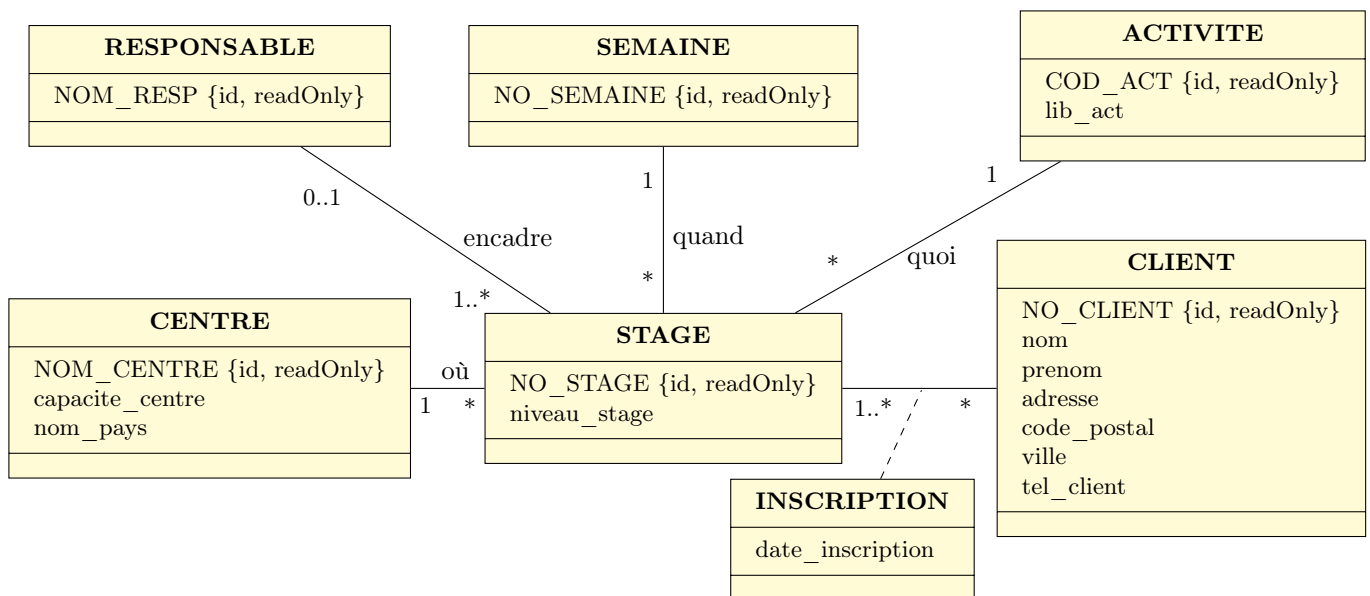


Passage d'UML et Tutorial D à SQL

Exercice 1 Traduction d'un schéma relationnel en SQL

Pour chaque schéma conceptuel apparaissant ci dessous, on demande de construire un schéma relationnel équivalent en SQL. Chaque schéma est fourni en Tutorial D. Vous penserez à préciser évidemment la ou les clés candidates, ainsi que les éventuelles clés étrangères, ainsi que les contraintes que vous jugerez nécessaires. Pour ce TP, nous nous contenterons de produire le code SQL sans l'implanter, pour le moment, sous PostgreSQL

1. Organisation des stages



```

VAR centre BASE RELATION{
    nom_centre      CHAR,
    capacite_centre INTEGER,
    nom_pays        CHAR
} KEY {nom_centre}
  
```

```

VAR semaine REAL RELATION{
    no_semaine  INTEGER
} KEY {no_semaine};
  
```

```

VAR activite BASE RELATION{
    cod_act      CHAR,
    lib_act      CHAR
} KEY {cod_act}
  
```

```

VAR stage BASE RELATION{
    no_stage      INTEGER,
    niveau_stage  CHAR,
    nom_centre    CHAR,
    no_semaine    INT,
    code_act      CHAR
} KEY {no_stage} ;
  
```

```

CONSTRAINT stage_fk_centre
    stage{nom_centre} <= centre{nom_centre};
  
```

```

CONSTRAINT stage_fk_semaine
    stage{no_semaine} <= semaine{no_semaine};
  
```

```

CONSTRAINT stages_fk_activite
    stage{cod_act} <= activite{code_act};
  
```

```

VAR encadre BASE RELATION{
    nom_resp      CHAR,
    no_stage      INTEGER
} KEY {no_stage};
  
```

```

CONSTRAINT encadre_fk1
    encadre{no_stage} <= stage{no_stage};
  
```

```

VAR client BASE RELATION{
  no_client    INTEGER,
  nom           CHAR,
  prenom        CHAR,
  adresse       CHAR,
  code_postal   CHAR,
  ville         CHAR,
  tel_client    CHAR
} KEY {no_client}

```

```

VAR inscription BASE RELATION{
  no_client    INTEGER,
  no_stage     INTEGER,
  date_inscription CHAR
}
KEY {no_client,no_stage};

CONSTRAINT inscription_fk1
  inscription{no_client} =
    client{no_client};

CONSTRAINT inscription_fk2
  inscription{no_stage} <=
    stage{no_stage};

```

2. Villes, pays et capitales :

VILLE	est capitale >	PAYS
NOM_V {id, readOnly}	1 0..1	NOM_P {id, readOnly}
nb_habit	se situe >	superficie
	* 1	popul

```

VAR ville BASE RELATION{
  nom_v    CHAR,
  nb_habit RATIONAL,
  nom_p    CHAR
}
KEY {nom_v};
CONSTRAINT ville_fk1
  ville{nom_p} <= pays{nom_p}

```

```

VAR pays BASE RELATION{
  nom_p    CHAR,
  superficie INTEGER,
  popul    INTEGER,
  capitale CHAR
}
KEY {nom_p} KEY {capitale};
CONSTRAINT pays_fk1
  pays{capitale} <= ville{nom_v}
  RENAME {nom_v AS capitale};

```

ANNEXE :

Nom	type	I	O	C	descriptif	rattachement
nb_habit	Décimal		X		Exprimé en millions d'habitants avec une précision de 1 millier d'habitants.	Ville
nom_P	Chaine(20)	X	X			Pays
nom_V	Chaine(20)	X	X			Ville
popul	Entier		X		Exprimée en millions d'habitants	Pays
superficie	Entier		X		Exprimée en Km2	Pays

TABLE 1 – Liste des Propriétés - *I = Identifiant, O = Obligatoire, C = Calculée*