

Yann GUILLEVIC – Enzo MANCINI

Compte Rendue SAE 1.04

Sommaire

Présentation de la SAE.....	3
Étape 1 :	4
Schéma Conceptuel de la BD :	4
Explication des multiplicités :	4
Dictionnaire des données :	7
Étape 2 :	8
Schéma logique :	8
Modélisation relationnelle :	9
Étape 3 :	11
Script SQL-LDD :	11
Étape 4 :	11
Script SQL-LMD :	11
Étape 5 :	11
Requêtes SQL-LID :	11

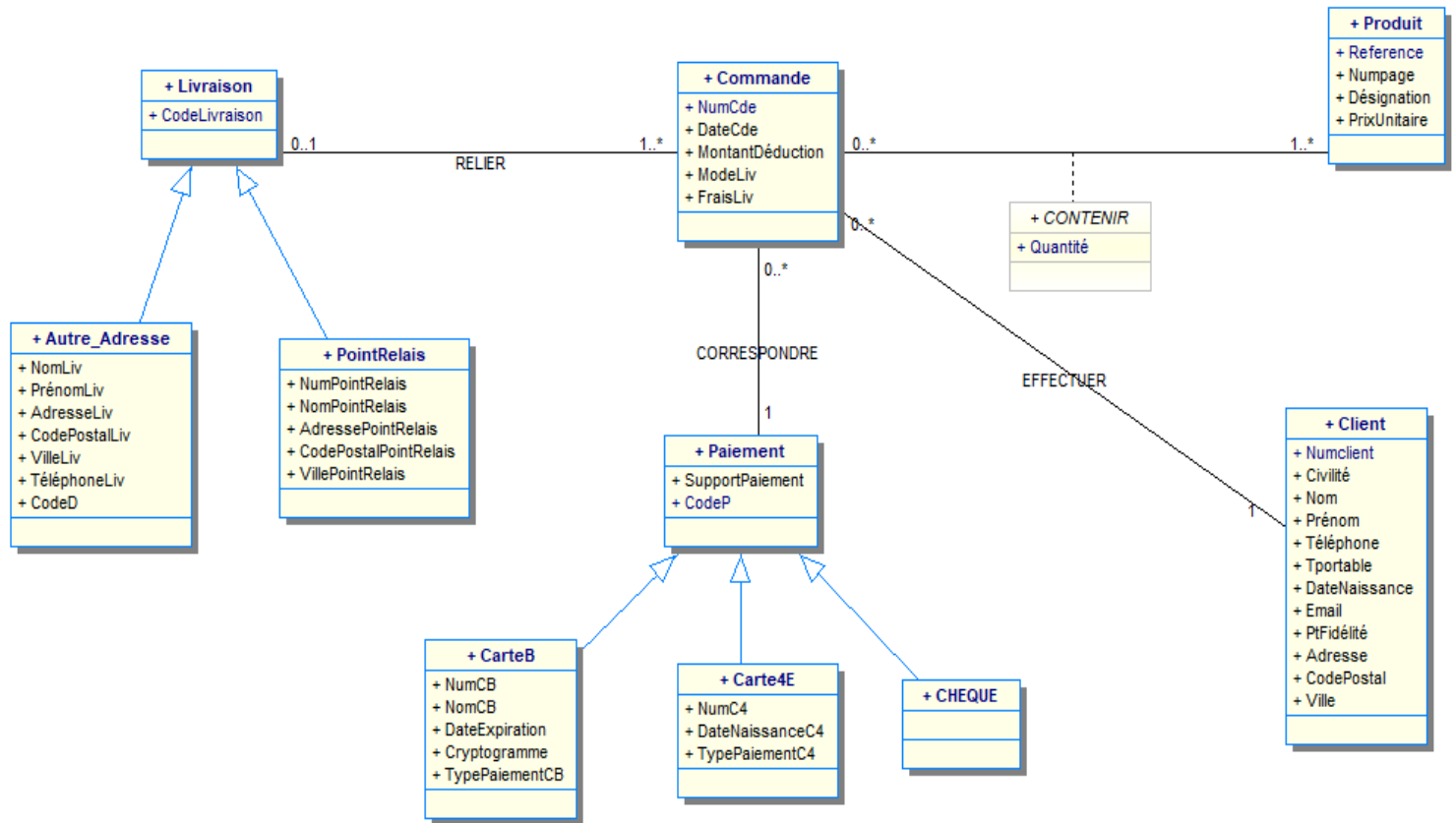
Présentation de la SAE

Dans cette SAE, nous devons créer une base de donnée permettant d'y recenser des informations liées à un bon de commande de l'entreprise Becquet. Ce projet nous permettra de nous entraîner sur la création d'un