



# Rapport de BD

*Documentation de la partie BD de la SAE - Dev. APP semaine 50*

---

# Sommaire :

<b>Sommaire :</b>	<b>1</b>
<b>Introduction</b>	<b>2</b>
<b>Refonte de la base de données :</b>	<b>3</b>
Base de données précédente (v2) :	3
Base de données actuelle (v3) :	5
<b>Procédure "Paielement" :</b>	<b>7</b>

---

# Introduction

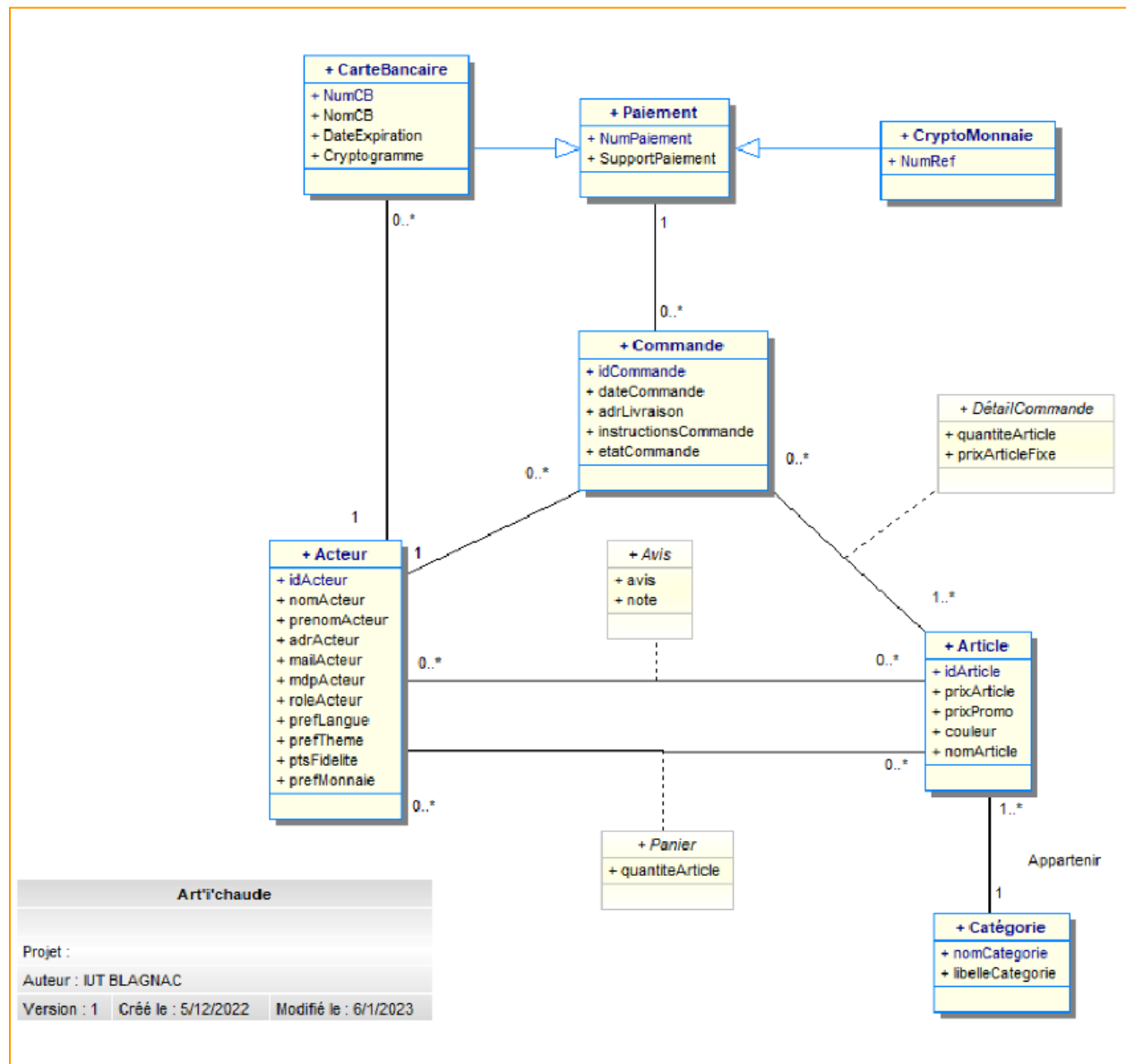
Ce document possède pour objectif, d'alimenter la partie destinée à la base de données de la SAÉ, accompagnée pour cela des modifications et des réalisations suivantes :

- Légère refonte de la base de données;
- Procédure "paiement".

Ce document est livré tardivement et pour cause, l'avancée du site web d'Art'i'Chaude a nécessité un temps plus conséquent que celui estimé. De ce fait, l'utilisation de la procédure étant erronée, il était inconcevable de fournir un livrable incomplet et d'informations incertaines.

