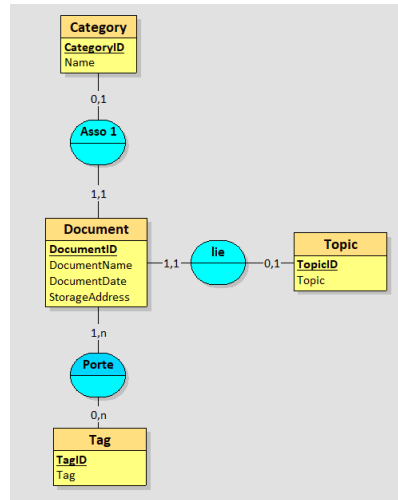


Atelier IT

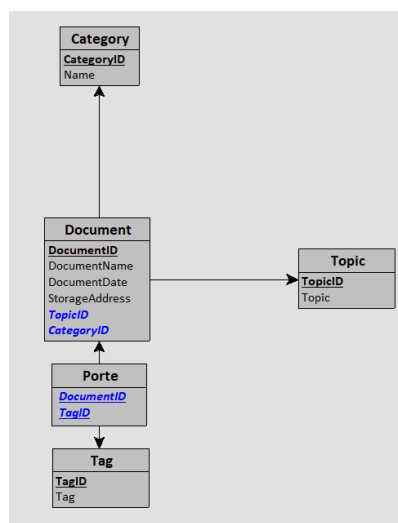
TP4

1- Créer la base de données

Afin de créer notre base de données sur MySQL Workbench, nous définissons un schéma relationnel sur Looping, à partir duquel nous générons un code SQL. Notre schéma relationnel est le suivant :



Pour les cardinalités, il est précisé qu'un document peut avoir n'importe quel nombre de balises (tag), alors la cardinalité est (1,n) pour un document, et (0,n), pour n'importe quel nombre de balises. En format MLD, cela nous permet d'avoir une cinquième table liant les entités « Topic », et « Document ». Les raisonnements pour les associations document/category et document/topic sont les mêmes. Looping ne permet de créer des associations aux cardinalités (1,1), (1,1). Or, un document n'a qu'un seul thème et qu'une seule catégorie. Alors nous avons pris la décision d'utiliser des cardinalités (1,1), (0,1). Cela signifie que dans le SQL généré par Looping, un document pourra ne pas avoir de catégorie et de thème. Nous veillerons donc à corriger cela lors de la création du programme en java.



Comme nous pouvons le voir dans le MLD correspondant, ces cardinalités nous permettent de créer un document avec les clés étrangères adaptées (une balise, un thème, et une catégorie pour chaque document). Les clés primaires restent les mêmes que dans le schéma relationnel. Cependant,

dans les clés étrangères, nous observons que les clés CategoryID et TopicID sont décrites comme uniques, or nous ne voulons pas que ce soit le cas. Alors dans le code SQL, nous retirons les paramètres « Unique (TopicID) » et « Unique (CategoryID) ».

Notre code SQL de création de la base de données et d'insertions dans les tables Category et Topic est disponible sur le fichier TP4.sql.

8	10:30:56	CREATE TABLE Category(CategoryID INT, Name VARCHAR(50), PRIMARY KEY(CategoryID))engine = InnoDB	0 row(s) affected	0.016 sec
9	10:30:56	CREATE TABLE Topic(TopicID INT, Topic VARCHAR(50), PRIMARY KEY(TopicID))engine = InnoDB	0 row(s) affected	0.000 sec
10	10:30:56	CREATE TABLE Tag(TagID INT, Tag VARCHAR(50), PRIMARY KEY(TagID))engine = InnoDB	0 row(s) affected	0.016 sec
11	10:30:56	CREATE TABLE Document(DocumentID INT, DocumentName VARCHAR(50), DocumentDate DATE, StorageAddress INT, TagID INT NO...	0 row(s) affected	0.031 sec
12	10:30:56	INSERT INTO Category (CategoryID, Name) VALUES (1, 'policy'), (2, 'plan'), (3, 'report'), (4, 'receipt'), (5, 'order')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
13	10:30:56	INSERT INTO Topic (TopicID, Topic) VALUES (1, 'CS240 Course Files in Fall 2019'), (2, 'CS241 Course Files in Spring 2020'), (3, 'CS242 Cours...	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.000 sec
14	10:30:56	COMMIT	0 row(s) affected	0.000 sec

