

Guillevic Yann

Guibert Rémy

Compte rendu de SAE BD 2.04

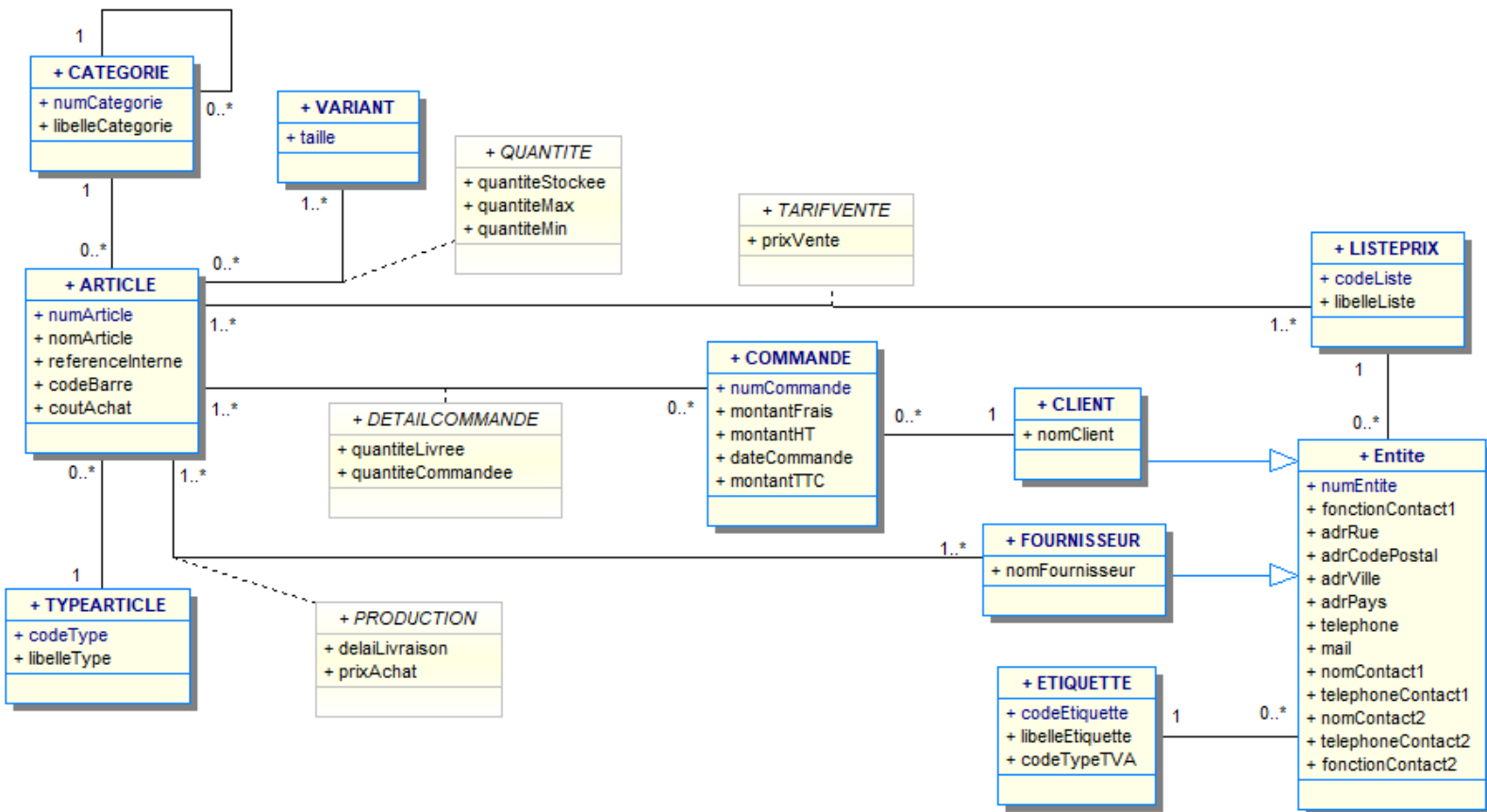
Livrable 2

Table des matières

1 Schéma Conceptuel (relationnel).....	3
2 Dictionnaire de données.....	4
3 Schéma Logique.....	5
4 Script SQL-LDD et SQL-LMD de transformation.....	8

