

TD SQL 2

sur la modélisation des Théâtres

pour rappel voici

- le texte de présentation de la modélisation
- les relations retenues pour la modélisation
- les commandes SQL ayant permis de créer les tables à partir de ces relations

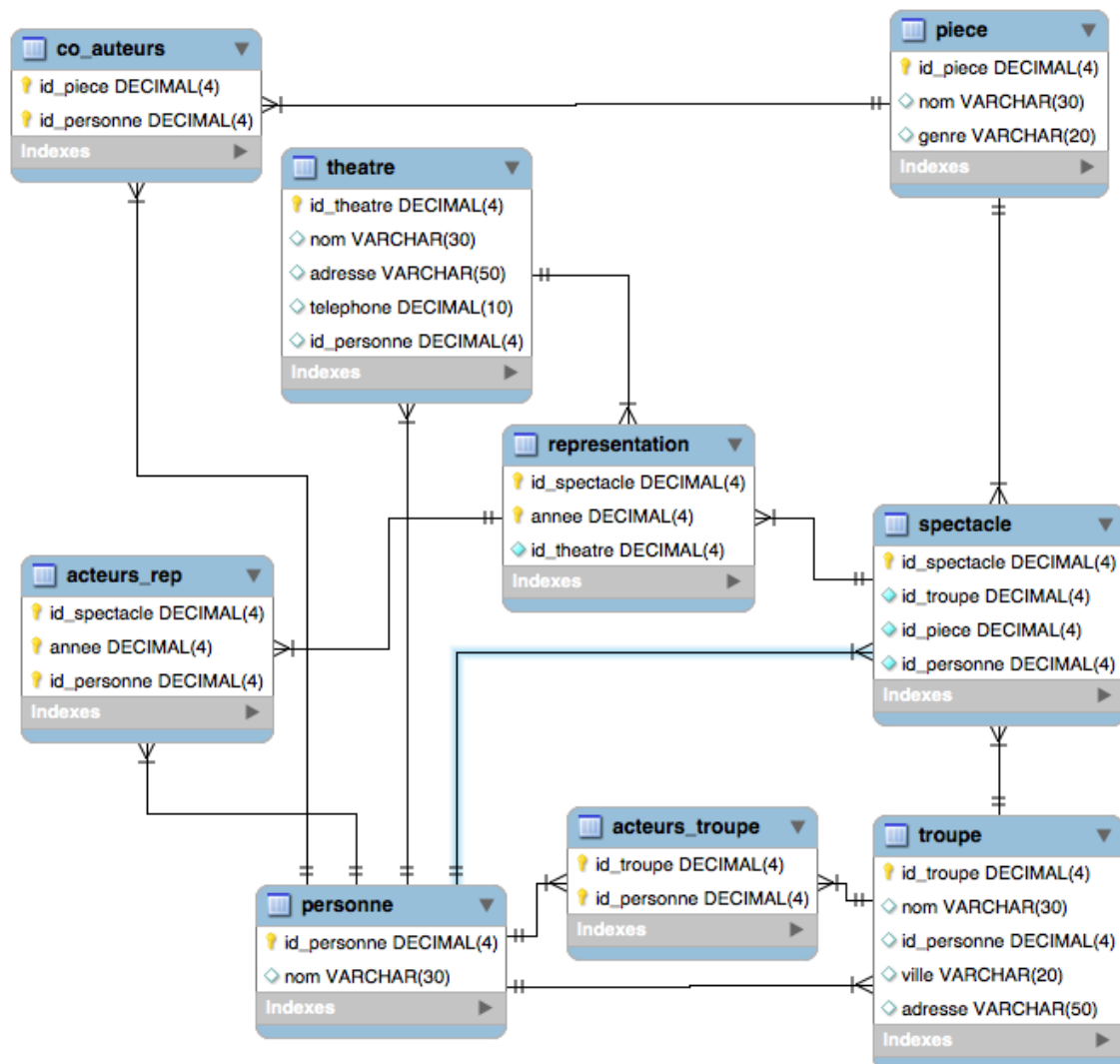
Descriptif de la base :

Un spectacle correspond à une pièce de théâtre jouée par une troupe. Certains spectacles sont joués plusieurs années de suite par la même troupe, dans des théâtres différents. On parle alors de représentations. Plus exactement, certaines informations seront enregistrées une seule fois par spectacle: le metteur en scène, la troupe, les infos concernant la pièce elle-même,... alors que d'autres seront propres au spectacle d'une année donnée (représentation) : le théâtre, en tant que lieu de représentation, l'appréciation, le ou les principaux acteurs. La description d'une pièce de théâtre comprend les attributs suivants : son identifiant (`id_pie`), son nom, son ou ses auteurs et son genre.

Une troupe est caractérisée par un identifiant (`id_troupe`), son nom, son directeur, sa ville d'attache et son adresse. Pour chaque troupe, on retient tous les acteurs qui en font (ou en ont fait) partie.

La notion de théâtre est étendue à tout lieu de représentation, y compris les spectacles en plein air ou sous chapiteau. Tout théâtre ou lieu de représentation est caractérisé par un identifiant (`id_theatre`), son nom, son adresse, un numéro de téléphone et son directeur.