▼ tp4

▼ Tables

- category
- document
- porte
- tag
- topic

1 • SELECT

2 *

3 FROM tp4.category

	CategoryID	Name
▶	1	policy
	2	plan
	3	report
	4	receipt
	5	order
*	NULL	NULL

1 • SELECT

2 *

3 FROM tp4.topic

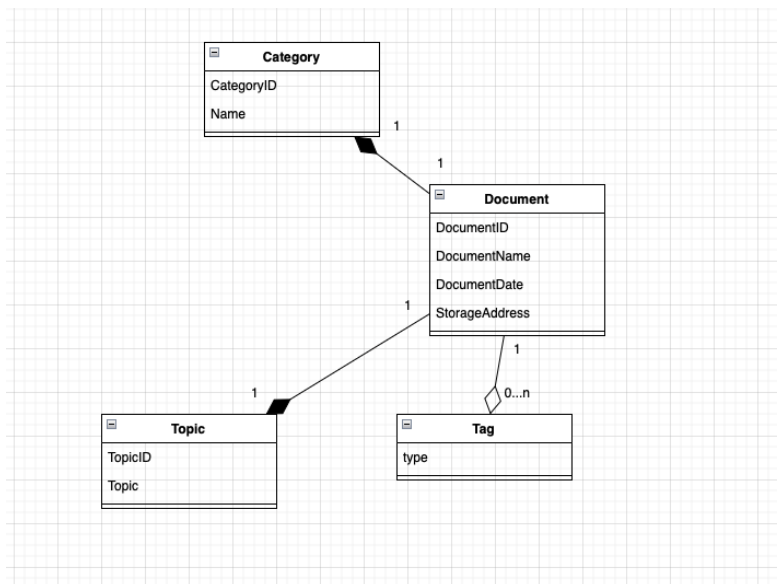
	TopicID	Topic
▶	1	CS240 Course Files in Fall 2019
	2	CS241 Course Files in Spring 2020
	3	CS242 Course Files SummerCamp 2020
	4	CS240 Course Files in Fall 2020
	5	CS241 Course Files in Spring 2021
	6	CS242 Course Files SummerCamp 2021
	7	CS240 Course Files in Fall 2021
	8	CS241 Course Files in Spring 2022
	9	CS242 Course Files SummerCamp 2022
*	NULL	NULL

Nous pouvons voir que notre scripte tp4.sql fonctionne correctement.

2- Créer le diagramme de classes

Concernant le diagramme de classe, il y a deux relations de composition pour Document envers Category, mais également pour Document envers Topic. En effet, selon la consigne, il est dit qu'un

document "appartient" à une catégorie, et de même, "chaque document a un thème". La relation de composition qui représente une appartenance forte est ainsi le meilleur choix, puisqu'un document nécessite à la fois une catégorie et un thème. Pour la relation entre document et balise, nous avons choisi l'agrégation, qui représente une appartenance faible, et dans la consigne, un document "peut" avoir des balises. Ainsi, il n'est pas forcé d'en avoir, au contraire de catégorie et de thème.



3- Le programme

Pour la partie « Le programme », tout notre code java est disponible à l'adresse locale suivante :

..\TP4\src\tp4

a.

Voici pour cet exercice des captures d'écran de notre terminal Eclipse et de notre SQL Workbench, pour prouver que nous avons créé les 5 instances de documents, avec toutes les propriétés nécessaires, et que nous avons pu sauvegarder ces ajouts sur notre base de données.

```
1 doc alpha 2022-06-21 1 7 4 1
3 doc final 2022-06-21 3 1 1 1
4 doc pre release 2022-06-21 4 4 5 2
5 doc test 2022-06-21 5 8 3 3
2 doc beta 2022-06-21 2 8 4 4
Query executed.
```

1 • `SELECT * from tp4.document Join tp4.porte on tp4.document.DocumentID=tp4.porte.DocumentID;`

	DocumentID	DocumentName	DocumentDate	StorageAddress	TopicID	CategoryID	DocumentID	TagID
▶	1	doc alpha	2022-06-21	1	7	4	1	1
	3	doc final	2022-06-21	3	1	1	3	1
	4	doc pre release	2022-06-21	4	4	5	4	2
	5	doc test	2022-06-21	5	8	3	5	3
	2	doc beta	2022-06-21	2	8	4	2	4

Pour remplir la base de données, l'utilisateur à simplement besoin de dire qu'il souhaite ajouter un document, donner son nom (« DocumentName »), et donner les numéros correspondants pour le thème, la balise, et la catégorie (« TagID », « TopicID », « CategoryID »). Pour le moment, nous partons du principe qu'un fichier a uniquement et obligatoirement une balise, un thème, et une catégorie.

b. i.

