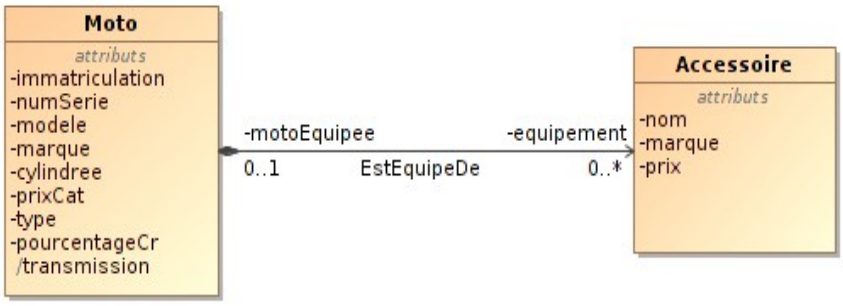
Compte rendu Objet Relationnel
TP Motos

Question n°1 : Diagramme de classes UML



Une moto possède de nombreux attributs, dont un attribut dérivé (transmission).
Elle peut être équipée de 0 ou plusieurs accessoires, qui ont un nom, une marque et un prix.
La destruction de la moto entraîne la destruction des accessoires : on a une composition.
La navigabilité est définie pour pouvoir récupérer facilement les accessoires d'une moto sans se soucier de retrouver une moto en fonction de l'accessoire.

Question n°2 :

Pour implanter ce modèle sous forme relationnelle-objet, il est préférable de créer une table Moto dans laquelle on imbrique une table imbriquée des Accessoires, permettant ainsi la navigabilité évidente Moto → Accessoire.

--On crée un type pour les accessoires possédant 3 attributs
```
CREATE TYPE accessoire_type AS OBJECT (
  nom VARCHAR2(40),
  marque VARCHAR2(40),
  prix FLOAT
)
/
```

--Création de la table imbriquée d'accessoires
```
CREATE OR REPLACE TYPE accessoire_ntab_type AS TABLE OF
accessoire_type
/
```

```
--Création d'un type pour les motos dont avec tous les attributs (y compris les accessoires)
--On ajoute également la méthode get_transmission qui pourra être appelée pour calculer la
--transmission d'une moto
CREATE TYPE moto_type AS OBJECT(
immatriculation VARCHAR2(20),
numSerie INTEGER,
modele VARCHAR2(40),
marque VARCHAR2(40),
cylindree INTEGER,
prixCat FLOAT,
type VARCHAR2(20),
pourcentageCr FLOAT,
accessoire accessoire_ntab_type,
MEMBER FUNCTION get_transmission RETURN VARCHAR2
);
/

--On crée une méthode qui renvoie un VARCHAR2 en faisant de simple tests grâce à un IF
CREATE OR REPLACE TYPE BODY moto_type AS
MEMBER FUNCTION get_transmission RETURN VARCHAR2 IS
     resultat VARCHAR2(45);
BEGIN
     IF self.type = 'SPORTIF' THEN
          resultat := 'transmission finale par chaîne';
     ELSIF self.type = 'CUSTOM' and self.cylindree <= 250 THEN
          resultat := 'transmission finale par chaîne';
     ELSIF self.type = 'CUSTOM' and self.cylindree > 250 THEN
          resultat := 'transmission finale par courroie crantée';
     ELSE
          resultat := 'unknown';
     END IF;
return resultat;
END;
END;
/

--Enfin création de la table Moto, avec l'immatriculation comme clé primaire
--et les accessoires comme table imbriquée
CREATE TABLE moto OF moto_type
(
     CONSTRAINT pk_immat PRIMARY KEY(immatriculation)
) NESTED TABLE accessoire STORE AS tab_access
/
```

C'est tout pour la création des tables, types et méthodes.