Toutes les personnes apparaissant dans cette base, en tant qu'acteur, auteur, metteur en scène ou directeur, seront identifiées par un entier (`id_personne`) et un nom.



Spectacle (id_spectacle, id_troupe, id_piece, id_personne) Représentation (id_spectacle, année, id_theatre, appreciation) Piece (id_piece, nom, genre) Troupe (id_troupe, nom, id_personne, ville, adresse) Theatre(id_theatre, nom, adresse, téléphone, id_personne) Co_Auteurs (id_piece, id_personne) Acteurs_Rep (id_spectacle, année, id_personne) Acteurs_Troupe(id_troupe, id_personne) Personne (id_personne, nom)	
<pre>create table personne (id_personne numeric(4), nom varchar(30), constraint pers_key Primary Key (id_personne));</pre>	<pre>create table piece (id_piece numeric(4), nom varchar(30), genre varchar(20), constraint piece_key Primary Key (id_piece));</pre>
<pre>create table theatre(id_theatre numeric(4), nom varchar(30), adresse varchar(50), telephone numeric(10), id_personne numeric(4), constraint theatre_key Primary Key(id_theatre), constraint id_personne_exists foreign key(id_personne) references personne);</pre>	<pre>create table acteurs_troupe (id_troupe numeric(4) NOT NULL, id_personne numeric(4), constraint acteurs_tr_key Primary Key (id_troupe, id_personne), constraint id_troupe_exists foreign key(id_troupe) references troupe, constraint id_personne_exists foreign key(id_personne) references personne);</pre>
<pre>create table troupe (id_troupe numeric(4), nom varchar(30), id_personne numeric(4), ville varchar(20), adresse varchar(50), constraint troupe_key Primary Key (id_troupe), constraint id_personne_exists foreign key(id_personne) references personne);</pre>	<pre>create table acteurs_rep (id_spectacle numeric(4) NOT NULL, annee numeric(4) NOT NULL, id_personne numeric(4), constraint acteurs_rep_key Primary Key (id_spectacle, annee, id_personne), constraint id_spect_annee_exists foreign key(id_spectacle, annee) references representation, constraint id_personne_exists foreign key(id_personne) references personne);</pre>
<pre>create table spectacle (id_spectacle numeric(4) NOT NULL, id_troupe numeric(4) NOT NULL, id_piece numeric(4) NOT NULL, id_personne numeric(4) NOT NULL, constraint spect_key Primary Key (id_spectacle), constraint id_piece_exists foreign key(id_piece) references piece, constraint id_troupe_exists foreign key(id_troupe) references troupe, constraint id_personne_exists foreign key(id_personne) references personne);</pre>	
<pre>create table representation(id_spectacle numeric(4) NOT NULL, annee numeric(4) NOT NULL, id_theatre numeric(4) NOT NULL, constraint rep_key Primary Key (id_spectacle, annee), constraint id_spectacle_exists foreign key(id_spectacle) references spectacle, constraint id_theatre_exists foreign key(id_theatre) references theatre);</pre>	
<pre>create table co_auteurs (id_piece numeric(4), id_personne numeric(4), constraint co_key Primary Key (id_piece, id_personne), constraint id_piece_exists foreign key(id_piece) references piece, constraint id_personne_exists foreign key(id_personne) references personne);</pre>	

Noms des membres du groupe :

.....
.....

Question : donner le type de squelettes correspondant aux requêtes ci-dessous

Donner la liste des représentations (nom de la pièce et du théâtre) vues en 2005.

.....
.....

Donner le nom de toutes les troupes disponibles accompagné du nom de leur directeur.

.....
.....

Donner les dates des représentations de « Duel ».

.....
.....

Donner l'identifiant, le nom et la ville de provenance des troupes n'ayant donné aucune représentation.

.....
.....

Donner la liste des genres (sans doublon) de pièces disponibles.

.....
.....

Combien y a t'il de pièces de chaque genre.

.....
.....

Question : Pour chacun des patrons suivant, proposez deux questions permettant d'interroger la base de données

Patron 1 - Projection : SELECT attributs FROM table

Question 1

.....
.....
.....
.....

Question 2

.....
.....
.....
.....

Patron 2 - Implication de plusieurs entités : SELECT attributs FROM table1 join table2

Question 1

.....
.....
.....
.....

Question 2

.....
.....
.....
.....

Patron 3.1 - Restriction simple : SELECT attributs FROM tables WHERE conditions

Question 1

.....
.....
.....
.....
.....

Question 2

.....
.....
.....
.....
.....

Patron 3.2 - Patron 3-2 - Restriction complexe : SELECT attributs FROM tables WHERE conditions avec sous-requête

Question 1

.....
.....
.....
.....
.....

Question 2

.....
.....
.....
.....
.....

Patron 4 - Fonction de groupes ou d'agrégation: SELECT attributs, fonction(attribut) FROM tables GROUP BY ...

Question 1

.....
.....
.....
.....
.....

Question 2

.....
.....
.....
.....
.....