Par thème :

```
1 CS240 Course Files in Fall 2019
2 CS241 Course Files in Spring 2020
3 CS242 Course Files SummerCamp 2020
4 CS240 Course Files in Fall 2020
5 CS241 Course Files in Spring 2021
6 CS242 Course Files SummerCamp 2021
7 CS240 Course Files in Fall 2021
8 CS241 Course Files in Spring 2022
9 CS242 Course Files SummerCamp 2022

Choisissez un thème :
8
Query executed.
Nom : doc beta / Thème : CS241 Course Files in Spring 2022
Nom : doc test / Thème : CS241 Course Files in Spring 2022
```

```
1 • SELECT
2     tp4.document.DocumentName,
3     tp4.topic.Topic
4 from tp4.document
5 Join tp4.topic on tp4.document.TopicID=tp4.topic.TopicID
6 WHERE tp4.document.TopicID=8;
```

	DocumentName	Topic
▶	doc beta	CS241 Course Files in Spring 2022
	doc test	CS241 Course Files in Spring 2022

Par balise :

Choisissez une balise :
 1
 Query executed.
 Nom : doc final / Balise : legal

```
1 • SELECT
2     tp4.document.DocumentName,
3     tp4.tag.Tag
4 from tp4.document
5 Join tp4.porte on tp4.porte.DocumentID=tp4.document.DocumentID
6 Join tp4.tag on tp4.porte.TagID=tp4.tag.TagID
7 WHERE tp4.document.CategoryID=1;
```

	DocumentName	Tag
►	doc final	legal

Par catégorie :

Choisissez une catégorie :
 1
 Query executed.
 Nom : doc final / Catégorie : policy

```
1 • SELECT
2     tp4.document.DocumentName,
3     tp4.category.Name
4 from tp4.document
5 Join tp4.category on tp4.document.CategoryID=tp4.category.CategoryID
6 WHERE tp4.document.CategoryID=1;
```

	DocumentName	Name
►	doc final	policy

ii.

4 - Afficher le sujet le plus fréquent
 5 - Afficher le nb d'occurrences par tag
 6 - Enregistrer un nouveau document sans date
 7 - Enregistrer nouvelle date sur un document
 8 - Enregistrer nouveau tag sur un document
 4
 Query executed.
 Thème le plus fréquent : CS241 Course Files in Spring 2022

```
1 • SELECT
2     tp4.topic.Topic,
3     Count(Topic) as nb_of_occurence
4 from tp4.document
5 Join tp4.topic on tp4.topic.TopicID=tp4.document.TopicID
6 Group by tp4.topic.Topic
7 Order by nb_of_occurence desc
8 Limit 1;
```

	Topic	nb_of_occurence
►	CS241 Course Files in Spring 2022	2

iii.

```

5 - Afficher le nb d'occurrences par tag
6 - Enregistrer un nouveau document sans date
7 - Enregistrer nouvelle date sur un document
8 - Enregistrer nouveau tag sur un document
5
Query executed.
Balise : legal / Nombre d'apparitions : 2
Balise : medical / Nombre d'apparitions : 1
Balise : administrative / Nombre d'apparitions : 1
Balise : technical / Nombre d'apparitions : 1

```

```

1 • SELECT
2     tp4.tag.Tag,
3     Count(Tag) as nb_of_occurrence
4 from tp4.document
5 Join tp4.porte on tp4.porte.DocumentID=tp4.document.DocumentID
6 Join tp4.tag on tp4.tag.TagID=tp4.porte.TagID
7 Group by tp4.tag.Tag;

```

	Tag	nb_of_occurrence
▶	legal	2
	medical	1
	administrative	1
	technical	1

c. i.

```

1 doc alpha 2022-06-21 1 7 4 1
3 doc final 2022-06-21 3 1 1 1
4 doc pre release 2022-06-21 4 4 5 2
5 doc test 2022-06-21 5 8 3 3
2 doc beta 2022-06-21 2 8 4 4
6 doc sans date null 6 4 3 6
Query executed.

