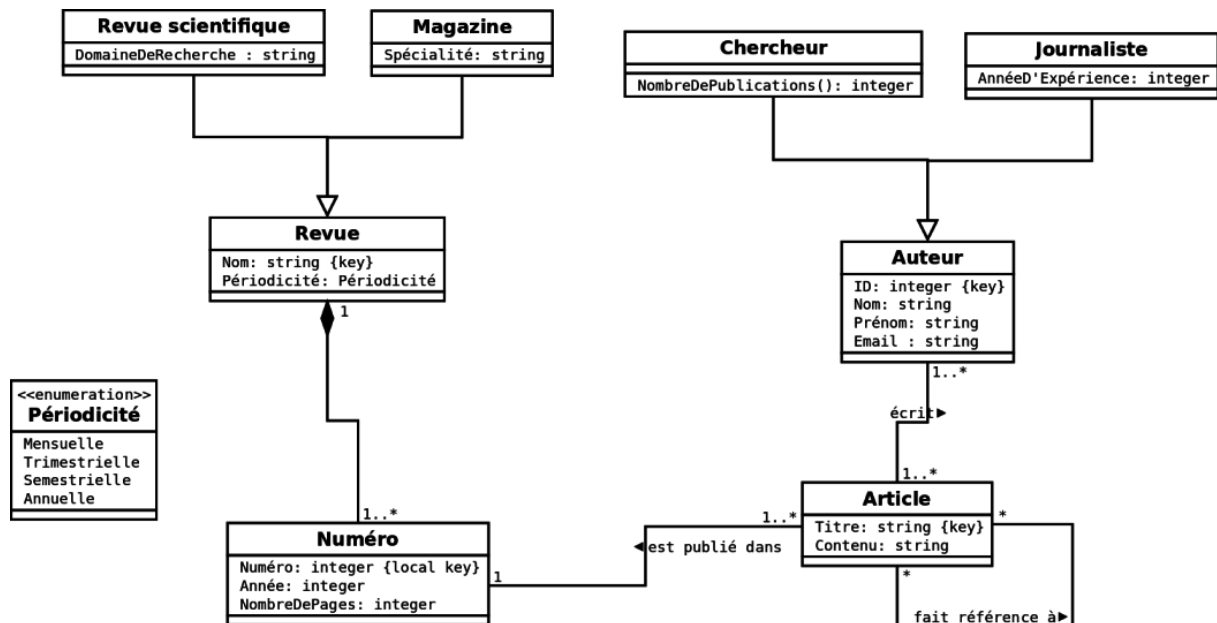


VUES ET GESTION DES DROITS – DEVOIRS

Articles

Une entreprise de presse souhaite avoir une base de données pour gérer l'ensemble des revues qu'elle publie. Le modèle UML suivant lui a été proposé afin de modéliser cette base.



1. Établissez un modèle relationnel correspondant à ce modèle UML.

On trouve que dans le dessin, on a 2 héritages. Car les 2 héritages sont presque complets, et ils ne sont pas abstraites, on choisit le passage par classe mère.

De plus, les classes mères ont des associations avec les autres tables, par conséquent, le passage par référence et le passage par filles ne sont des bons choix.

Le MLD est comme ci-dessous :

Revue (#Nom : string, Périodicité : {Mensuelle|Trimestrielle|Semestrielle|Annuelle},
Domaine_recherche : string, Spécialité : string, Type : {'Revue scientifique', 'Magazine'})
AVEC

Type = 'Revue scientifique', alors Spécialité NULL

Type = 'Magazine', alors Domaine_recherche NULL

Numéro (#Nom_revue => Revue(Nom), #Numéro : integer, Année : integer,
NombreDePages : integer)

AVEC

Numéro LOCAL KEY

PROJECTION (Revue, Nom) = PROJECTION (Numéro, Nom_revue)