# Refonte de la base de données :

Base de données précédente (v2) :



---

Après avoir optimisé la table 'ACTEUR' qui, auparavant, demandait un nombre conséquent de requêtes inutilement, nous étions fixés sur une base de données jugée 'conforme' à l'attente du client et à son rapport au site web.

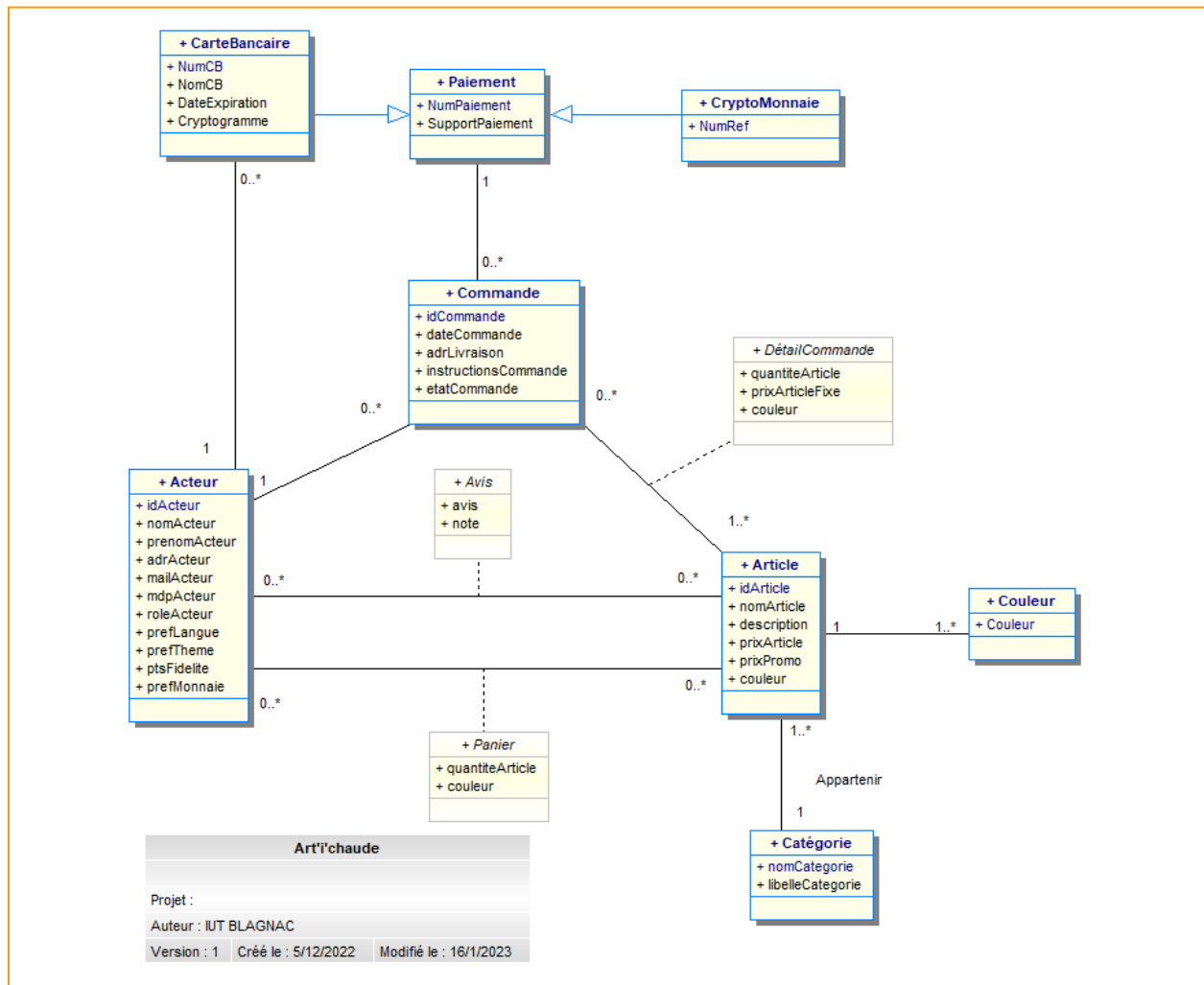
Pour autant, un problème subsistait : les variantes de couleur. Concrètement, un utilisateur a la possibilité de choisir des variantes de couleur pour un même article et un artiste peut créer un produit et définir sa propre couleur, sans oublier l'administrateur qui pourrait la modifier.

En apparence, le problème n'est pas flagrant puisque l'on peut qualifier qu'un produit possède une couleur. Mais si le produit possède plusieurs couleurs, comment la base de données et comment les données relatives à ces caractéristiques sont figurées ?

