

# Exercices

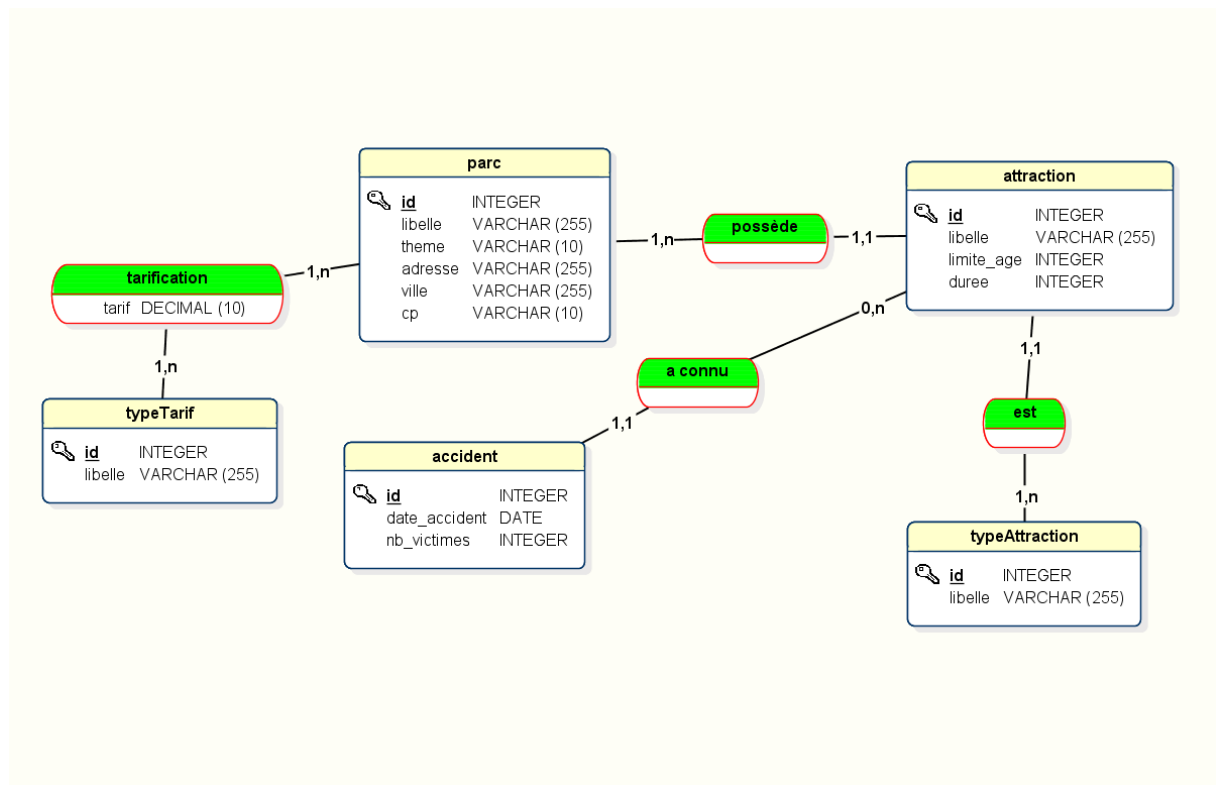
## Entités :

- parc (Le PAL, Walibi, Parc Astérix...)
- attraction (Catapultor, Chenille, Totem, Space Mountain...)
- typeAttraction (Montagnes Russes, Cinéma 4DX, Visite...)
- typeTarif (- de 12 ans, Senior, étudiant...)
- accident
- tarification (C'est l'association d'un parc, d'un type de tarif. Par exemple pour les – de 12 ans au Pal ça sera 10€, mais ça peut être 14€ à Walibi.)

