

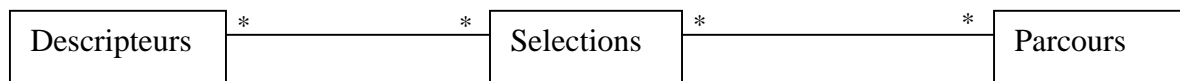
## Modelisation et conception

Nous avons choisi de créer des tables dans la base de données afin de résoudre le problème du stockage des descripteurs.

Nous voulons les informations suivantes :

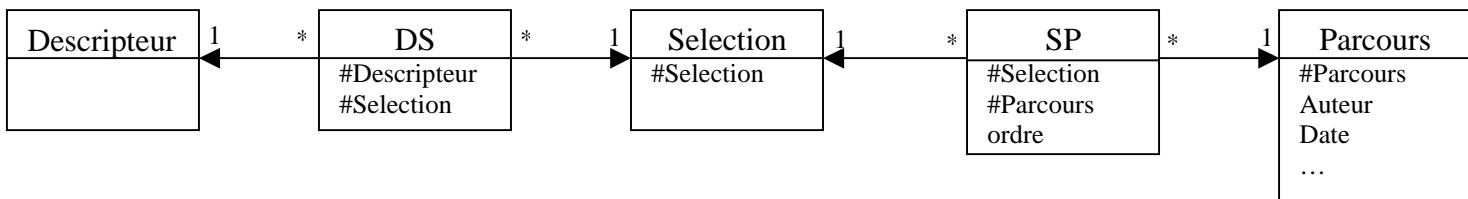
- L'état d'une sélection
- L'auteur du graphe de parcours
- Le nom du parcours

Nous avons à notre disposition dans la base de données une table de descripteurs. Nous avons donc choisi de mettre en place le système suivant.



Afin de réaliser le schéma ci-dessous, nous allons devoir créer quatre tables. Nous utiliserons une table Descripteurs-Selection ou DS, une table Selection, une table Selection-Parcours, et une table Parcours.

Nous obtiendrons alors le schéma suivant :



Nous allons maintenant passer à l'explication en détails de ces tables...

La table Descripteurs sera celle utilisée dans Porphyry.

La table DS aura comme attributs l'identifiant du Descripteur ainsi que l'identifiant de la sélection à laquelle il est associé.

La table Selection sera composée d'un attribut identifiant la sélection.

Table DS :

<i>Attributs</i>	<i>Contraintes</i>
#Descripteur	Number Cle principale, Cle etrangere
#Selection	Number Cle principale, Cle etrangere

Table Selection :

<i>Attributs</i>	<i>Contraintes</i>
#Selection	Number Cle principale

Table SP :

<i>Attributs</i>	<i>Contraintes</i>
#Selection	Number Cle principale, Cle etrangere
#Parcours	Number Cle principale, Cle etrangere
Ordre	Number not null

Table Parcours :

<i>Attributs</i>	<i>Contraintes</i>
#Parcours	Number Cle principale
Auteur	Varchar(2) not null
Date	Date not null
Nom_parcours	Varchar2(20) not null

Creation des tables :

```

Create table DS( Id_descripteur Number(*,*) CONSTRAINT nn_desc NOT NULL,
                Id_selection Number(*,*) CONSTRAINT nn_selection NOT NULL,
                CONSTRAINT fk_selection FOREIGN KEY (Id_selection)
REFERENCES selection(Id_selection),
                CONSTRAINT fk_desc FOREIGN KEY (Id_descripteur)
REFERENCES descripteur(Id_descripteur),
                CONSTRAINT pk_ds PRIMARY KEY(Id_descripteur, Id_selection))

```

```

Create table Selection(Id_selection Number(*,*) CONSTRAINT pk_selection PRIMARY
KEY)