1 Schéma Conceptuel (relationnel)

Diagramme de classes UML :



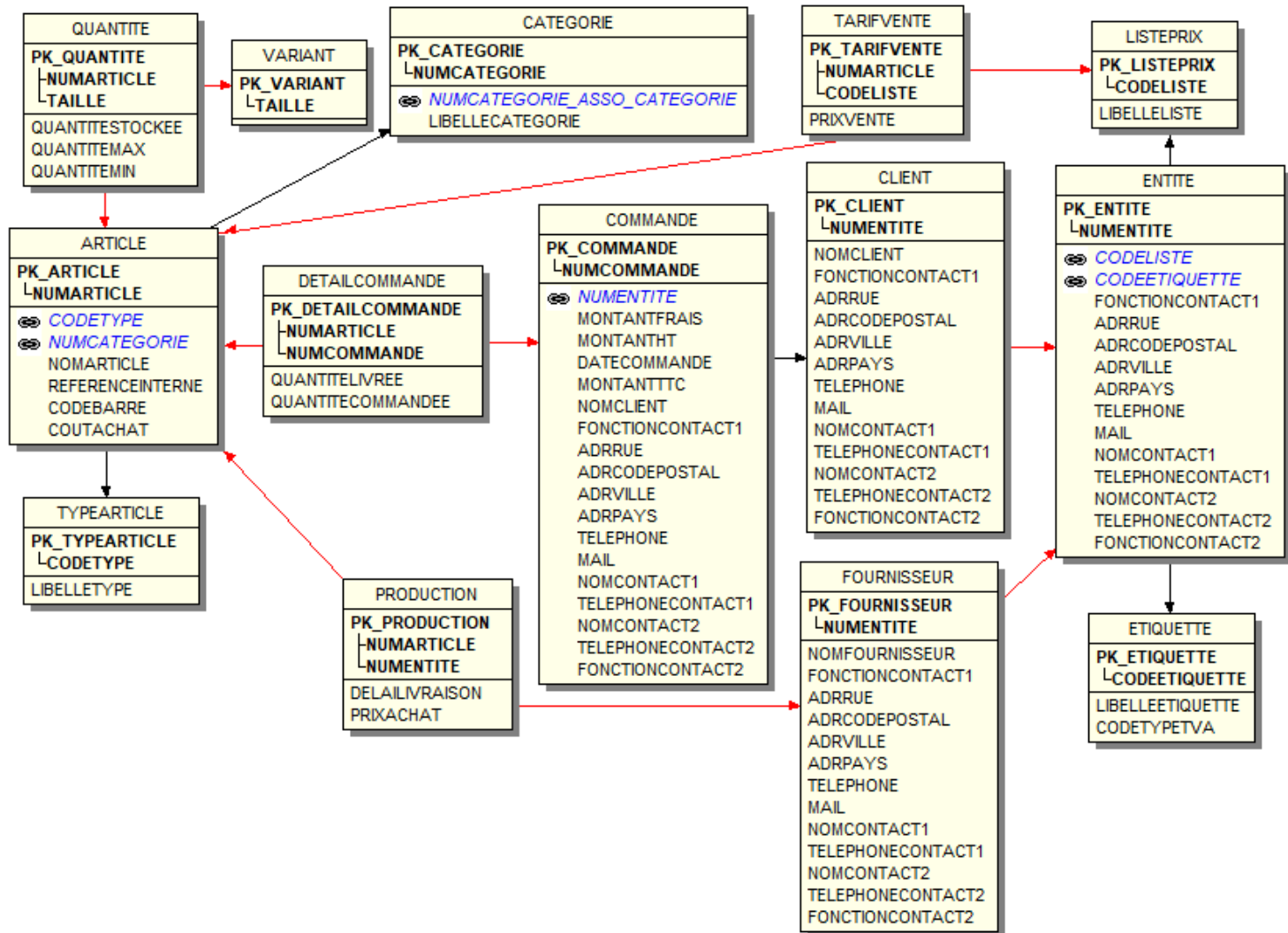
2 Dictionnaire de données

Dictionnaire de données :