Question n°3 : Tests

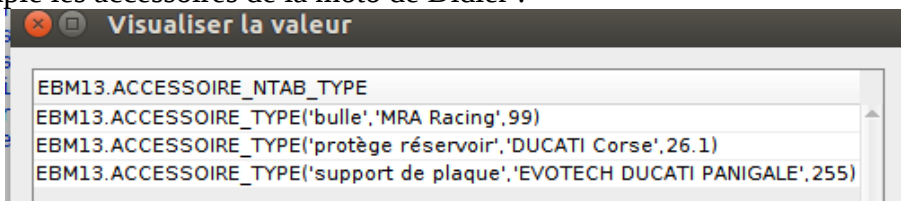
```
--On insère une par une les motos dans la table moto
INSERT INTO moto VALUES('777 ADA 95',1007,'EBR 1190
RX','EBR',1191,16998,'sportif',NULL,
                              NEW accessoire_ntab_type());
INSERT INTO moto VALUES('123 JDK 14',1008,'899
PANIGALE','DUCATI',898,15890,'sportif',NULL,
                              NEW accessoire_ntab_type());
INSERT INTO moto VALUES('145 UML 59',1009,'Electra Glide Ultra
Classic','HARLEY DAVIDSON',1449,26195,'custom',15.5,
                              NEW accessoire_ntab_type());
INSERT INTO moto VALUES('689 PHP 59',1010,'VT Shadow
125','HONDA',124,7000,'custom',0,
                              NEW accessoire_ntab_type());


--On définit une procédure permettant d'ajouter un nouvel accessoire dans la table accessoire
--imbriquée dans une moto désignée par son numéro de série
CREATE OR REPLACE PROCEDURE set_accessoire (p_nSerie VARCHAR2,
p_nom VARCHAR2, p_marque VARCHAR2 , p_prix FLOAT)
 IS
BEGIN
INSERT INTO TABLE (SELECT m.accessoire FROM moto m WHERE
m.numSerie = p_nSerie)
  VALUES (p_nom,p_marque,p_prix);
  COMMIT;
END ;
/

--Enfin on rajoute chaque accessoire en utilisant la procédure set_accessoire définie précédemment
Begin
set_accessoire('1007','amortisseur de direction','EBR', NULL);
set_accessoire('1009', 'valises', 'HARLEY DAVIDSON', NULL);
set_accessoire('1009', 'topcase', 'HARLEY DAVIDSON', NULL);
set_accessoire('1009', 'poignées chromées', NULL, 900);
set_accessoire('1009', 'repose-pieds passager à LED', NULL,
856.55);
set_accessoire('1008', 'bulle', 'MRA Racing', 99);
set_accessoire('1008', 'protège réservoir', 'DUCATI Corse',
26.10);
set_accessoire('1008', 'support de plaque', 'EVOTECH DUCATI
PANIGALE', 255);
commit;
end;
/
```

--On constate le résultat grâce à cette requête :

```
SELECT m.*, m.get_transmission() as TRANSMISSION
FROM moto m;
```

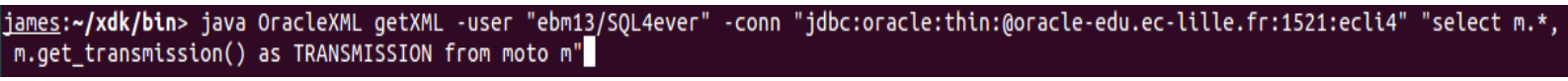
	IMMATRI...	...	MODELE	MARQUE	...	PR...	TYPE	PO...	ACCESSOIRE	TRANSMISSION
1	777 ADA 95	1007	EBR 1190 RX	EBR	1191	16998	sportif	(null)	EBM13.ACC...	transmission finale par chaîne
2	123 JDK 14	1008	899 PANIGALE	DUCATI	898	15890	sportif	(null)	EBM13.ACC...	transmission finale par chaîne
3	145 UML 59	1009	Electra Glide Ultra Classic	HARLEY DAVIDSON	1449	26195	custom	15,5	EBM13.ACC...	transmission finale par courroie cranté
4	689 PHP 59	1010	VT Shadow 125	HONDA	124	7000	custom	0	EBM13.ACC...	transmission finale par chaîne

Voici par exemple les accessoires de la moto de Didier :



Question n°4 : XML

En utilisant la commande suivante :




```
java OracleXML getXML -user "ebm13/SQL4ever" -conn
"jdbc:oracle:thin:@oracle-edu.ec-lille.fr:1521:ecli4" "select m.*,
m.get_transmission() as TRANSMISSION from moto m"
```

On obtient le fichier XML nommé *HYOT_moto.xml* présent dans l'archive du TP.
En voici un extrait pour la première moto :

```
<ROW num="1">
      <IMMATRICULATION>777 ADA 95</IMMATRICULATION>
      <NUMSERIE>1007</NUMSERIE>
      <MODELE>EBR 1190 RX</MODELE>
      <MARQUE>EBR</MARQUE>
      <CYLINDREE>1191</CYLINDREE>
      <PRIXCAT>16998</PRIXCAT>
      <TYPE>sportif</TYPE>
      <ACCESSOIRE>
         <ACCESSOIRE_ITEM>
            <NOM>amortisseur de direction</NOM>
            <MARQUE>EBR</MARQUE>
         </ACCESSOIRE_ITEM>
      </ACCESSOIRE>
      <TRANSMISSION>transmission finale par chaîne</TRANSMISSION>
  </ROW>
```