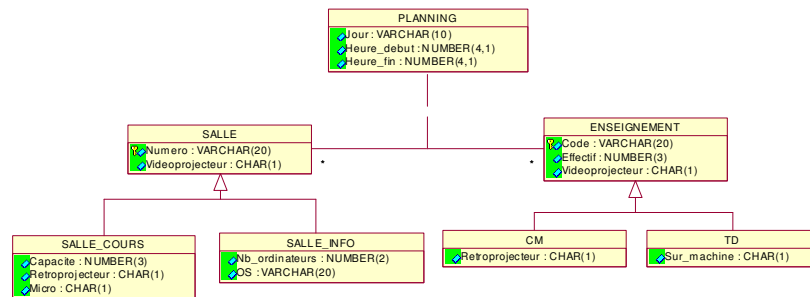


On souhaite mettre en œuvre le schéma conceptuel UML ci-dessous, représentant la gestion (simplifiée) du planning des salles de l'université, en utilisant les fonctionnalités orientées objets d'Oracle.



Classes

1. Créer les classes figurant sur le diagramme UML ci-dessus (sauf PLANNING) en tenant compte des liens d'héritage entre elles.

NB : SALLE est une classe abstraite qui n'aura aucune instance, tandis qu'ENSEIGNEMENT pourra en avoir.

Convention de nommage des types : préfixer le nom de la classe par T_ (ex. T_Salle).

2. Matérialiser la classe-association PLANNING sous forme d'un type T_Planning incluant en plus des attributs mentionnés dans le diagramme de classes deux attributs références de salle et d'enseignement, respectivement.

Tables objets

Créer trois tables objets afin de stocker les données relatives à la gestion du planning :

- Salle (stockage des instances des types T_Salle, T_Salle_cours et T_Salle_info) ;
- Enseignement (stockage des instances des types T_Enseignement, T_CM et T_TD) ;
- Planning (stockage des instances du type T_Planning).

Ne pas oublier d'assurer l'intégrité référentielle (références de Planning). Est-il possible de définir une clé primaire sur des références ?

⇒ Définir un déclencheur prenant en charge la contrainte de clé primaire sur la table Planning.

Instances

Peupler les tables Salle, Enseignement et Planning à l'aide des données ci-dessous.

Salles de cours

Numéro	Vidéoprojecteur	Capacité	Rétroprojecteur	Micro
Amphi Cassin	O	400	O	O
L231	N	80	N	N
K188	N	50	O	N

Salles informatiques

Numéro	Vidéoprojecteur	Nb_ordinateurs	OS
L219	N	13	Windows
K192	O	13	Windows/Linux

Enseignements

Code	Effectif	Vidéoprojecteur
S0INFO	350	N

Cours magistraux (CM)

Code	Effectif	Vidéoprojecteur	Rétroprojecteur
S1BDPROGCM	50	O	N
S2BDACM	25	O	N

Travaux dirigés (TD)

Code	Effectif	Vidéoprojecteur	Sur_machine
S1BDPROGTD1	50	O	N
S1BDPROGTD2	25	N	O
S2BDATD	25	N	N

Planning

Ref_salle	Ref_ens	Jour	Heure_début	Heure_fin
(L231)	(S1BDPROGCM)	Mardi	8	9,5
(K188)	(S2BDACM)	Mercredi	8	9,5
(L231)	(S2BDACM)	Mercredi	9,5	11
(Amphi Cassin)	(S0INFO)	Lundi	15	16,5
(L231)	(S1BDPROGTD1)	Mardi	9,5	11
(K192)	(S1BDPROGTD2)	Mardi	15	16,5
(K192)	(S2BDATD)	Jeudi	9,5	11