schéma logique compatible au recensement des données, de nous entraîner sur la modélisation relationnelle ainsi que sur les différents langage de programmation en SQL comme le SQL-LMD, SQL-LDD ou SQL-LID.

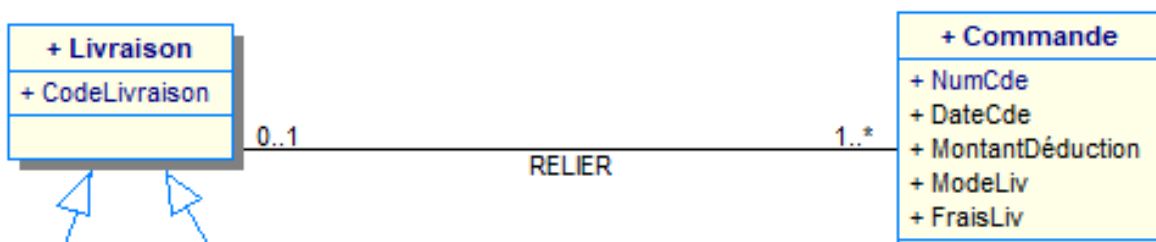
Ce projet sera fait en binôme.

Étape 1 :

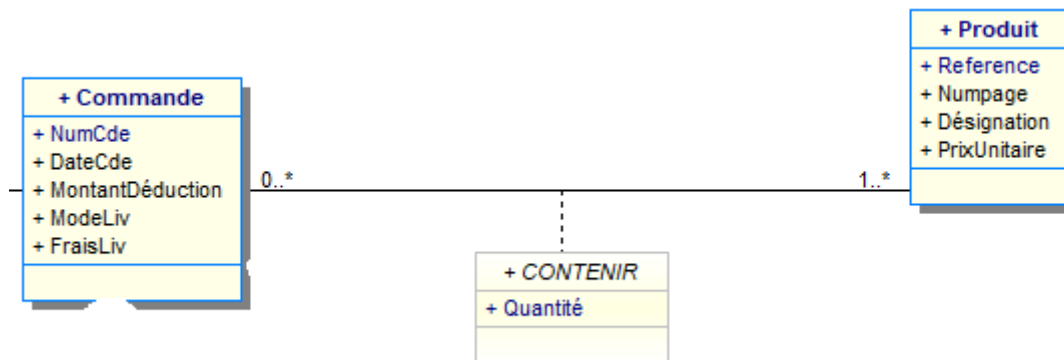
Schéma Conceptuel de la BD :



Explication des multiplicités :

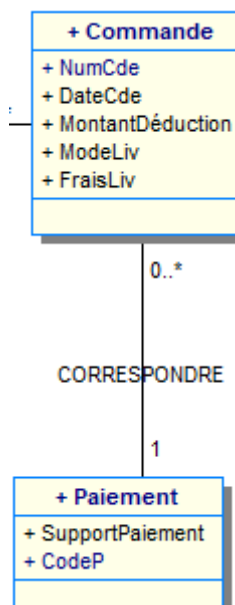


Ici, nous avons choisi de mettre une multiplicité 0..1 du côté LIVRAISON car une commande est reliée à un maximum de une livraison. De l'autre côté, une livraison peut être reliée, au minimum, à une commande.