Nom table	Nom attribut	Description	Type	Contrainte	Calcul
CATEGORIE					
	numCategorie	Numéro de la catégorie	Réel	!= NULL	
	numCatPere	Numéro de la catégorie père	Réel		
	libelleCategorie	Libellé de la catégorie	Chaîne variable de 40 caractères		
ARTICLE					
	numArticle	Numéro de l'article	Réel	!= NULL	
	nomArticle	Nom de l'article	Chaîne variable de 50 caractères		
	referenceInterne	Référence interne	Chaîne variable de 10 caractères		
	codeBarre	Code barre	Chaîne de 13 caractères		
	coutAchat	Coût à l'achat	Réel		
VARIANT					
	taille	Taille du produit	Entier	!= NULL	
QUANTITE					
	quantiteStockee	Quantité stockée du produit	Entier	>= 0	
	quantiteMin	Quantité minimale avant un réapprovisionnement	Entier	>= 0	
	quantiteMax	Quantité maximale stockée	Entier	> 0	
TYPEARTICLE					
	codeType	Code du type	Chaîne de 1 caractère	!= NULL	
	libelleType	Libellé du type	Chaîne variable de 40 caractères		
DETAILCOMMANDE					
	quantiteLivree	Quantité livrée	Réel		
	quantiteCommandee	Quantité commandée	Réel		
COMMANDE					
	numCommande	Numéro de la commande	Réel	!= NULL	
	montantFrais	Montant des frais	Réel		
	montantHT	Montant hors-taxes	Réel		
	dateCommande	Date de la commande	Date		
	montantTTC	Montant toutes taxes comprises	Réel		
LISTEPRIX					
	codeListe	Code de la liste	Chaîne de 1 caractère	!= NULL	
	libelleListe	Libellé de la liste	Chaîne variable de 20 caractères		
PRODUCTION					
	delaiLivraison	Delai de la livraison	Entier		
	prixAchat	Prix d'achat	Réel		
FOURNISSEUR					
	nomFournisseur	Nom du fournisseur	Chaîne variable de 40 caractères		
CLIENT					
	nomClient	Nom du client	Chaîne variable de 40 caractères		
TARIFVENTE					
	prixVente	Prix de vente	Réel		
ETIQUETTE					
	codeEtiquette	Code de l'étiquette	Chaîne de 2 caractères	!= NULL	
	libelleEtiquette	Libellé de l'étiquette	Chaîne variable de 70 caractères		
	codeTypeTVA	Code de la TVA du type	Réel		
ENTITE					
	numEntite	Numéro du client	Réel	!= NULL	
	nomContact1	Nom du contact 1	Chaîne variable de 50 caractères		
	telephoneContact1	Numéro du contact 1	Chaîne de 12 caractères		
	fonctionContact1	Fonction du contact 1	Chaîne variable de 20 caractères		
	nomContact2	Nom du contact 2	Chaîne variable de 50 caractères		
	telephoneClient2	Numéro du contact 2	Chaîne de 12 caractères		
	fonctionContact2	Fonction du contact 2	Chaîne variable de 20 caractères		
	adrRue	Nom de la rue du client	Chaîne variable de 50 caractères		
	adrCodePostal	Code postal du client	Chaîne variable de 10 caractères		
	adrVille	Ville du client	Chaîne variable de 40 caractères		
	adrPays	Pays du client	Chaîne variable de 30 caractères		
	mail	Mail du client	Chaîne variable de 60 caractères		
	telephone	Numéro de téléphone du client	Chaîne de 12 caractères		

3 Schéma Logique

Modèle Logique Relationnel :



R1 Traduction des classes d'objets

R2 Traduction des classes d'associations (1,n)