Requêtes

1. Liste de toutes les salles. Est-ce qu'un `SELECT *` fonctionne ?
2. Liste de tous les enseignements. Est-ce qu'un `SELECT *` fonctionne ?
3. État « brut » du planning. Est-ce qu'un `SELECT *` fonctionne ? Pourquoi ?
4. Planning des salles : code d'enseignement, numéro de salle, jour, heure de début et heure de fin de l'enseignement (jointure implicite).
5. Code d'enseignement et numéro de salle pour les enseignements nécessitant un vidéoprojecteur et affectés à une salle qui n'en dispose pas.
6. Nombre d'enseignements différents par salle.
7. Numéro des salles qui ne figurent pas au planning (salles libres).
8. Numéro des salles de cours. Utiliser le pseudo-attribut `OBJECT_VALUE`¹ et la fonction `IS OF()`.
9. Numéro de salle et nombre d'ordinateurs des salles informatiques. Pour accéder aux attributs d'une sous-classe, utiliser la fonction `TREAT()`, le pseudo-attribut `OBJECT_VALUE` et la notation pointée².
10. Numéro de salle et nombre d'ordinateurs de toutes les salles. Conclusion ?
11. Code d'enseignement, besoin de rétroprojecteur (pour l'enseignement), numéro de salle et disponibilité de rétroprojecteur (dans la salle) pour les cours magistraux se déroulant dans les salles de cours. Exploiter la fonction `DEREF()` combinée aux fonctions `TREAT()` et `IS OF()`. Cela fonctionne-t-il correctement ?
12. Idem, mais en exploitant la fonction `SYS_TYPEID()` au lieu de `IS OF()`.
13. Code d'enseignement, effectif de l'enseignement, numéro de salle, nombre d'ordinateurs x 2 (la capacité d'une salle informatique est supposée égale au double du nombre de machines) pour les travaux dirigés se déroulant en salle informatique et pour lesquels la capacité de la salle (nombre d'ordinateurs x 2) est inférieure à l'effectif de l'enseignement (pas assez de place !).

¹ http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/pseudocolumns006.htm#sthref830

² http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/functions206.htm#SQLRF06148

Correction

-- Classes

```
CREATE TYPE T_Salle AS OBJECT(  
    Numero VARCHAR(20),  
    Videoprojecteur CHAR(1))  
NOT INSTANTIABLE  
NOT FINAL  
/
```

```
CREATE TYPE T_Salle_cours UNDER T_Salle(  
    Capacite NUMBER(3),  
    Retroprojecteur CHAR(1),  
    Micro CHAR(1))  
/
```

```
CREATE TYPE T_Salle_info UNDER T_Salle(  
    Nb_ordinateurs NUMBER(2),  
    OS VARCHAR(20))  
/
```

```
CREATE TYPE T_Enseignement AS OBJECT(  
    Code VARCHAR(20),  
    Effectif NUMBER(3),  
    Videoprojecteur CHAR(1))  
NOT FINAL  
/
```

```
CREATE TYPE T_CM UNDER T_Enseignement(  
    Retroprojecteur CHAR(1))  
/
```

```
CREATE TYPE T_TD UNDER T_Enseignement(  
    Sur_machine CHAR(1))  
/
```

```
CREATE TYPE T_Planning AS OBJECT(  
    Ref_salle REF T_Salle,  
    Ref_ens REF T_Enseignement,  
    Jour VARCHAR(10),  
    Heure_debut NUMBER(4,1),  
    Heure_fin NUMBER(4,1))  
/
```

-- Tables

```
CREATE TABLE Salle OF T_Salle(  
    CONSTRAINT Salle_pk PRIMARY KEY(Numero));
```

```
CREATE TABLE Enseignement OF T_Enseignement(  
    CONSTRAINT Ens_pk PRIMARY KEY(Code));
```

```
CREATE TABLE Planning OF T_Planning(  
-- CONSTRAINT Planning_pk PRIMARY KEY(Ref_salle, Ref_ens) impossible !  
    CONSTRAINT Planning_ref_salle Ref_salle REFERENCES Salle,  
    CONSTRAINT Planning_ref_salle_null CHECK (Ref_salle IS NOT NULL),  
    CONSTRAINT Planning_ref_ens Ref_ens REFERENCES Enseignement,  
    CONSTRAINT Planning_ref_ens_null CHECK (Ref_ens IS NOT NULL));
```

```

CREATE OR REPLACE TRIGGER trig_pk_planning
BEFORE INSERT OR UPDATE ON PLANNING
FOR EACH ROW

DECLARE
    n INTEGER;
    unicity EXCEPTION;

BEGIN
    SELECT COUNT(*) INTO n FROM PLANNING p
    WHERE p.ref_salle = :NEW.ref_salle AND p.ref_ens = :NEW.ref_ens;

    IF n > 1 THEN
        RAISE unicity;
    END IF;

EXCEPTION
    WHEN unicity THEN
        RAISE_APPLICATION_ERROR(-20502, 'Clé primaire existante !');

END;
/
SHOW ERRORS