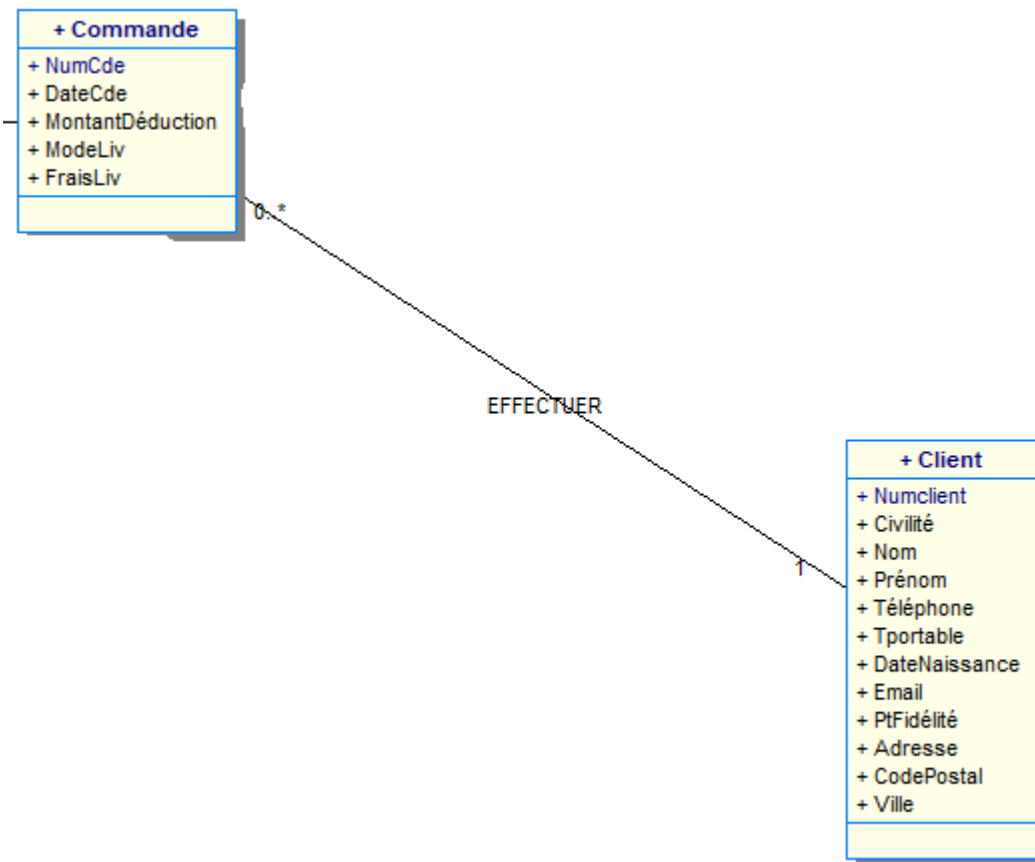


Ici, nous avons choisi de mettre une multiplicité 1..* du côté PRODUIT car une commande au minimum un produit. De l'autre côté, la multiplicité 0..* permet de dire qu'un produit peut être appliqué à un nombre quelconque de commande.

La classe d'association CONTENIR avec l'attribut « Quantité » est nécessaire car cet attribut ne peut pas se situer dans une des deux tables, par soucis de conflit de commande entre plusieurs clients.



Ici, nous avons choisi de mettre une multiplicité 1..1 du côté PAIEMENT car une commande correspond à un et un seul paiement (il n'y a qu'un montant final). De l'autre côté, la multiplicité 0..* permet de dire qu'un paiement peut correspondre à un nombre quelconque de commande.