Article (#Titre : string, Contenu : string, NomRevue => Numéro(Nom_revue), Numéro => Numéro(Numéro))
 AVEC
 NumRevue NOT NULL AND Numéro NOT NULL
 PROJECTION (Numéro, Nom_revue, Numéro) = PROJECTION (Article, NomRevue, Numéro)

Référence_article (#TitreRef1 => Article(Titre), (#TitreRef2 => Article(Titre))

Auteur (#ID : integer, Nom : string, Prénom : string, Email : string, AnnéeExpérience : integer, Type : {'Chercheur', 'Journaliste'})
 AVEC
 Type = 'Chercheur', alors AnnéeExpérience NULL

Écrit (#AuteurID => Auteur(ID), #ArticleTitre => Article(Titre))

PROJECTION (Auteur, ID) = PROJECTION (Écrit, AuteurID)
 PROJECTION (Auteur, Titre) = PROJECTION (Écrit, ArticleTitre)

2. En se basant sur le modèle relationnel que vous avez produit, écrivez les requêtes SQL LDD permettant de créer les tables dans une base de données.

La création des tables :

```

1  --Revue (#Nom : string, Périodicité : {Mensuelle|Trimestrielle|Semestrielle|Annuelle}, Domaine_recherche : string, Spécialité : string,
2  --Type : {'Revue scientifique', 'Magazine'})
3
4  create table Revue(
5  nom VARCHAR(50) PRIMARY KEY,
6  Periodicite VARCHAR(50) CHECK (Periodicite IN ('mensuelle', 'trimestrielle', 'semestrielle', 'annuelle')),
7  Domaine_recherche VARCHAR(50),
8  Spécialite VARCHAR(50),
9  Type CHAR(2) CHECK (Type IN ('R', 'M'))
10 );
11
12 --Numéro (#Nom_revue => Revue(Nom), #Numéro : integer, Année : integer, NombreDePages : integer)
13 CREATE TABLE Numero(
14 NomRevue VARCHAR(50),
15 Numero INTEGER,
16 Année INTEGER,
17 Nombre_pages integer,
18 primary key (NomRevue, Numero),
19 foreign key (NomRevue) references Revue(Nom)
20 );
21
22 --Article (#Titre : string, Contenu : string, NomRevue => Numéro(Nom_revue), Numéro => Numéro(Numéro))
23
24 create table Article(
25 Titre varchar(50) primary key,
26 Contenu varchar(50),
27 NomRevue varchar(50),
28 Numero integer,
29 foreign key (NomRevue, Numero) references Numero(NomRevue, Numero)
30 );
31
32 --Référence_article (#TitreRef => Article(Titre), (#TitreRef => Article(Titre))
33
34 create table Reference(
35 TitreRef1 varchar(50) references Article(Titre),
36 titreRef2 varchar(50) references Article(Titre),
37 primary key (TitreRef1, TitreRef2)
38 );

```

```

39
40 --Auteur (#ID : integer, Nom : string, Prénom : string, Email : string, AnnéeExpérience : integer, Type : {'Chercheur', 'Journaliste'})
41
42 create table Auteur(
43 ID integer primary key,
44 Nom varchar(50),
45 Prenom varchar(50),
46 Email varchar(50),
47 AnneeExperience integer,
48 Type char(2) check (Type in ('C', 'J'))
49 );
50
51 --Écrit (#AuteurID => Auteur(ID), #ArticleTitre => Article(Titre))
52
53 create table Ecrit(
54 AuteurID integer references Auteur(ID),
55 ArticleTitle references Article(Titre),
56 primary key (AuteurID, ArticleTitle)
57 );

```

La création des Views :

```

58
59 --VIEW
60 create view RevueScientifique as
61 select *
62 from revue
63 where Type = 'R';
64
65 create view Magazine as
66 select *
67 from revue
68 where Type = 'M';
69
70 create view Journaliste as
71 select *
72 from Auteur
73 where Type = 'J';
74
75 create view Chercheur as
76 select *
77 from Auteur
78 where Type = 'C';