En réalité, la base de données jusque-là présente contraignait cette multitudes de possibilités, ne laissant alors qu'une possibilité de couleur pour un article.

Mais alors, comment le problème a été réglé ?

## Base de données actuelle (v3) :



Pour alors résoudre ce problème, nous avons décidé d'en réaliser une table à part, nommée 'COULEUR' et prenant pour caractéristiques :

- L'identifiant de l'article (IDARTICLE);
- Un attribut 'couleur' (COULEUR).

Cette distinction ne change rien à comparer de l'ancienne table 'ARTICLE' où un attribut 'COULEUR' était également figuré. Cependant, en observant la création de cette table, une différence notable est à déclarer :

```
CREATE TABLE COULEUR
(
  IDARTICLE NUMBER(5) NOT NULL,
  COULEUR VARCHAR(32) NULL,
  CONSTRAINT PK_COULEUR PRIMARY KEY (IDARTICLE, COULEUR)
) ;
```

La clé primaire de cette même table est composée des deux attributs. Cela signifie qu'entre-autres, pour un même article, celui-ci a la possibilité de posséder plusieurs couleurs tant qu'une combinaison 'IDARTICLE'-COULEUR n'existe pas dans la base de données. Ainsi, aucune contrainte lors d'un ajout, d'une modification, d'une sélection d'un produit avec une couleur spécifique sera envisageable.

Par ailleurs, ce même attribut sera retrouvé dans la table 'DETAILCOMMANDE' où il est nécessaire pour un client, de retrouver les caractéristiques de cette dernière, dont la couleur d'un article. Pour toujours éviter cette contrainte d'unicité, la clé primaire de cette table sera alors constituée des attributs :

- De l'identifiant de la commande (IDCOMMANDE);
- De l'identifiant de l'article (IDARTICLE);
- De la couleur associée à l'article (COULEUR).

```
CREATE TABLE DETAILCOMMANDE
(
  IDCOMMANDE NUMBER(10) NOT NULL,
  IDARTICLE NUMBER(5) NOT NULL,
  QUANTITEARTICLE NUMBER(3) NULL,
  PRIXARTICLEFIXE DECIMAL(8,2) NULL,
  COULEUR VARCHAR(32) NULL,
  CONSTRAINT PK_DETAILCOMMANDE PRIMARY KEY (IDCOMMANDE, IDARTICLE, COULEUR),
  CONSTRAINT CH_QUANTITEARTICLEDC CHECK(QUANTITEARTICLE >= 1 AND QUANTITEARTICLE <= 999),
  CONSTRAINT CH_PRIXARTICLEFIXE CHECK(PRIXARTICLEFIXE > 0 AND PRIXARTICLEFIXE < 1000000)
) ;
```

# Procédure “Paiement” :

En voici la procédure :

```
CREATE OR REPLACE PROCEDURE PASSERCOMMANDE (

    p_idAuteur ACTEUR.IDACTEUR%TYPE,

    p_adresseLivraison COMMANDE.ADRDLIVRAISON%TYPE,

    p_instructionsLivraison COMMANDE.INSTRUCTIONSCOMMANDE%TYPE,

    p_supportPaiement PAIEMENT.SUPPORTPAIEMENT%TYPE,

    p_numRef CRYPTOMONNAIE.NUMREF%TYPE,

    p_numCB CARTEBANCAIRE.NUMCB%TYPE,

    p_nomCB CARTEBANCAIRE.NOMCB%TYPE,

    p_dateCB CARTEBANCAIRE.DATEEXPIRATION%TYPE,

    p_cryptoCB CARTEBANCAIRE.CRYPTOGRAMME%TYPE,

    p_typeProcedure CHAR --ins del ras (insert, delete, rien a signaler)

)
```

**IS**

```
v_dateCommande COMMANDE.DATECOMMANDE%TYPE;

v_numPaiement PAIEMENT.NUMPAIEMENT%TYPE;

v_idCommande COMMANDE.IDCOMMANDE%TYPE;

v_prixArticleFixe DETAILCOMMANDE.PRIXARTICLEFIXE%TYPE;

v_prixTotalCommande INT;
```



```
CURSOR c_panier IS

    SELECT PANIER.IDACTEUR, PANIER.IDARTICLE, PANIER.QUANTITEARTICLE,
    PANIER.COULEUR, ARTICLE.PRIXARTICLE, ARTICLE.PRIXPROMO

    FROM PANIER, ARTICLE

    WHERE PANIER.IDARTICLE = ARTICLE.IDARTICLE

    AND PANIER.IDACTEUR = p_idAuteur;

BEGIN

v_dateCommande := CURRENT_DATE;

v_prixTotalCommande := 0;

INSERT INTO PAIEMENT (NUMPAIEMENT, SUPPORTPAIEMENT)

VALUES (SEQ_PAIEMENT.nextval, p_supportPaielement);

v_numPaielement := SEQ_PAIEMENT.currval;

INSERT INTO COMMANDE (IDCOMMANDE, NUMPAIEMENT, IDACTEUR, DATECOMMANDE,
ADRLIVRAISON, INSTRUCTIONSCOMMANDE, ETATCOMMANDE)

VALUES (SEQ_COMMANDE.nextval, v_numPaielement, p_idAuteur, v_dateCommande,
p_adresseLivraison, p_instructionsLivraison, 'T');

v_idCommande := SEQ_COMMANDE.currval;

-- curseur panier -> detailcommande

FOR ELEMENT IN c_panier LOOP

    v_prixArticleFixe := ELEMENT.PRIXARTICLE - (ELEMENT.PRIXARTICLE *
ELEMENT.PRIXPROMO / 100);

    v_prixTotalCommande := v_prixTotalCommande + (v_prixArticleFixe *
ELEMENT.QUANTITEARTICLE);

    INSERT INTO DETAILCOMMANDE (IDCOMMANDE, IDARTICLE, QUANTITEARTICLE,
PRIXARTICLEFIXE, COULEUR)
```

