

# Devoir de maison

Kossivi Justin AYIVI  
[justin.ayivi@yahoo.com](mailto:justin.ayivi@yahoo.com)

March 21, 2022

La liste des pistes (avec leur nom, leur longueur, et un booléen indiquant l'ouverture de la piste), la liste des remontées mécaniques (avec leur nom et un booléen indiquant l'ouverture de la remontée) et des mesures de hauteur de neige sur les pistes (avec l'emplacement de la mesure et la hauteur mesurée) sont stockée dans une base relationnelle selon le modèle conceptuel des données suivant :

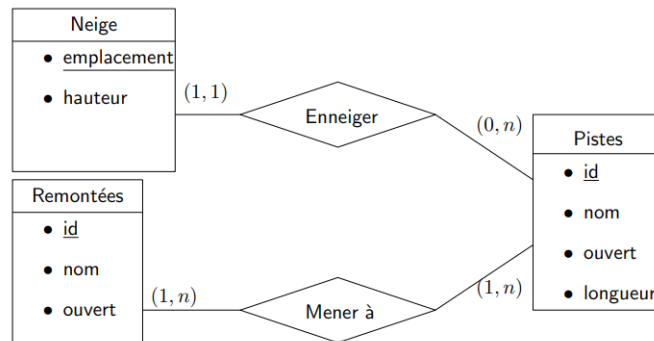


Figure 1: Schéma conceptuel initial

Remarques : — Les attributs id sont des entiers choisis arbitrairement. — L'attribut emplacement est un texte arbitraire indiquant la position de la mesure. — L'attribut hauteur est une hauteur en centimètres. — L'attribut longueur est une longueur en mètres

## Question 1

**Expliquer les cardinalités indiquées sur le schéma, en précisant le nombre de mesures de hauteur de neige possibles par pistes, et le nombre de remontées mécaniques menant à une piste, ainsi que le nombre de pistes accessibles par une remontée**

Pistes  $\overset{0,n}{\text{---}}$  Neige : Sur une piste on peut avoir plusieurs mesures de hauteur de neige ou pas du tout.

Neige  $\overset{1,1}{\text{---}}$  Piste : Une mesure de hauteur de neige correspond à 1 et 1 seule piste.

Piste  $\overset{1,n}{\text{---}}$  Remontées : Sur une piste on a au moins une remontée mécanique.

Remontées  $\overset{1,n}{\text{---}}$  Pistes : Une remontée mécanique mène au moins à 1 piste.

## Question 2

**Donner le modèle conceptuel des données associés au modèle précédent.**

Nous avons la cardinalité (1,1) au niveau de Neige  $\rightarrow$  Enneiger. Cela indique que cette dernière entité sera celle qui recevra la clé étrangère de Pistes. On a ainsi le schéma logique avec les tables suivantes :

```
Neige(emplacement, hauteur, #Pistes.id)
Pistes(id, hauteur, nom, ouvert, longueur)
Remontées(id, nom, ouvert) puis l'association
```

Mener à (#Pistes.id, #Remontées.id)

Question 3

**Écrire une requête pour donner la liste des pistes.**

```
SELECT * FROM Pistes;
```

Question 4

**Écrire une requête pour donner la liste des pistes ouvertes.**

```
SELECT*FROM Pistes WHERE Pistes.ouvert = 1 ;
```

Question 5

**Écrire une requête pour donner la plus longue des pistes ouvertes.**

<pre>SELECT*FROM Pistes WHERE Pistes.ouvert = 1 ORDER BY longueur DESC LIMIT 1;</pre>	<pre>SELECT*FROM Pistes WHERE Pistes.ouvert = 1 AND longueur = (SELECT Max(longueur) FROM Pistes);</pre>
---	--

Question 6

**Écrire une requête pour donner la liste des remontées mécaniques ouvertes.**

```
SELECT * FROM Remontées
WHERE ouvert = 1;
```

Question 7

**Écrire une requête pour donner la liste des hauteurs de neige pour la piste portant le nom Hiver.**

```
SELECT * FROM Neige.emplacement, Neige.hauteur, Pistes.nom
FROM Pistes, Neige
WHERE Pistes.nom = 'Hiver' AND Pistes.id = Neige.id;
```

Question 8

**Écrire une requête pour donner la liste des remontées mécaniques menant à la piste portant le nom Hiver.**

```
SELECT Remontées.id, Remontées.nom, Remontées.longueur, Pistes.nom
FROM Pistes, Remontées
WHERE Pistes.nom = 'Hiver' AND Pistes.id = Remontées.id;
```