```

```

Create table SP(Id_selection Number(*,*) CONSTRAINT nn_selection NOT NULL,
                Id_parcours Number(*,*) CONSTRAINT nn_parcours NOT NULL,
                Ordre Number(*,*) CONSTRAINT nn_ordre NOT NULL,
                CONSTRAINT fk_selection FOREIGN KEY (Id_selection)
REFERENCES selection(Id_selection),
                CONSTRAINT fk_parcours FOREIGN KEY (Id_parcours)
REFERENCES parcours(Id_parcours),
                CONSTRAINT pk_sp PRIMARY KEY(Id_selection, Id_parcours))

```

Create table parcours (Id\_parcours Number(\*,\*) **CONSTRAINT** pk\_parcours **PRIMARY KEY**,

Auteur varchar2(25) **CONSTRAINT** nn\_auteur **NOT NULL**,  
Date date **CONSTRAINT** nn\_date **NOT NULL**)

Ces requetes sont a encastrer dans le langage java.

Il faudra se connecter sur la base existante, créée avec sql server.

Ce ne sont la que des requetes de test pour pouvoir concretement observer les modifs sur la base de donnees.

Nous etablirons des requetes plus detaillees et plus correctes apres les tests pour la realisation

De plus, il faudra mettre en place des triggers de mise a jour afin de remplir ces nouvelles tables a l'aide des donnees presentes dans les tables deja dans la base de donnees.

Il faut créer des triggers en java qui puissent mettre a jour nos nouvelles tables avec les données fournies par l'utilisateur.

## Interface graphique :

Pour arriver a l'écran présente ci dessous ainsi qu'aux suivants, il faudra mettre en place un bouton dans l'interface graphique de porphyry ; on pourrait l'intituler recherche. Il permettrait de renseigner sous ce nom les graphes de parcours enregistrés par des utilisateurs pour aider a la recherche.

La recherche sera effectuée sur une sélection entrée par l'utilisateur, on effectuera ensuite une recherche parmi les sélections situées dans notre table. On renverra ensuite les graphes de parcours correspondant avec le nom de l'auteur, laissant a l'utilisateur la possibilité de choisir celui qu'il veut...l'affichage de ces parcours de recherche est encore a définir...

On appelle selection une liste de descripteurs.

Ecran 1 :

**RECHERCHE DANS UN GRAPHE DE PARCOURS**

Entrer votre selection :

« selection de l'utilisateur »	Recherche
--------------------------------	-----------

Utilisation d'un champ, zone de texte ou l'utilisateur entre la selection qu'il souhaite trouver parmi les graphes de recherche. On recupere cette selection ds une variable que lon integre ensuite a une requete sur a base de donnees.

Le bouton recherche lance la recherche dans la base de donnees pour trouver les graphes de recherche possédant dans leur parcours notre selection. On fait ainsi le choix de faire une recherche sur l'ensemble des selections des graphes et non pas sur la premiere selection des graphes. De plus, on ne fait pas la recherche sur le corpus auquel on aboutit avec la selection mais sur la selection elle meme.

Ecran 2 :

## RESULTAT DE LA RECHERCHE

Les graphes suivants contiennent votre selection :

<u>Nom parcours 1</u>	<input checked="" type="checkbox"/>
<u>Nom graphe 2</u>	<input type="checkbox"/>
<u>Nom graphe 3</u>	<input type="checkbox"/>
<u>Nom graphe 4</u>	<input type="checkbox"/>
<u>Nom graphe 5</u>	<input type="checkbox"/>
<u>Nom graphe 6</u>	<input type="checkbox"/>
.	<input type="checkbox"/>
.	<input type="checkbox"/>
.	<input type="checkbox"/>
<u>Nom graphe n</u>	<input type="checkbox"/>

**RETOUR**

Utilisation d'une liste de choix, resultat d'une requete sur la base de donnees. Si l'on clique sur un choix alors on aboutit sur l'ecran suivant.