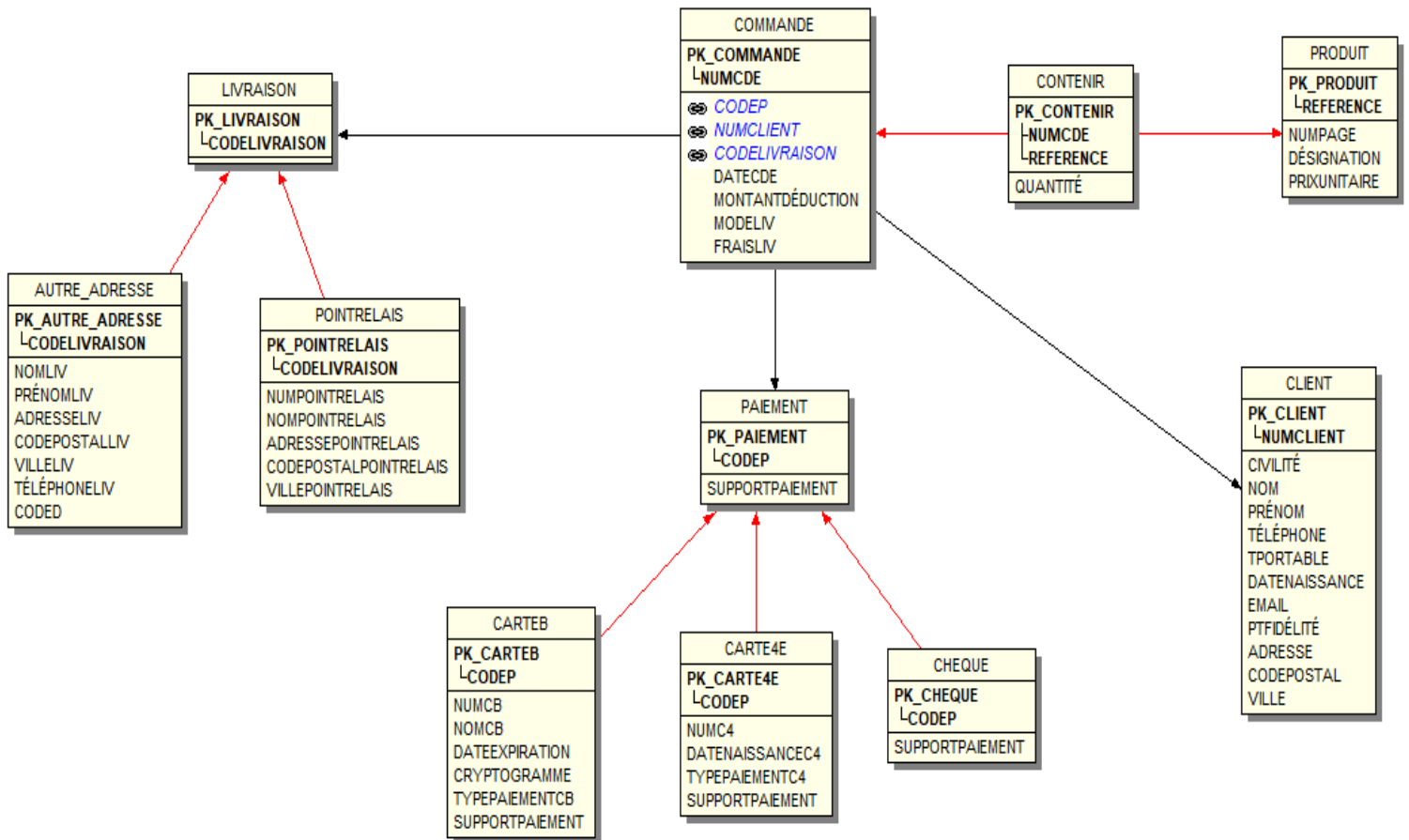
Ici, nous avons choisi de mettre une multiplicité 1..1 du côté CLIENT car une commande peut être effectuée par un et un seul client. De l'autre côté, la multiplicité 0..* veut dire qu'un client peut effectuer un nombre quelconque de commandes.

Dictionnaire des donnés :