Question 9

**Écrire une requête pour donner la liste des pistes accessibles par la remontée mécanique portant le nom Grande.**

```
SELECT Pistes.id, Pistes.nom, Pistes.ouvert, Remontées.nom
FROM Pistes, Remontées
WHERE Pistes.ouvert = 1 AND Remontées.nom = 'Grande' AND Pistes.id = Remontées.id;
```

Question 10

**Écrire une requête pour donner la hauteur de neige moyenne de la piste portant le nom Hiver.**

```
SELECT AVG(Neige.hauteur)
FROM Pistes, Neige
WHERE Pistes.nom = 'Hiver' AND Pistes.id = Neige.id ;
```

#### Question 11

Écrire une requête pour donner la hauteur de neige moyenne globale sur toutes les pistes.

```
SELECT AVG(Neige.hauteur)
FROM Neige;
```

#### Question 12

Écrire une requête insérant une mesure de hauteur de neige, à l'emplacement Sommet pour la piste Hiver, avec une valeur de 230cm.

On fait une insertion dans deux tables vu que la clé étrangère de Pistes se trouve dans Neige.

INSERT	INSERT
INTO Neige	INTO Pistes (id, nom)
VALUES('Sommet', 230, 99999999);	VALUES(99999999, 'Hiver');

#### Question 13

On veut désormais stocker le graphe des pistes (il s'agit d'un graphe orienté sans cycles). On doit pour cela connaître la liste des pistes accessibles à partir de chaque piste. Décrire les modifications nécessaires dans la base.

Le graphe, étant choisi comme alternative à la normalisation relationnelle, nous permet de gérer les problèmes de dépendances entre les éléments d'une table en particulier les pistes dans ce cas. Nous devons transformer la structure de stockage et opter pour une solution de stockage dérivé d'une famille NoSQL précisément orienté graphe. La modification se fera à trois niveaux. Nous créons d'abord :

- Les noeuds: Ils représenteront ici les différentes pistes et leurs informations (nom, longueur, etc)
- Les liens : Ils désigneront quant à eux les relations entre les pistes; notamment l'accès à une piste à partir d'une piste donnée;
- Enfin, les propriétés sur ces noeuds et ces liens : par exemple l'ordre dans lequel les pistes sont atteintes.

#### Question 14

Écrire une requête pour donner la liste des pistes accessibles à partir de la piste Hiver.

Il serait plus judicieux de répondre à cette question dans un contexte de non relationnel. Mais dans un raisonnement relationnel, nous suggérons de créer un attribut *Accessible* dans la table Pistes qui représentera le nom de la piste accessible à partir d'un nom de piste i.e.

```
Pistes(id, hauteur, nom, ouvert, longueur, Accessible)
```

```
SELECT Accessible FROM Pistes
WHERE nom = 'Hiver'
```

#### Question 15

Proposer, sans code, une solution pour trouver l'enchaînement de pistes le plus long (sans emprunter de remontée).

1. On commence par le noeud qui a la piste la plus longue,
2. Ensuite, on cherche parmi l'ensemble des pistes reliées à ce noeud celle qui a la piste la plus longue,

3. On réitère ce processus jusqu'à ce qu'aucune piste ne soit accessible à partir de la piste précédente,
4. On s'arrêtera ainsi sur cette piste.

## Skieurs

Note : cette partie est volontairement laissée beaucoup plus libre. On veut désormais analyser le comportement des skieurs. Quelques objectifs sont les suivants :

- connaître les pistes les plus fréquentées, les moins fréquentées ;
- connaître la longueur parcourue et la durée passée sur les pistes, pour chaque skieur ;
- connaître les remontées mécaniques les plus fréquentées, les moins fréquentées ;
- envoyer des offres promotionnelles aux clients qui passent le plus de temps sur les pistes.

Cette liste n'est pas exhaustive, d'autres types d'analyses intéressantes seront les bienvenus. On supposera que l'on dispose d'un moyen technique pour connaître le passage d'un skieur sur la piste (le badge sans-contact de son forfait par exemple).