Ecran 3 :

<p style="text-align: center;"><b>RESULTAT DE LA RECHERCHE</b> <b>Nom_graphe_1</b> <b>Auteur</b> <b>Date de creation</b></p> <p style="text-align: center;"><b>RETOUR</b></p>
---

Pour ce qui est de l'affichage des graphes de parcours il faut trouver une methode qui puissent nous permettre de montrer des eventuels retours en arriere ds la recherche.

La première idée est une liste chaînée mais il ne sera pas facile de montrer a l'utilisateur un éventuel retour en arrière dans la recherche.  
Je pense que nous pourrions utiliser une liste chaînée simple pour afficher les graphes de parcours des utilisateurs...nous pourrions résoudre le problème des retours en arrière en utilisant un cadre ou on pourrait afficher les retours en arrière éventuel dans l'arbre de parcours.

Cette solution me paraît la plus simple a envisager, il suffirait de placer les selections de l'utilisateur dans les cellules et de relier celles ci entre elles.  
On obtiendrait ainsi le parcours de recherche de l'utilisateur. On pourrait realiser une sorte de compteur qui indiquerait si le parcours de recherche contient des retours en arriere, et si oui ou se trouvent ces retours en arriere, ou bien mettre en place un systeme de signalement des retours en arriere sur les cellules concernees(couleur...).

Ca me semble difficile de faire en sorte que ces retours en arriere soient autrement visible...

### Detecter les retours en arriere :

Il faut pour cela parcourir la liste des selections d'un parcours et les retours en arriere apparaitront s'il on a deux fois la meme selection, on peut ensuite grace a l'ordre les positionner sur le parcours.

Une seconde solution pour l'affichage de ces graphes serait de créer une structure propre basee sur les listes chainees avec une cellule qui contiendrait la selection courante. Il faudrait ensuite pouvoir créer une structure capable de gerer le probleme des retours en arriere dans le graphe de recherche.

Reunion avec aurelien le 31/07/01 :

Idee de l'interface graphique : la recherche seffectue sur la selection courante donc le probleme de savoir recuperer la selection ne se pose plus.

Il faut memoriser au fur et a mesure les selections de l'utilisateur dans un parcours appele « parcours en cours... » qui sera conserve sur le cache de la machine. Il aura la possibilite de choisir par un bouton de créer ou consulter les parcours de recherche.

Il sera alors envoye vers un ecran du type :

## ***PARCOURS DE RECHERCHE*** ***Consultation/modification***

Parcours en cours...	▼
Parcours_1_contenant_selection_courante	
Parcours_2_contenant_selection_courante	
Parcours_3_contenant_selection_courante	
Parcours_4_contenant_selection_courante	
Parcours_5_contenant_selection_courante	
Parcours_6_contenant_selection_courante	

<i>Parcours_1_contenant_sele ction_courante</i>	▲
Selection1 Selection2 . .	▼
selection_courante	<b>Enregistrer le parcours</b>
Selectionn	<b>Exporter en document XML</b>

Copier
Coller
Supprimer

### Explications sur la nouvelle interface :

Le premier menu rassemble les parcours de recherche contenant la selection courante de l'utilisateur (a fortiori son parcours en cours).

Lorsque l'utilisateur selectionne un parcours on affiche dessous la liste des selections de ce parcours avec la selection courante choisie. L'utilisateur peut ensuite aller et venir dans le parcours grace aux fleches designant la selection suivante et precedente.

On laisse la possibilite a l'utilisateur de modifier le parcours, ainsi il peut bouger la selection choisie la ou il le souhaite et de la meme facon il peut supprimer des selections qu'il estime inutiles.

Il peut ensuite enregistrer son parcours parmi les parcours de recherche deja existants. Il peut egalement exporter ce parcours en tant que document xml mais ce la ne sera peut etre pas gere ici.

Une autre idee :

Créer un menu horizontale deroulant dans linterface de porphyry....