R3 Traduction des classes d'associations (n,m)

R5 Traduction de l'héritage

<p>R1 Traduction des classes d'objets</p>	<p>CATEGORIE (<u>numCategorie</u>, libelleCategorie) VARIANT (<u>taille</u>) ARTICLE (<u>numArticle</u>, nomArticle, referenceInterne, codeBarre, coutAchat) TYPEARTICLE (<u>codeType</u>, libelleType) COMMANDE (<u>numCommande</u>, montantFrais, montantHT, dateCommande, montantTTC) LISTEPRIX (<u>codeListe</u>, libelleListe) ETIQUETTE (<u>codeEtiquette</u>, libelleEtiquette, codeTypeTVA) ENTITE (<u>numEntite</u>, adrRue, adrVille, adrCodePostal, adrPays, telephone, mail, nomContact1, telephoneContact1, fonctionContact1, nomContact2, telephoneContact2, fonctionContact2) CLIENT (nomClient) FOURNISSEUR (nomFournisseur)</p>
<p>R2 Traduction des classes d'associations (1,n)</p>	<p>CATEGORIE (<u>numCategorie</u>, libelleCategorie, numCatPere#) VARIANT (<u>taille</u>) ARTICLE (<u>numArticle</u>, nomArticle, referenceInterne, codeBarre, coutAchat) TYPEARTICLE (<u>codeType</u>, libelleType, numCategorie#, codeType#) COMMANDE (<u>numCommande</u>, montantFrais, montantHT, dateCommande, montantTTC, numEntite#) LISTEPRIX (<u>codeListe</u>, libelleListe) ETIQUETTE (<u>codeEtiquette</u>, libelleEtiquette, codeTypeTVA) ENTITE (<u>numEntite</u>, adrRue, adrVille, adrCodePostal, adrPays, telephone, mail, nomContact1, telephoneContact1, fonctionContact1, nomContact2, telephoneContact2, fonctionContact2, codeListe#, codeEtiquette#) CLIENT (nomClient) FOURNISSEUR (nomFournisseur)</p>
<p>R3 Traduction des classes d'associations (n,m)</p>	<p>CATEGORIE (<u>numCategorie</u>, libelleCategorie, numCatPere#) VARIANT (<u>taille</u>) ARTICLE (<u>numArticle</u>, nomArticle, referenceInterne, codeBarre, coutAchat) TYPEARTICLE (<u>codeType</u>, libelleType, numCategorie#, codeType#) COMMANDE (<u>numCommande</u>, montantFrais, montantHT, dateCommande, montantTTC, numEntite#) LISTEPRIX (<u>codeListe</u>, libelleListe) ETIQUETTE (<u>codeEtiquette</u>, libelleEtiquette, codeTypeTVA) ENTITE (<u>numEntite</u>, adrRue, adrVille, adrCodePostal, adrPays, telephone, mail, nomContact1, telephoneContact1, fonctionContact1, nomContact2, telephoneContact2, fonctionContact2, codeListe#, codeEtiquette#) CLIENT (nomClient) FOURNISSEUR (nomFournisseur) QUANTITE (<u>numArticle#</u>, <u>taille#</u>, quantiteStockee, quantiteMin, quantiteMax) TARIFVENTE (<u>numArticle#</u>, <u>codeListe#</u>, prixVente) DETAILCOMMANDE (<u>numArticle#</u>, <u>numCommande#</u>, quantiteLivree, quantiteCommandee) PRODUCTION (<u>numArticle#</u>, <u>numEntite#</u>, delaiLivraison, prixAchat)</p>