Nom	description	Type	Contrainte	Calcul
N°Cmd	numéro (unique) du Bon de Commande	Entier	>0	/
Date_Cmd	Date de saisie du Bon de Commande	Date	JJ/MM/AAAA	/
N°client	numero(unique) du client	chaîne de 12 caractères	/	/
Civilité	Civilité du client	chaîne de 4 caractères	∈{MR;MME;MLLE}	/
nom	nom du client	chaîne de 30 caractères	/	/
prenom	prenom du client	chaîne de 30 caractères	/	/
adresse	libellé de l'adresse postale du client	chaîne de 30 caractères	/	/
CodePostal	code postal du client	chaîne fixe de 5 caractères	/	/
ville	ville du client	chaîne de 30 caractères	/	/
téléphone	Numéro de téléphone fixe du client	chaîne fixe de 10 caractères	/	/
téléphone_por	Numéro de téléphone portable du client	chaîne fixe de 10 caractères	/	/
date_nais	Date naissance du client	Date	JJ/MM/AAAA	/
mail	mail du client	chaîne de 30 caractères	/	/
PtFidélité	Capital des points de fidélités du client	Entier	>0	/
N°Page	Numéro de page du catalogue où apparaît l'article	Chaîne de 3 caractères	/	/
Désignation	désignation de l'article	Chaîne de 30 caractères	/	/
Référence	Référence (unique) de l'article	Chaîne fixe de 6 caractères	/	/
Quantité	Quantité commandée de l'article	Entier	>0	/
PrixUnitaire	Prix unitaire de l'article	Réel	>0	/
Montant	Montant sous-total avant réduction	Réel	>0	Σ Quantité x PrixUnitaire
MontantDédution	Montant du chèque déduction	Entier	∈{15;30;50;65;100}	/
TotalAprèsDédution	Montant sous-total après déduction	Réel	>0	Montant – MontantDédution
MontantCmd	Montant de la commande	Réel	>0	TotalAprèsRéduction
FraisTraitement	Frais de participation au traitement de Bon de commande	Réel	=6.99	6.99€
Modeliv	Mode de livraison	Chaîne de 12 caractères	∈{PointRelais;Domicile;Express}	/
FraisLiv	Frais de livraison	Réel	∈{0.0;2.0;9.9}	/
TotalCode	Montant total du bon de livraison	Réel	>0	MontantCmd + FraisTraitement + FraisLiv
NomLiv	Nom de la livraison	Chaîne de 30 caractères	/	/
PrénomLiv	Prénom de la livraison	Chaîne de 30 caractères	/	/
AdresseLiv	Libellé de l'adresse postale de la livraison (si différente de celle du client)	Chaîne de 30 caractères	/	/
CodePostalLiv	Code postal de la livraison (si différente de celle du client)	Chaîne fixe de 5 caractères	/	/
VilleLiv	Ville de livraison (si différente de celle du client)	Chaîne de 30 caractères	/	/
TéléphoneLiv	Téléphone à la livraison	Chaîne fixe de 10 caractères	/	/
N°PointRelais	Numéro (unique) du point relais	Chaîne fixe de 5 caractères	/	/
NomPointRelais	Nom du point relais	Chaîne de 30 caractères	/	/
AdressePointRelais	Libellé de l'adresse postale du point relais	Chaîne de 30 caractères	/	/
CodePostalPointRelais	Code postale du point relais	Chaîne fixe de 5 caractères	/	/
VillePointRelais	Ville du point relais	Chaîne de 30 caractères	/	/
SupportPaiement	Support de paiement choisi pour le Bon de Commande	Chaîne fixe de 6 caractères	∈{Chèque;Carte4;Carte8}	/
N°C4	Numéro (unique) de Carte 4 Etoiles	Chaîne fixe de 9 caractères	/	/
DateNaissanceC4	Date de naissance du porteur de Carte 4 Etoiles	Date	JJ/MM/AAAA	/
TypePaiementC4	Type de paiement de Carte 4 Etoiles choisi pour le Bon de Commande	Chaîne fixe de 1 caractères	∈{C;M} C : Comptant ; M : Mensuel	/
N°CB	Numéro (unique) de Carte Bancaire	Chaîne fixe de 16 caractères	/	/
NomCB	Nom du porteur de la Carte Bancaire	Chaîne de 30 caractères	/	/
DateExpiration	Date d'expiration de la Carte Bancaire	Chaîne fixe de 4 caractères	MM/AA	/
Cryptogramme	Cryptogramme (code à 3 chiffre) de la Carte Bancaire	Chaîne fixe de 3 caractères	/	/
TypePaiementCB	Type de paiement de Carte Bancaire choisi pour le Bon de Commande	Chaîne fixe de 1 caractères	∈{C;M} C : Comptant ; M : Mensuel	/

Étape 2 :

Schéma logique :



Modélisation relationnelle :

code couleur :

Vert = explication de la règle.

Rouge = clé primaire d'une table.

Bleu = clé étrangère d'une table.

R1 : classe d'objets :

CLIENT(NumClient, civilité, nom, prénom, Téléphone, Tportable,
DateNaissance, Email, PtFidelite, Adresse, CodePostal, Ville)
PRODUIT(Reference, Numpage, Désignation, PrixUnitaire)
COMMANDE(NumCde, DateCde, MontantDédution, ModeLiv, FraisLiv)

Chaque classe d'objets du schéma devient une classe de même nom avec les mêmes attributs. La clé primaire est soit choisi parmi les attributs de la table soit créée si aucun attribut ne peut jouer ce rôle.