```

-- Instances

```

INSERT INTO Salle VALUES(
    T_Salle_cours('Amphi Cassin', 'O', 400, 'O', 'O'));
INSERT INTO Salle VALUES(
    T_Salle_cours('L231', 'N', 80, 'N', 'N'));
INSERT INTO Salle VALUES(
    T_Salle_cours('K188', 'N', 50, 'O', 'N'));
INSERT INTO Salle VALUES(
    T_Salle_info('L219', 'O', 12, 'Windows'));
INSERT INTO Salle VALUES(
    T_Salle_info('K192', 'N', 12, 'Windows/Linux'));

INSERT INTO Enseignement VALUES(
    'S0INFO', 500, 'N');
INSERT INTO Enseignement VALUES(
    T_CM('S1BDPROGCM', 50, 'O', 'N'));
INSERT INTO Enseignement VALUES(
    T_CM('S2BDACM', 25, 'O', 'N'));
INSERT INTO Enseignement VALUES(
    T_TD('S1BDPROGTD1', 50, 'O', 'N'));
INSERT INTO Enseignement VALUES(
    T_TD('S1BDPROGTD2', 25, 'N', 'O'));
INSERT INTO Enseignement VALUES(
    T_TD('S2BDATD', 25, 'N', 'N'));

INSERT INTO Planning VALUES(
    (SELECT REF(s) FROM Salle s WHERE s.Numero = 'L231'),
    (SELECT REF(e) FROM Enseignement e WHERE e.Code = 'S1BDPROGCM'),
    'Mardi', 8, 9.5);

INSERT INTO Planning VALUES(
    (SELECT REF(s) FROM Salle s WHERE s.Numero = 'K188'),
    (SELECT REF(e) FROM Enseignement e WHERE e.Code = 'S2BDACM'),
    'Mercredi', 8, 9.5);

```

```

INSERT INTO Planning VALUES(
    (SELECT REF(s) FROM Salle s WHERE s.Numero = 'L231'),
    (SELECT REF(e) FROM Enseignement e WHERE e.Code = 'S2BDACM'),
    'Mercredi', 9.5, 11);

INSERT INTO Planning VALUES(
    (SELECT REF(s) FROM Salle s WHERE s.Numero = 'Amphi Cassin'),
    (SELECT REF(e) FROM Enseignement e WHERE e.Code = 'S0INFO'),
    'Lundi', 15, 16.5);

INSERT INTO Planning VALUES(
    (SELECT REF(s) FROM Salle s WHERE s.Numero = 'L231'),
    (SELECT REF(e) FROM Enseignement e WHERE e.Code = 'S1BDPROGTD1'),
    'Mardi', 9.5, 11);

INSERT INTO Planning VALUES(
    (SELECT REF(s) FROM Salle s WHERE s.Numero = 'K192'),
    (SELECT REF(e) FROM Enseignement e WHERE e.Code = 'S1BDPROGTD2'),
    'Mardi', 15, 16.5);

INSERT INTO Planning VALUES(
    (SELECT REF(s) FROM Salle s WHERE s.Numero = 'K192'),
    (SELECT REF(e) FROM Enseignement e WHERE e.Code = 'S2BDATD'),
    'Jeudi', 9.5, 11);