```

```

1 • SELECT
2     *
3 FROM tp4.document

```

	DocumentID	DocumentName	DocumentDate	StorageAddress	TopicID	CategoryID
▶	1	doc alpha	2022-06-21	1	7	4
	2	doc beta	2022-06-21	2	8	4
	3	doc final	2022-06-21	3	1	1
	4	doc pre release	2022-06-21	4	4	5
	5	doc test	2022-06-21	5	8	3
	6	doc sans date	NULL	6	4	3
*	NULL	NULL	NULL	NULL	NULL	NULL

Pour les deux exercices restants, nous ne nous contenterons pas de modifier le dernier document sans date (il aurait suffi d'implémenter les fonctions ensemble pour forcer l'utilisateur à rentrer une date et ajouter des balises). Nous allons donc permettre à l'utilisateur de modifier les dates ou d'ajouter des balises à tous les documents.

ii.

```

2 doc beta 2022-06-21 2 8 4 4
6 doc sans date null 6 4 3 6
Query executed.
Quel document souhaitez-vous modifier ? (donnez le numéro) :
6
Entrez une nouvelle date (format jj/mm/aaaa) :
12/05/2021
Query executed.
1 doc alpha 2022-06-21 1 7 4 1
2 doc beta 2022-06-21 2 8 4 4
3 doc final 2022-06-21 3 1 1 1
4 doc pre release 2022-06-21 4 4 5 2
5 doc test 2022-06-21 5 8 3 3
6 doc sans date 2021-05-12 6 4 3 6

```

```

1 • SELECT
2      *
3 FROM tp4.document

```

	DocumentID	DocumentName	DocumentDate	StorageAddress	TopicID	CategoryID
▶	1	doc alpha	2022-06-21	1	7	4
	2	doc beta	2022-06-21	2	8	4
	3	doc final	2022-06-21	3	1	1
	4	doc pre release	2022-06-21	4	4	5
	5	doc test	2022-06-21	5	8	3
	6	doc sans date	2021-05-12	6	4	3
*	NULL	NULL	NULL	NULL	NULL	NULL

En trouvant une solution pour modifier la date d'un dossier, nous pouvons modifier notre code afin de laisser l'utilisateur choisir la date des fichiers. Cependant, lors de la création du fichier, nous ne laissons pas le choix de la date, car par soucis de réalisme, nous partons du principe que la date de création est celle du jour où le fichier a été implémenté dans la base de données. Ainsi nous prouvons que nous savons récupérer une date donnée par l'utilisateur, mais nous ne l'appliquons pas pour répondre aux questions précédentes.

iii.

Pour répondre à cette dernière question, nous avons simplement à insérer de nouvelles valeurs dans la table « porte », ici par exemple, avec les paramètres DocumentID = 6 et TagID = 1.

```

Quelle balise voulez-vous ajouter ? (donnez le numéro) :
1
Query executed.
1 doc alpha 2022-06-21 1 7 4 1
2 doc beta 2022-06-21 2 8 4 4
3 doc final 2022-06-21 3 1 1 1
4 doc pre release 2022-06-21 4 4 5 2
5 doc test 2022-06-21 5 8 3 3
6 doc sans date 2021-05-12 6 4 3 1
6 doc sans date 2021-05-12 6 4 3 6
Query executed.

```

```

1 • SELECT * from tp4.document Join tp4.porte on tp4.document.DocumentID=tp4.porte.DocumentID;

```

	DocumentID	DocumentName	DocumentDate	StorageAddress	TopicID	CategoryID	DocumentID	TagID
▶	1	doc alpha	2022-06-21	1	7	4	1	1
	2	doc beta	2022-06-21	2	8	4	2	4
	3	doc final	2022-06-21	3	1	1	3	1
	4	doc pre release	2022-06-21	4	4	5	4	2
	5	doc test	2022-06-21	5	8	3	5	3
	6	doc sans date	2021-05-12	6	4	3	6	1
	6	doc sans date	2021-05-12	6	4	3	6	6