```

Insert des données :

```

87 --INSERT
88 INSERT INTO Revue (Nom, Periodicite, Domaine_recherche, Specialite, Type) VALUES ('karl', 'semestrielle', 'CS', 'FDD', 'M') ;
89 INSERT INTO Revue (Nom, Periodicite, Domaine_recherche, Specialite, Type) VALUES ('anna', 'mensuelle', 'CS', 'Isri', 'R') ;
90 INSERT INTO Revue (Nom, Periodicite, Domaine_recherche, Specialite, Type) VALUES ('coco', 'annuelle', 'CS', 'Itie', 'R') ;
91 INSERT INTO Revue (Nom, Periodicite, Domaine_recherche, Specialite, Type) VALUES ('toto', 'annuelle', 'QR', 'Iyyy', 'M') ;
92 INSERT INTO Revue (Nom, Periodicite, Domaine_recherche, Specialite, Type) VALUES ('gaga', 'annuelle', 'QR1', 'Iyy1y', 'R') ;
93 INSERT INTO Revue (Nom, Periodicite, Domaine_recherche, Specialite, Type) VALUES ('momo', 'annuelle', 'QR2', 'Iyy2y', 'M') ;
94
95 select * from revue;
96
97 INSERT INTO numero (Nomrevue, numero, annee, NOMBRE_PAGES) VALUES ('karl', '123', '1998', '150') ;
98 INSERT INTO numero (Nomrevue, numero, annee, NOMBRE_PAGES) VALUES ('anna', '124', '1995', '14') ;
99 INSERT INTO numero (Nomrevue, numero, annee, NOMBRE_PAGES) VALUES ('coco', '888', '2013', '380') ;
100 INSERT INTO numero (Nomrevue, numero, annee, NOMBRE_PAGES) VALUES ('karl', '321', '2008', '151') ;
101 INSERT INTO numero (Nomrevue, numero, annee, NOMBRE_PAGES) VALUES ('coco', '999', '2017', '370') ;
102 INSERT INTO numero (Nomrevue, numero, annee, NOMBRE_PAGES) VALUES ('coco', '333', '2003', '388') ;
103
104 select * from numero;
105
106 INSERT INTO article (titre, contenu, nomrevue, numero) VALUES ('how to make one PC', 'make one PC', 'karl', '123') ;
107 INSERT INTO article (titre, contenu, nomrevue, numero) VALUES ('data visualisation', 'python and R', 'karl', '321') ;
108 INSERT INTO article (titre, contenu, nomrevue, numero) VALUES ('reseau centralisation', 'reduire and centrage', 'anna', '124') ;
109
110 select * from article;
111
112 INSERT INTO reference (titreref1, titreref2) VALUES ('how to make one PC', 'data visualisation') ;
113 INSERT INTO reference (titreref1, titreref2) VALUES ('reseau centralisation', 'how to make one PC') ;
114
115 select * from reference;
116
117 INSERT INTO auteur (ID, nom, prenom, email, anneeexperience, Type) VALUES ('111', 'HU', 'Jingyi', '111@etu.utc.fr', '10', 'C') ;
118 INSERT INTO auteur (ID, nom, prenom, email, anneeexperience, Type) VALUES ('222', 'ZHANG', 'Haozhou', '222@etu.utc.fr', '2', 'J') ;
119 INSERT INTO auteur (ID, nom, prenom, email, anneeexperience, Type) VALUES ('333', 'LI', 'Shilun', '333@etu.utc.fr', '11', 'C') ;
120
121 select * from auteur;
122
123 INSERT INTO ecrit (auteurid, articletitle) VALUES ('111', 'data visualisation') ;
124 INSERT INTO ecrit (auteurid, articletitle) VALUES ('222', 'how to make one PC') ;
125 INSERT INTO ecrit (auteurid, articletitle) VALUES ('333', 'reseau centralisation') ;
126

