTILISATION

TITRE(30)	UTILISATEUR(20)	ACCESSOIRE(30)
Les Zoupalas	Louis	Ballon
Les Zoupalas	Léo	Ballon
Les Zoupalas	Louis	Barre
Le coche infernal	Groistas	Bicyclette à éléphant
Le coche infernal	Lulu	Fouet
Les fauves	Jerry	Trompette
Les Smilers	Attention	Cercle magique
Les Smilers	Attention	Boule
Les Smilers	Coloquinte	Bicyclette à éléphant
La passoire magique	Crinière	Cage à lions
La passoire magique	Crinière	Chaise longue de lion
Les Zozos	Jerry	Bicyclette à éléphant
Les Zozos	Jerry	Peigne de chimpanzé
Les Tartarins	Groistas	Bicyclette à éléphant
Le coche infernal	Sangtrèspur	Etrier

- Créer la structure des quatre tables de la base en intégrant les contraintes d'intégrité nécessaires (4 clés primaires, 4 clés étrangères).
- Remplir les tables PERSONNEL et NUMERO en utilisant des requêtes paramétrées. Recopier les données avec la plus grande fidélité (attention aux majuscules/minuscules).
- Remplir les tables ACCESSOIRE et UTILISATION en recopiant le contenu des tables DARMONT.ACCESSOIRE et DARMONT.UTILISATION. Pour cela, utiliser une requête du type `INSERT INTO table_destination SELECT * FROM table_source;`

## Requêtes

### Projections

- Nature de tous les numéros sans suppression des doubles.
- Nature de tous les numéros avec suppression des doubles.
- Camions et volumes des accessoires, par ordre croissant des camions et, pour chaque camion, par ordre décroissant des volumes.

### Restrictions

- Accessoires de volume supérieur ou égal à 0,3 ou de couleur rouge.
- Accessoires de volume compris entre 0,3 et 0,6.
- Accessoires rangés dans un râtelier dont le numéro est égal à son numéro de camion.

### Mises à jour

- Dans la table ACCESSOIRE, substitution d'un trapèze pour lion à la chaise longue de lion.
- Ajouter 10 au volume de la cage à lions.
- Remplacer les ballons par des balles de volume divisé par deux et qui seront rangées dans le râtelier n° 7 du camion n° 8.
- Supprimer Crinière, qui démissionne du cirque.
- Dans la table UTILISATION, détruire la contrainte de clé étrangère sur le TITRE, puis la recréer en lui ajoutant à la fin la clause `ON DELETE CASCADE`.

\* E. Pichat, R. Bodin, "Ingénierie des données", Masson, 1990

- Valider les mises à jour précédentes, puis effacer le numéro « Les Smilers » dans la table NUMERO. Résultat dans NUMERO et UTILISATION ?
- Annuler la suppression précédente.

#### Jointures

- Accessoires servant aux jongleurs.
- Accessoires servant dans plusieurs numéros (solution 1).
- Camions contenant les accessoires servant aux jongleurs.
- Liste des accessoires non-utilisés (solution 1). Insérer un nouvel accessoire quelconque dans la table ACCESSOIRE pour tester la requête.

#### Agrégats et groupements

- Nombre de numéros.
- Nombre de numéros utilisant une balle ou un ballon.
- Volume total et volume moyen des accessoires.
- Accessoires servant dans plusieurs numéros. Proposer deux solutions différentes (solutions 2 et 3).

#### Opérations ensemblistes

- Liste des accessoires non-utilisés (solution 2).
- Liste des jongleurs et des utilisateurs de boule.
- Liste des jongleurs qui utilisent une balle.
- Liste des jongleurs qui n'utilisent pas de balle.

#### Division

- Liste des accessoires utilisés dans tous les numéros. Insérer suffisamment de n-uplets dans la table UTILISATION pour tester la requête et proposer deux solutions différentes.

#### Question

Commentaire sur les clés ?