---

```
VALUES (v_idCommande, ELEMENT.IDARTICLE, ELEMENT.QUANTITEARTICLE,
v_prixArticleFixe, ELEMENT.COULEUR);

END LOOP;

DELETE FROM PANIER WHERE IDACTEUR = p_idAuteur;

UPDATE ACTEUR SET PTSFIDELITECLIENT = PTSFIDELITECLIENT + v_prixTotalCommande *
0.1 WHERE IDACTEUR = p_idAuteur;

IF (p_typeProcedure = 'ins') THEN

    INSERT INTO CARTEBANCAIRE (NUMCB, NUMPAIEMENT, IDACTEUR, NOMCB,
DATEEXPIRATION, CRYPTOGRAMME, SUPPORTPAIEMENT) VALUES (p_numCB, v_numPaie ment,
p_idAuteur, p_nomCB, p_dateCB, p_cryptoCB, p_supportPaie ment);

ELSIF (p_typeProcedure = 'del') THEN

    DELETE FROM CARTEBANCAIRE WHERE IDACTEUR = p_idAuteur;

ELSIF (p_supportPaie ment = 'B' OR p_supportPaie ment = 'E') THEN

    INSERT INTO CRYPTOMONNAIE (NUMREF, NUMPAIEMENT, SUPPORTPAIEMENT) VALUES
(p_numRef, v_numPaie ment, p_supportPaie ment);

END IF;

COMMIT;

DBMS_OUTPUT.PUT_LINE('Commande passée.');
```

END; /

---

Concrètement, cette procédure sera exécutée lorsque l'utilisateur aura soumis le formulaire de paiement disponible sur le site web. Celle-ci permet, d'une manière synthétisée, de :

- Créer la commande concernée;
- Basculer les informations du panier dans une table intitulée 'DETAILCOMMANDE' afin d'en conserver un récapitulatif de celui-ci;
- Vider le panier, une fois la commande réalisée;
- Ajouter le mode de paiement selon le choix de l'utilisateur.

Pour rentrer dans les détails, on définit un curseur (qui permet la sélection multi-lignes) pour l'ensemble des attributs du panier pour un utilisateur. Évidemment, cette procédure s'adapte à chaque utilisateur et donc, à chaque panier propre à ceux-ci.

La date de la commande correspond à la date actuelle, traduite par la fonction `CURRENT_DATE`.

Par ailleurs, dans la table 'PAIEMENT', correspondant à une table intermédiaire entre les tables 'CARTEBANCAIRE' et 'CRYPTOMONNAIE', on va insérer le numéro de paiement de la commande avec son support paiement (selon le choix de l'utilisateur) ainsi que créer la commande elle-même dans la table 'COMMANDE', contenant l'ensemble de ses caractéristiques.

Ensuite, pour chaque élément du panier, on va calculer le prix fixe de l'article (selon s'il est en promotion ou non) ainsi que le prix total, selon le nombre d'articles. Le premier de ces deux résultats est enregistré dans la table 'DETAILCOMMANDE' et une fois la commande passée, le panier sera mis à jour (autrement dit, vidé) et l'utilisateur bénéficiera de points de fidélité, correspondant à 10% du prix total de la commande.

Enfin, selon si l'utilisateur avait ou non les informations de sa carte bancaire dans la table 'CARTEBANCAIRE', ces mêmes informations seront retirées s'il ne décide pas à ce qu'on se souvienne de ses informations, ou seront enregistrées s'il ne possède pas ses informations sur la base de données mais surtout s'il le souhaite. Ou bien, si son mode de paiement était par crypto monnaie, on viendra stocker les caractéristiques propres à un paiement par crypto monnaie (à noter que deux numéros de référence ne peuvent être identiques).