R5
Traduction
de
l'héritage

CATEGORIE (numCategorie, libelleCategorie, numCatPere#)
 VARIANT (taille)
 ARTICLE (numArticle, nomArticle, referenceInterne, codeBarre, coutAchat)
 TYPEARTICLE (codeType, libelleType, numCategorie#, codeType#)
 COMMANDE (numCommande, montantFrais, montantHT,
 dateCommande, montantTTC, numEntite#)
 LISTEPRIX (codeListe, libelleListe)
 ETIQUETTE (codeEtiquette, libelleEtiquette, codeTypeTVA)
 ENTITE (numEntite, adrRue, adrVille, adrCodePostal, adrPays, telephone,
 mail, nomContact1, telephoneContact1, fonctionContact1, nomContact2,
 telephoneContact2, fonctionContact2, codeListe#, codeEtiquette#)
 CLIENT (nomClient, numEntite#, adrRue, adrVille, adrCodePostal, adrPays,
 telephone, mail, nomContact1, telephoneContact1, fonctionContact1,
 nomContact2, telephoneContact2, fonctionContact2, codeListe#,
 codeEtiquette#)
 FOURNISSEUR (nomFournisseur, numEntite#, adrRue, adrVille,
 adrCodePostal, adrPays, telephone, mail, nomContact1,
 telephoneContact1, fonctionContact1, nomContact2, telephoneContact2,
 fonctionContact2, codeListe#, codeEtiquette#)
 QUANTITE (numArticle#, taille#, quantiteStockee, quantiteMin,
 quantiteMax)
 TARIFVENTE (numArticle#, codeListe#, prixVente)
 DETAILCOMMANDE (numArticle#, numCommande#, quantiteLivree,
 quantiteCommandee)
 PRODUCTION (numArticle#, numEntite#, delaiLivraison, prixAchat)

4 Script SQL-LDD et SQL-LMD de transformation

-- supprimer "prix vente" de ARTICLE:

```
ALTER TABLE ARTICLE  
DROP COLUMN PRIXVENTE;
```

-- créer table variant avec taille:

```
CREATE TABLE VARIANT  
(  
    taille DECIMAL(6),  
    CONSTRAINT pk_VARIANT PRIMARY KEY (taille)  
);
```

-- créer table quantite :

```
CREATE TABLE QUANTITE  
(  
    quantiteStockee DECIMAL(6),  
    quantiteMax DECIMAL(6),  
    quantiteMin DECIMAL(6),  
    taille DECIMAL(6),  
    numArticle DECIMAL(38),  
    CONSTRAINT pk_QUANTITE PRIMARY KEY (taille,numArticle),  
    CONSTRAINT ck_QUANTITE_taille CHECK (taille>0),  
    CONSTRAINT ck_QUANTITE_numArticle CHECK (numArticle>0),  
    CONSTRAINT fk_QUANTITE_taille FOREIGN KEY (taille) REFERENCES VARIANT  
    (taille),  
    CONSTRAINT fk_QUANTITE_numArticle FOREIGN KEY (numArticle)  
    REFERENCES ARTICLE (NUMARTICLE)  
);
```


-- retirer nomClient dans ENTITE :

```
ALTER TABLE CLIENT  
DROP COLUMN NOMCLIENT;
```

-- modifier le nom de CLIENT en ENTITE :

```
ALTER TABLE CLIENT  
RENAME TO ENTITE;
```

-- supprimer la liaison entre COMMANDE et ENTITE :

```
ALTER TABLE COMMANDE  
DROP CONSTRAINT fk_COMMANDE_CLIENT;
```

-- Gestion de la table ENTITE :

"supprimer les anciennes contraintes de ENTITE pour les recréer avec le bon nom"

```
ALTER TABLE ENTITE  
DROP CONSTRAINT fk_CLIENT_ETIQUETTE;  
ALTER TABLE ENTITE  
ADD CONSTRAINT FK_ENTITE_CODEETIQUETTE FOREIGN KEY  
(CODEETIQUETTE) REFERENCES ETIQUETTE (CODEETIQUETTE);
```

```
ALTER TABLE ENTITE  
DROP CONSTRAINT fk_CLIENT_LISTEPRIX;  
ALTER TABLE ENTITE  
ADD CONSTRAINT FK_ENTITE_CODELISTE FOREIGN KEY (CODELISTE)  
REFERENCES LISTEPRIX (CODELISTE);
```