### Question 16

**Proposer et discuter des solutions pour réaliser cette analyse. On détaillera notamment les schémas logiques et conceptuels nécessaires.**

Pour réaliser cette analyse, on créera trois tables : une première table à savoir la table skieur pour les informations sur le skieur et notamment son passage sur une piste donnée à travers l'association *a badgé*. Une deuxième table qui est celle de la localisation qui donne l'information sur la VILLE de provenance du skieur ou de localisation de la piste puis une troisième table qui donne l'entreprise dont est client le skieur pour profiter éventuellement des offres promotionnelles. Les tables sont les suivantes:

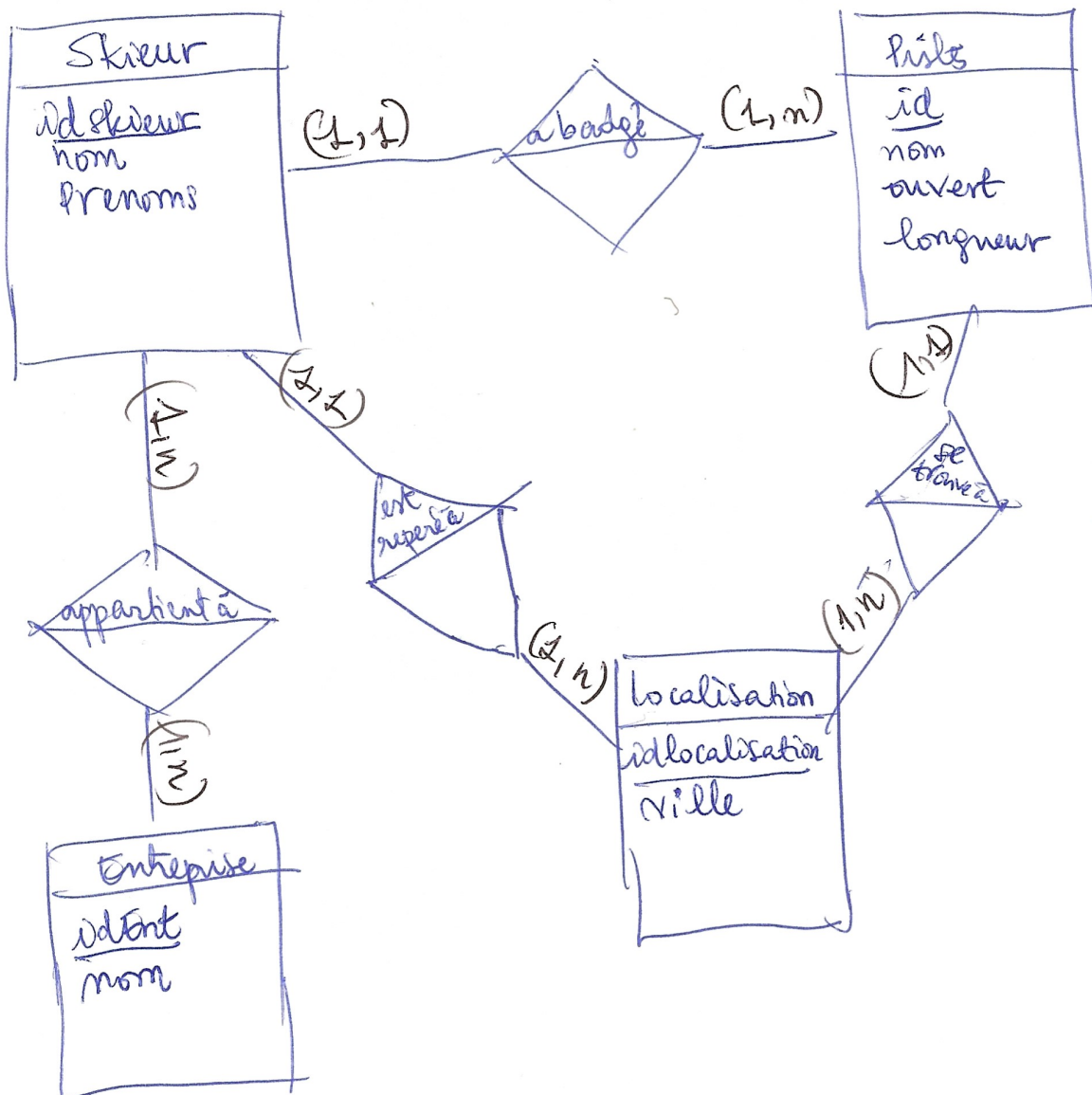
```
Skieur(idskieur, nom, prenom)
Entreprise(ident, nom)
Localisation(idlocalisation, ville) et des relations
a badgé(#idskieur, #Pistes.id) entre la table skieur et la table Pistes
Est repéré à (#Pistes.id, #idskieur) entre la table skieur et la table Localisation
Se trouve à (#idskieur, #Pistes.id) entre la table Piste et la table Localisation
Appartient à (#idskieur, #ident) entre la table Skieur et la table Banque
```