```

-- Requetes

```

-- 1
SELECT VALUE(s) FROM Salle s;

-- 2
SELECT VALUE(e) FROM Enseignement e;

-- 3
SELECT * FROM Planning;

-- 4
SELECT p.Ref_ens.Code, p.Ref_salle.Numero, Jour, Heure_debut, Heure_fin
FROM Planning p;

-- 5
SELECT p.Ref_ens.Code, p.Ref_salle.Numero
FROM Planning p
WHERE p.Ref_ens.Videoprojecteur = 'O'
AND p.Ref_salle.Videoprojecteur = 'N';

-- 6
SELECT p.Ref_salle.Numero, COUNT(DISTINCT p.Ref_ens)
FROM Planning p
GROUP BY p.Ref_salle.Numero;

-- 7
SELECT Numero FROM Salle s1
WHERE (SELECT REF(s2) FROM Salle s2 WHERE s2.Numero = s1.Numero)
NOT IN (SELECT Ref_salle FROM Planning);

-- 8
SELECT s.Numero
FROM Salle s
WHERE OBJECT_VALUE IS OF(T_Salle_cours);

```

```

-- 9
SELECT s.Numero, TREAT(OBJECT_VALUE AS T_Salle_info).Nb_ordinateurs
FROM Salle s
WHERE OBJECT_VALUE IS OF(T_Salle_info);

-- 10
SELECT s.Numero, TREAT(OBJECT_VALUE AS T_Salle_info).Nb_ordinateurs
FROM Salle s;

-- 11
SELECT p.Ref_ens.Code, TREAT(DEREF(p.Ref_ens) AS T_CM).Retroprojecteur,
p.Ref_salle.Numero, TREAT(DEREF(p.Ref_salle) AS T_Salle_cours).Retroprojecteur
FROM Planning p
WHERE DEREF(p.Ref_ens) IS OF(T_CM)
AND DEREF(p.Ref_salle) IS OF(T_Salle_cours);

-- 12
SELECT p.Ref_ens.Code, TREAT(DEREF(p.Ref_ens) AS T_CM).Retroprojecteur,
      p.Ref_salle.Numero,
      TREAT(DEREF(p.Ref_salle) AS T_Salle_cours).Retroprojecteur
FROM Planning p
WHERE SYS_TYPEID(DEREF(p.Ref_ens))
      = (SELECT TYPEID FROM USER_TYPES WHERE TYPE_NAME = 'T_CM')
AND SYS_TYPEID(DEREF(p.Ref_salle))
      = (SELECT TYPEID FROM USER_TYPES WHERE TYPE_NAME = 'T_SALLE_COURS');

-- 12 (bidouille)
SELECT p.Ref_ens.Code, TREAT(DEREF(p.Ref_ens) AS T_CM).Retroprojecteur,
      p.Ref_salle.Numero,
      TREAT(DEREF(p.Ref_salle) AS T_Salle_cours).Retroprojecteur
FROM Planning p
WHERE SYS_TYPEID(DEREF(p.Ref_ens))
      = SYS_TYPEID(TREAT(DEREF(p.Ref_ens) AS T_CM))
AND SYS_TYPEID(DEREF(p.Ref_salle))
      = SYS_TYPEID(TREAT(DEREF(p.Ref_salle) AS T_Salle_cours));

-- 13
SELECT p.Ref_ens.Code, p.Ref_ens.Effectif, p.Ref_salle.Numero,
      TREAT(DEREF(p.Ref_salle) AS T_Salle_info).Nb_ordinateurs * 2
FROM Planning p
WHERE SYS_TYPEID(DEREF(p.Ref_ens))
      = (SELECT TYPEID FROM USER_TYPES WHERE TYPE_NAME = 'T_TD')
AND SYS_TYPEID(DEREF(p.Ref_salle))
      = (SELECT TYPEID FROM USER_TYPES WHERE TYPE_NAME = 'T_SALLE_INFO')
AND TREAT(DEREF(p.Ref_salle) AS T_Salle_info).Nb_ordinateurs * 2
      < p.Ref_ens.Effectif;

```