### Règles :

- Une attraction est d'un seul type
- Un parc possède une ou plusieurs attractions
- Une attraction est dans un seul parc
- Une attraction peut ne jamais connaître d'accidents... Ou plusieurs.
- Un accident ne concerne qu'une seule attraction
- Un parc possède différents types de tarifs (-12 ans, étudiant...)
- Une tarification est le résultat de l'association d'un parc et d'un type de tarification. Elle porte la donnée « tarif »

### MCD :



### Exercices :

1 - Trouver les attractions tout public (Celle qui n'ont pas de limite d'âge renseignée)

```
SELECT *
FROM attraction
WHERE limite_age IS NULL;
```

2 – Titouan et sa famille sont pressés car leur bus de touriste part dans 45 minutes. Trouver les attractions qui durent moins de 30 minutes

```
SELECT *  
FROM attraction  
WHERE duree < 30;
```

3 - Trouver les parcs avec le thème "Super-héros"

```
SELECT *  
FROM parc  
WHERE theme LIKE "%Super-héros%";
```

4 - Trouver le tarif pour une famille d'un adulte et un enfant de onze ans pour Le Pal (Pas de tarifs de groupe)

Version simplifiée

```
SELECT SUM(tarif)  
FROM tarification t  
INNER JOIN parc p ON p.id = t.id_parc  
WHERE p.libelle = "Le Pal"  
AND t.id_typeTarif IN(5, 1);
```

Version avec deux jointures

```
SELECT SUM(tarif)  
FROM tarification t  
INNER JOIN parc p ON p.id = t.id_parc  
INNER JOIN typetarif tt ON tt.id = t.id_typeTarif  
WHERE p.libelle = "Le Pal"  
AND (tt.libelle = "Plein tarif" OR tt.libelle = "- de 12 ans");
```

5 - Trouver les attractions réservées aux adultes (plus de 18 ans)

```
SELECT * FROM attraction WHERE limite_age >= 18;
```

6 - Trouver les attractions qui n'ont pas eu d'accident depuis dix ans, y compris celles qui n'en ont jamais eu du tout

```
SELECT a.*, MAX(date_accident) as date_dernier_accident  
FROM attraction a  
LEFT JOIN accident ON accident.id_attraction = a.id  
WHERE (date_accident < DATE_SUB(CURDATE(), INTERVAL 10 YEAR) OR date_accident  
IS NULL)  
GROUP BY a.libelle;
```

7 - Faire le total du nombre de victimes par attraction et les ranger de la plus meurtrière à la plus sécurisée (Ici on parle du nombre de victimes, pas du nombre d'accidents...)

```
SELECT attraction.libelle, SUM(nb_victimes)
FROM accident
INNER JOIN attraction ON attraction.id = accident.id_attraction
GROUP BY attraction.libelle
ORDER BY nb_victimes DESC;
```

8 - Calculer le tarif pour un adulte seul à Walibi

```
SELECT tarif FROM tarification
INNER JOIN parc ON parc.id = tarification.id_parc
INNER JOIN typetarif tt ON tt.id = tarification.id_typeTarif
WHERE parc.libelle = "Walibi"
AND tt.libelle = "Plein tarif"
```

9 - Aïe... Dédé le technicien a laissé traîner ses chaussures sur les rails du Catapultor, un accident est à déplorer. Insérer l'accident dans la table correspondante, à la date du jour. Que les 17 braves âmes qui nous ont quittés aujourd'hui reposent en paix.

```
INSERT INTO accident (date_accident, nb_victimes, id_attraction)
VALUES (CURDATE(), 17, 16);
```

10 – Compter le nombre d'accidents survenus et les grouper par attraction (On parle du nombre d'accidents, pas du nombre de victimes)

```
SELECT COUNT(*), a.libelle
FROM accident
LEFT JOIN attraction a ON a.id = accident.id_attraction
GROUP BY a.libelle;
```

11 – Oh, le petit Timothée est revenu d'entre les morts. Bien qu'il ne sera plus jamais le même, il faut mettre à jour l'enregistrement de l'accident du jour et retirer un mort de notre compteur, ça sera bon pour nos statistiques.

```
UPDATE accident
SET nb_victimes = nb_victimes - 1
WHERE id = 12
```

12 – Faire une moyenne du nombre de victimes par accident

```
SELECT AVG(nb_victimes)
FROM accident
```

13 – Faire une moyenne du nombre de victimes et les grouper par attraction

```
SELECT ROUND(AVG(nb_victimes), 2), a.libelle
FROM accident
INNER JOIN attraction a ON a.id = accident.id_attraction
GROUP BY a.libelle
```