"on renomme NUMCLIENT en NUMENTITE"

ALTER TABLE ENTITE

RENAME COLUMN NUMCLIENT TO NUMENTITE;

ALTER TABLE ENTITE

DROP CONSTRAINT pk_CLIENT;

ALTER TABLE ENTITE

ADD CONSTRAINT pk_ENTITE_NUMENTITE PRIMARY KEY (NUMENTITE);

"on modifie tous les noms d'attributs comportant CLIENT"

ALTER TABLE ENTITE

RENAME COLUMN ADRRUECLIENT TO ADRRUE;

ALTER TABLE ENTITE

RENAME COLUMN ADRCODEPOSTALCLIENT TO ADRCODEPOSTAL;

ALTER TABLE ENTITE

RENAME COLUMN ADRVILLECLIENT TO ADRVILLE;

ALTER TABLE ENTITE

RENAME COLUMN ADRPAYSCLIENT TO ADRPAYS;

ALTER TABLE ENTITE

RENAME COLUMN TELEPHONECLIENT TO TELEPHONE;

ALTER TABLE ENTITE

RENAME COLUMN MAILCLIENT TO MAIL;

-- créer la nouvelle table CLIENT :

```
CREATE TABLE CLIENT
(
  numEntite NUMBER(38),
  nomClient VARCHAR(50),
  adrRue VARCHAR2(50),
  adrCodePostal VARCHAR2(10),
  adrVille VARCHAR2(40),
  adrPays VARCHAR2(30),
  telephone CHAR(12),
  mail VARCHAR2(60),
  numContact1 VARCHAR2(50),
  telephoneContact1 CHAR(12),
  fonctionContact1 VARCHAR2(20),
  numContact2 VARCHAR2(50),
  telephoneContact2 CHAR(12),
  fonctionContact2 VARCHAR2(20),
  CONSTRAINT pk_CLIENT PRIMARY KEY (NUMENTITE),
  CONSTRAINT fk_CLIENT_NUMENTITE FOREIGN KEY (NUMENTITE)
  REFERENCES ENTITE (NUMENTITE)
);
```

-- créer la nouvelle table FOURNISSEUR :

```
CREATE TABLE FOURNISSEUR
(
  numEntite NUMBER(38),
  nomFournisseur VARCHAR(50),
  adrRue VARCHAR2(50),
  adrCodePostal VARCHAR2(10),
  adrVille VARCHAR2(40),
  adrPays VARCHAR2(30),
```

```
telephone CHAR(12),
mail VARCHAR2(60),
numContact1 VARCHAR2(50),
telephoneContact1 CHAR(12),
fonctionContact1 VARCHAR2(20),
numContact2 VARCHAR2(50),
telephoneContact2 CHAR(12),
fonctionContact2 VARCHAR2(20),
CONSTRAINT pk_FOURNISSEUR PRIMARY KEY (NUMENTITE),
CONSTRAINT fk_FOURNISSEUR_NUMENTITE FOREIGN KEY (NUMENTITE)
REFERENCES ENTITE (NUMENTITE)
);
```

-- remettre la liaison entre COMMANDE et CLIENT :

```
ALTER TABLE COMMANDE
ADD CONSTRAINT fk_COMMANDE_NUMCLIENT FOREIGN KEY (NUMCLIENT)
REFERENCES CLIENT (NUMENTITE);
```

-- ajout de la table PRODUCTION :

```
CREATE TABLE PRODUCTION
(
delaiLivraison DECIMAL(6),
prixAchat DECIMAL(3,1),
numArticle NUMBER(38),
numEntite NUMBER(38,0),
CONSTRAINT pk_PRODUCTION PRIMARY KEY (numArticle,numEntite),
CONSTRAINT fk_PRODUCTION_numArticle FOREIGN KEY (numArticle)
REFERENCES ARTICLE (numArticle),
CONSTRAINT fk_PRODUCTION_numEntite FOREIGN KEY (numEntite)
REFERENCES FOURNISSEUR (numEntite)
);
```

```
COMMIT ;
```