```

Les tables

Revue:

	⚡ NOM	⚡ PERIODICITE	⚡ DOMAINE_RECHERCHE	⚡ SPECIALITE	⚡ TYPE
1	karl	semestrielle	CS	FDD	M
2	anna	mensuelle	CS	Isri	R
3	coco	annuelle	CS	Itie	R
4	toto	annuelle	QR	Iyyy	M
5	gaga	annuelle	QR1	Iyy1y	R
6	momo	annuelle	QR2	Iyy2y	M

Numero:

	⚡ NOMREVUE	⚡ NUMERO	⚡ ANNEE	⚡ NOMBRE_PAGES
1	karl	123	1998	150
2	anna	124	1995	14
3	coco	888	2013	380
4	karl	321	2008	151
5	coco	999	2017	370
6	coco	333	2003	388

Article :

	⚡ TITRE	⚡ CONTENU	⚡ NOMREVUE	⚡ NUMERO
1	how to make one PC	make one PC	karl	123
2	data visualisation	python and R	karl	321
3	reseau centralisation	reduire and centrage	anna	124

Reference:

	⚡ TITREREF1	⚡ TITREREF2
1	how to make one PC	data visualisation
2	reseau centralisation	how to make one PC

Auteur:

	⚡ ID	⚡ NOM	⚡ PRENOM	⚡ EMAIL	⚡ ANNEEEXPERIENCE	⚡ TYPE
1	111	HU	Jingyi	111@etu.utc.fr	10	C
2	222	ZHANG	Haozhou	222@etu.utc.fr	2	J
3	333	LI	Shilun	333@etu.utc.fr	11	C

Écrit:

	AUTEURID	ARTICLETITLE
1	111	data visualisation
2	222	how to make one PC
3	333	reseau centralisation

3. À partir du modèle réalisé, écrivez les requêtes permettant de répondre aux questions ci-après.

Écrivez en **algèbre relationnel** et en **SQL** la requête permettant de trouver les **noms** et **prénoms** des auteurs des articles qui ont un **contenu non null**.

Algèbre relationnel :

R1 = RESTRICTION (Article, Contenu non null)
R2 = JOINTURE (R1, Écrit, R1. Titre = Écrit. ArticleTitle)
R3 = JOINTURE (R2, Auteur, R2.AuteurID = Auteur.ID)

```
132 | select nom, prenom
133 | from article a, écrit e, auteur at
134 | where a.contenu is not null
135 | and a.titre = e.articletitle
136 | and e.auteurid = at.id;
```

Résultat :

	NOM	PRENOM
1	HU	Jingyi
2	ZHANG	Haozhou
3	LI	Shilun

Écrivez en **algèbre relationnel** et en **SQL** la requête permettant de trouver les **titres** des articles qui ne sont **jamais** référencés.

Algèbre relationnel :

R1 = JointureExterneGauche (Article, Reference, Article.titre = Reference.titreRef2)
R2 = Restriction (R1, R1.titreRef2 is null)
R3 = Projection (R2, Titre)

SQL :

```
144 | select titre
145 | from article a left outer join reference r on a.titre = r.titreRef2
146 | where r.titreRef2 is null;
```

Résultat :

	TITRE
1	reseau centralisation

Écrivez en SQL la requête permettant de renvoyer le **nom** de toutes les revues **annuelles** avec le **nombre de numéros** qu'elles ont publié et la **somme des pages** que cela représente, sous réserve que la revue ait publié au moins **deux** numéros.

SQL :

```
159 | select nom, count(numero), sum(nombre_pages)
160 | from revue r, numero n
161 | where r.nom = n.NOMREVUE
162 | and r.periodicite = 'annuelle'
163 | group by nom
164 | having count(numero) >= 2;
```

Résultat:

	NOM	COUNT(NUMERO)	SUM(NOMBRE_PAGES)
1	coco	3	1138