#### Correction : Base de données

```
create table personnel (nom varchar(20), role varchar(20),
constraint personnel_pri primary key(nom));
```

```
create table numero (titre varchar(30), nature varchar(20), responsable
varchar(20),
constraint numero_pri primary key(titre),
constraint numero_etr foreign key(responsable) references personnel(nom));
```

```
create table accessoire (nom varchar(30), couleur varchar(10), volume
number(4,1), ratelier number(2), camion number(1),
constraint accessoire_pri primary key(nom));
```

```
create table utilisation (titre varchar(30), utilisateur varchar(20), accessoire
varchar(30),
constraint utilisation_pri primary key(titre, utilisateur, accessoire),
constraint utilisation_et1 foreign key(titre) references numero(titre),
constraint utilisation_et2 foreign key(utilisateur) references personnel(nom),
constraint utilisation_et3 foreign key(accessoire) references accessoire(nom));
```

```
insert into personnel values ('&nom','&role');
/
insert into numero values ('&titre','&nature','&responsable');
/
```

```
insert into accessoire select * from darmont.accessoire;
insert into utilisation select * from darmont.utilisation;
```

#### Correction : Requêtes

##### Projections

```
select nature from numero;

select distinct nature from numero;

select camion, volume from accessoire order by camion, volume desc;
```

##### Restrictions

```
select nom from accessoire where volume>=0.3 or couleur='Rouge';

select nom from accessoire where volume between 0.3 and 0.6;

select nom from accessoire where ratelier=camion;
```

##### Mises à jour

```
insert into accessoire values('Trapèze pour lion', 'Bleu', 0.9, 11, 5);
update utilisation set accessoire='Trapèze pour lion'
where accessoire='Chaise longue de lion';
delete from accessoire where nom='Chaise longue de lion';

update accessoire set volume=volume+10 where nom='Cage à lions';

insert into accessoire values('Balle',
(select couleur from accessoire where nom = 'Ballon'),
(select volume/2 from accessoire where nom = 'Ballon'),
7, 8);
```

```

update utilisation set accessoire='Balle' where accessoire='Ballon';
delete from accessoire where nom='Ballon';

delete from utilisation where nom='Crinière';
delete from numero where nom='Crinière';
delete from personnel where nom='Crinière';

alter table utilisation drop constraint utilisation_et1;
alter table utilisation add constraint
    utilisation_et1 foreign key(titre) references numero(titre) on delete cascade;

commit;
delete from numero where titre='Les Smilers';
rollback;

```

### *Jointures*

```

select distinct accessoire from utilisation, personnel
where nom=utilisateur and role='Jongleur';

select distinct u1.accessoire from utilisation u1, utilisation u2
where u1.accessoire=u2.accessoire and u1.titre<>u2.titre;

select distinct camion from accessoire a, utilisation, personnel p
where a.nom=accessoire and utilisateur=p.nom and role='Jongleur';

select nom from accessoire where nom not in
    (select accessoire from utilisation);
ou
select nom from accessoire where not exists
    (select * from utilisation where accessoire=nom);

```

### *Agrégats et groupements*

```

select count(*) from numero;

select count(distinct titre) from utilisation
where accessoire='Balle' or accessoire='Ballon';

select sum(volume), avg(volume) from accessoire;

select accessoire from utilisation
group by accessoire
having count(distinct titre)>1;
ou
select nom from accessoire
where 1<(select count(distinct titre) from utilisation where accessoire=nom);

```

### *Opérations ensemblistes*

```

select nom from accessoire
minus
select distinct accessoire from utilisation;

select nom from personnel where role='Jongleur'
union
select utilisateur from utilisation where accessoire='Boule';

```

```

select nom from personnel where role='Jongleur'
intersect
select utilisateur from utilisation where accessoire='Balle';

select nom from personnel where role='Jongleur'
minus
select utilisateur from utilisation where accessoire='Balle';

```

### *Division*

```

select nom from accessoire where not exists
    (select * from numero n where not exists
        (select * from utilisation u where u.titre=n.titre and accessoire=nom));
ou
select accessoire from utilisation
group by accessoire
having count(distinct titre)=(select count(titre) from numero);

```

### **Correction : Question**

L'utilisation de clés textuelles rend les requêtes plus « lisibles », mais ces clés sont plus difficiles à manipuler que des codes numériques ou alphanumériques (saisie plus fastidieuse et probabilité de faire des erreurs plus grande).