Nous expliquons les cardinalités de la façon suivante :

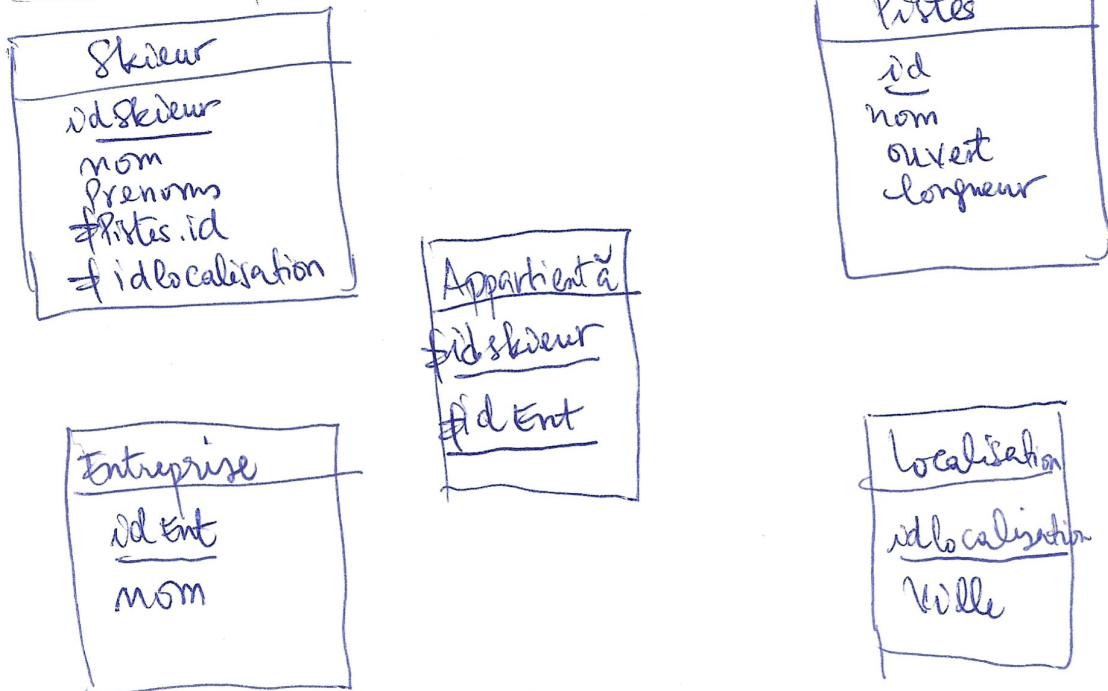
Skieur  $\overset{1,1}{-}-$  Piste : Un skieur a normalement badgé sur une seule piste.  
Piste  $\overset{1,n}{-}-$  Skieur : Une piste peut voir le passage d'un ou plusieurs skieurs.  
Skieur  $\overset{1,1}{-}-$  Localisation : Un skieur se trouve à une et une seule localisation.  
Localisation  $\overset{1,n}{-}-$  Skieur : Une localisation peut permettre de repérer un ou plusieurs skieurs.  
Pistes  $\overset{1,1}{-}-$  Localisation : Une piste se trouve à une et une seule localisation.  
Localisation  $\overset{1,n}{-}-$  Piste : A une localisation donnée se trouve une ou plusieurs pistes.  
Skieur  $\overset{1,n}{-}-$  Entreprise : A une entreprise appartient au moins un skieur  
Entreprise  $\overset{1,n}{-}-$  Skieur : Un skieur peut appartenir à une ou plusieurs entreprises.

Le schéma conceptuel associé au modèle ci-dessus est présenté à la page suivante auquel l'on joint le modèle de schéma conceptuel de la figure (1) :

## Schema conceptuel



## Schema logique



Question 17

**Illustrer les nouvelles possibilités permises par votre schéma à travers quelques exemples.**

On peut répondre à la question de la provenance des skieurs et leur densité au niveau des différentes pistes. On peut également trouver la localisation des pistes où l'on a le plus de skieurs. Enfin, on peut trouver l'entité où les clients passent beaucoup plus de temps sur les pistes. Notamment dénicher le service qui offrira beaucoup d'offres promotionnelles à ces clients.