R2 : classe d'associations (1,n) :

LIVRAISON(CodeLivraison)

PAIEMENT(CodeP, SupportPaiement)

CLIENT(NumClient, civilité, nom, prénom, Téléphone, Tportable,
DateNaissance, Email, PtFidelite, Adresse, CodePostal, Ville)

COMMANDE(NumCde, CodeLivraison#, CodeP#, NumClient#, DateCde,
MontantDédution, ModeLiv, FraisLiv)

Les attributs formant la clé primaire du coté 1..1 sont dupliqués dans la classe opposé et deviennent des clés étrangères dans ces tables.

R3 : classe d'associations (n,m) :

COMMANDE(NumCde, DateCde, MontantDédution, ModeLiv, FraisLiv)
PRODUIT(Reference, Numpage, Désignation, PrixUnitaire)
CONTENIR(Reference#, NumCde#, Quantité)

Les attributs formant des clés primaires dans les classes d'objets reliés à la classe d'association, deviennent des clé primaires et étrangères dans la classe d'association.

R4 : classe d'association (1,1) :

Nous ne pouvons pas appliquer cette règle car nous n'avons aucune multiplicité égale à 0..1 ou d'un coté 1..1 et de l'autre 0..1.

R5 : Héritage :

Lien avec LIVRAISON :

LIVRAISON(CodeLivraison)

Autre_Adresse(CodeLivraison, NomLiv, PrénomLiv, CodePostalLiv, VilleLiv, TéléphoneLiv)

Point_Relais(CodeLivraison, NumPointRelais, NomPointRelais, AdressePointRelais, CodePostalPointRelais, VillePointRelais)

Lien avec PAIEMENT :

PAIEMENT(CodeP, SupportPaiement)

CARTEB(CodeP, NumCB, NomCB, DateExpiration, Cryptogramme, TypePaiementCB, SupportPaiement)

CARTE4E(CodeP, NumC4, DateNaissanceC4, TypePaiementC4, SupportPaiement)

CHEQUE(CodeP, SupportPaiement)

L'héritage permet de créer, à partir d'une table, des « sous-tables » spécifiques relié à la table mère.

Pour les deux héritages cités ci-dessus, nous avons choisi de faire un héritage avec partition. L'attribut formant la clé primaire dans la table mère devient clé primaire dans les tables filles et les autres attributs présent dans la table mère sont également transférés dans les tables filles.

Avec ce choix, il ne doit y avoir aucune clé primaire déjà présente dans les tables filles. Cependant, il est précisé, dans le dictionnaire des données, que certains attributs soit tout de même unique (NumCB, NumC4, NumPointRelais). Donc, il nous sera utile de créer des contraintes d'unicités dans le script sur Oracle pour respecter ces contraintes.

Étape 3 :

Script SQL-LDD :

Le script sera fournis sous fichier .pdf avec le compte rendue. Par soucis de compréhension du code par SQL Developer, nous avons remplacé tous les « N° » par « Num ». La ligne Create DataBase MLR1 a été mis en commentaire dans le script car cela générerait une erreur étant donné que nous avons déjà une database.

On rajoute à la fin :

CONSTRAINT UN_POINTRELAIS UNIQUE (NUMPOINTRELAIS)

CONSTRAINT UN_CARTE4E UNIQUE (NUMC4)

CONSTRAINT UN_CARTEB UNIQUE (NUMCB)

Étape 4 :

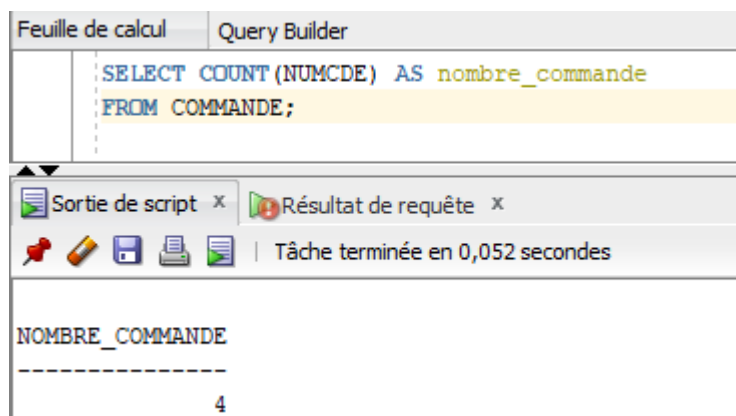
Script SQL-LMD :

Le script sera fournis sous fichier .pdf avec le compte rendue.

Étape 5 :

Requêtes SQL-LID :

1) Nombre de commande passées



2) Montant total et nombre de commande

Feuille de calcul		Query Builder	
		<pre>SELECT C.NumCde, SUM(P.prixunitaire*CO.quantité) - SUM(DISTINCT (C.montantdeduction))+ SUM(DISTINCT(C.fraisLiv)) +6.99 AS montant_Total FROM PRODUIT P, CONTENIR CO, COMMANDE C WHERE P.reference = CO.reference AND CO.NUMCDE = C.NUMCDE GROUP BY C.NumCde, C.montantdeduction,C.fraisliv ORDER BY C.numCde ASC;</pre>	
Sortie de script		Résultat de requête	
		Toutes les lignes extraites : 4 en 0,041 secondes	
NUMCDE	MONTANT_TOTAL		
1	52,84		
2	83,17		
3	156,69		
4	296,69		

3) Nombre de vente par produit

Feuille de calcul

Query Builder

```
SELECT COUNT(quantité) as nb_vente, P.REFERENCE
FROM PRODUIT P, CONTENIR C
WHERE C.REFERENCE = P.REFERENCE
GROUP BY P.REFERENCE;
```

Sortie de script x

Résultat de requête x

Tâche terminée en 0,022 secondes

NB_VENTE	REFERENCE
1	471147
1	471159
1	471162
1	905784
1	905968
1	950614
2	950728

7 lignes sélectionnées.

4) Nombre de commandes passées par client

Feuille de calcul		Query Builder
		<pre>SELECT COUNT(NUMCDE) as nb_commande, CE.NUMCLIENT FROM COMMANDE C, CLIENT CE WHERE C.NUMCLIENT = CE.NUMCLIENT GROUP BY CE.NUMCLIENT;</pre>
		<div>Sortie de script x</div> <div>Tâche terminée en 0,09 secondes</div>
		NB_COMMANDE NUMCLIENT

		1 000100
		2 000200
		1 000300

5) Nombre de commandes passées et montant total par genre(H et F) des clients.


Feuille de calcul

Query Builder

```
SELECT CE.civilité, COUNT(DISTINCT(C.NUMCDE)) as nombre_commande, SUM(P.prixunitaire*CO.quantité) - SUM(DISTINCT (C.montantdeduction))+ SUM(DISTINCT(C.fraisLiv)) +6.99 AS montant_Total
FROM PRODUIT P, CONTENIR CO, COMMANDE C, CLIENT CE
WHERE P.reference = CO.reference
AND CO.NUMCDE = C.NUMCDE
AND CE.NUMCLIENT = C.NUMCLIENT
GROUP BY CE.civilité;
```

Sortie de script x

Résultat de requête x

 | Tâche terminée en 0,23 secondes

CIVILITÉ	NOMBRE_COMMANDE	MONTANT_TOTAL
MME	2	202,54
MR	2	372,87

6) Obtenir le prix unitaire du produit n°950728

Feuille de calcul		Query Builder
		<pre>SELECT PRIXUNITAIRE FROM PRODUIT WHERE REFERENCE = 950728;</pre>
		Sortie de script x
		Tâche terminée en 0,027 sec
		PRIXUNITAIRE

		129,9

7) Obtenir les clients ayant commandé le produit n°950728 avec la quantité commandée

Feuille de calcul		Query Builder
		<pre>SELECT C.NUMCLIENT, CE.quantité FROM CLIENT C, CONTENIR CE, COMMANDE CO WHERE CE.REFERENCE = 950728 AND CO.NUMCLIENT = C.NUMCLIENT AND CO.NUMCDE = CE.NUMCDE;</pre>
		Sortie de script x
		Tâche terminée en 0,037 secondes
		NUMCLIENT

		000300
		000200
		QUANTITÉ

		1
		3