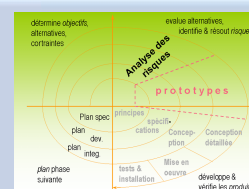
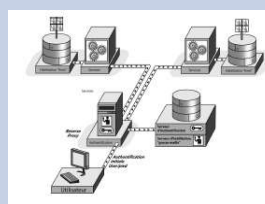


Secteur Tertiaire Informatique
Filière « Etude et développement »

Séquence « Développer des composants dans le langage de la base de données »

Cas Papyrus SQL Server :
Programmer des Triggers

Apprentissage



Mise en pratique



Evaluation

TABLE DES MATIERES

Table des matières	3
1. PRESENTATION	5
1.1 Le cahier des charges.....	5
1.2 Le modèle conceptuel des données.....	6
2. CREATION DE LA BASE DE DONNEES	7
2.1 Le dictionnaire des données	7
2.2 Le modèle physique	8
3. LE TRAVAIL A EFFECTUER	9
3.1 Implémentation de la base	9
3.2 Démarche proposée.....	9
3.3 Requêtes SQL complexes à réaliser	10
3.4 Triggers à programmer.....	12
3.4.1 Création d'un déclencheur AFTER DELETE.....	12
3.4.2 Création d'un déclencheur AFTER UPDATE	12
3.4.3 Création d'un trigger de contrôle d'évolutivité	12

Préambule

Ce document présente le cas Papyrus et propose une série de triggers (ou 'déclencheurs') à programmer sur la base de données.

Objectifs

Méthodologie

1. PRESENTATION

1.1 LE CAHIER DES CHARGES

Le souci majeur de M. PURCHASE, chef de la production informatique de la société BIDOUILLE EXPRESS, est d'assurer la gestion et le suivi des produits consommables tels que :

- papier listing en continu sous toutes ses formes,
- papier pré imprimé (commandes, factures, bulletins paie, ...)
- rubans pour imprimantes
- bandes magnétiques,
- disquettes,
-

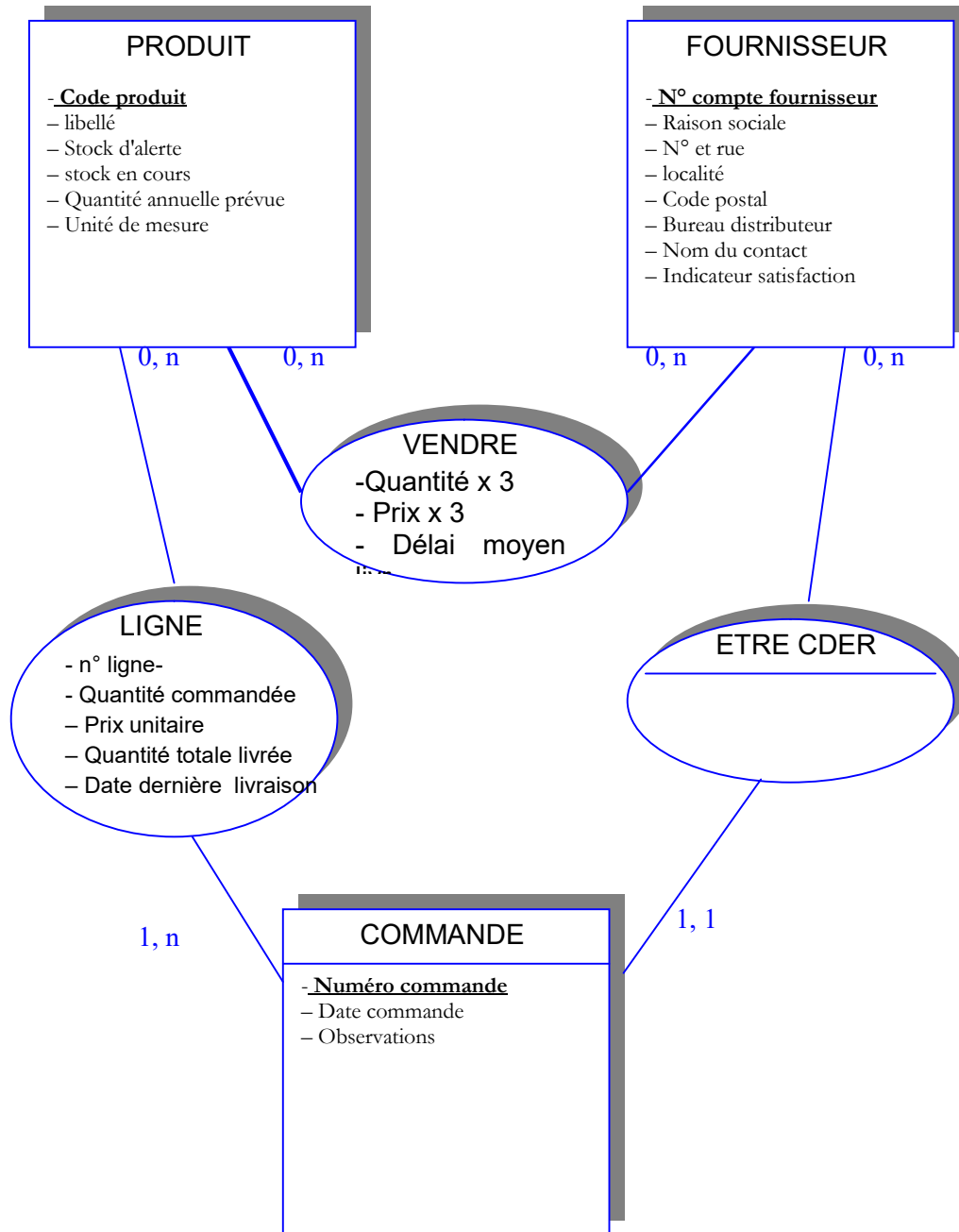
Pour chacun de ces produits, il existe plusieurs fournisseurs possibles ayant déjà livré la société ou avec lesquels M. PURCHASE est en contact. De plus, de nombreux représentants passent régulièrement vanter leurs produits et leurs conditions de vente : ceci permet à M. PURCHASE de conserver leurs coordonnées pour d'éventuelles futures commandes ou futurs appels d'offres.

Un appel d'offre se matérialise par un envoi de courrier précisant la nature de la demande (type de consommable, quantité prévisible de la commande, quantité annuelle, délai de livraison courant, délai de livraison en cas de rupture de stock, ...) aux fournisseurs susceptibles de faire une offre valable.

En retour, les fournisseurs intéressés par le marché renvoient leurs conditions de vente et M. PURCHASE peut faire son choix.

La commande est envoyée au fournisseur pour l'achat de un ou plusieurs produits pour une quantité donnée. Cette quantité peut être livrée en plusieurs fois. Les seules informations mémorisées sur la livraison sont la date de dernière livraison ainsi que la quantité livrée totale.

1.2 LE MODELE CONCEPTUEL DES DONNEES



2. CREATION DE LA BASE DE DONNEES

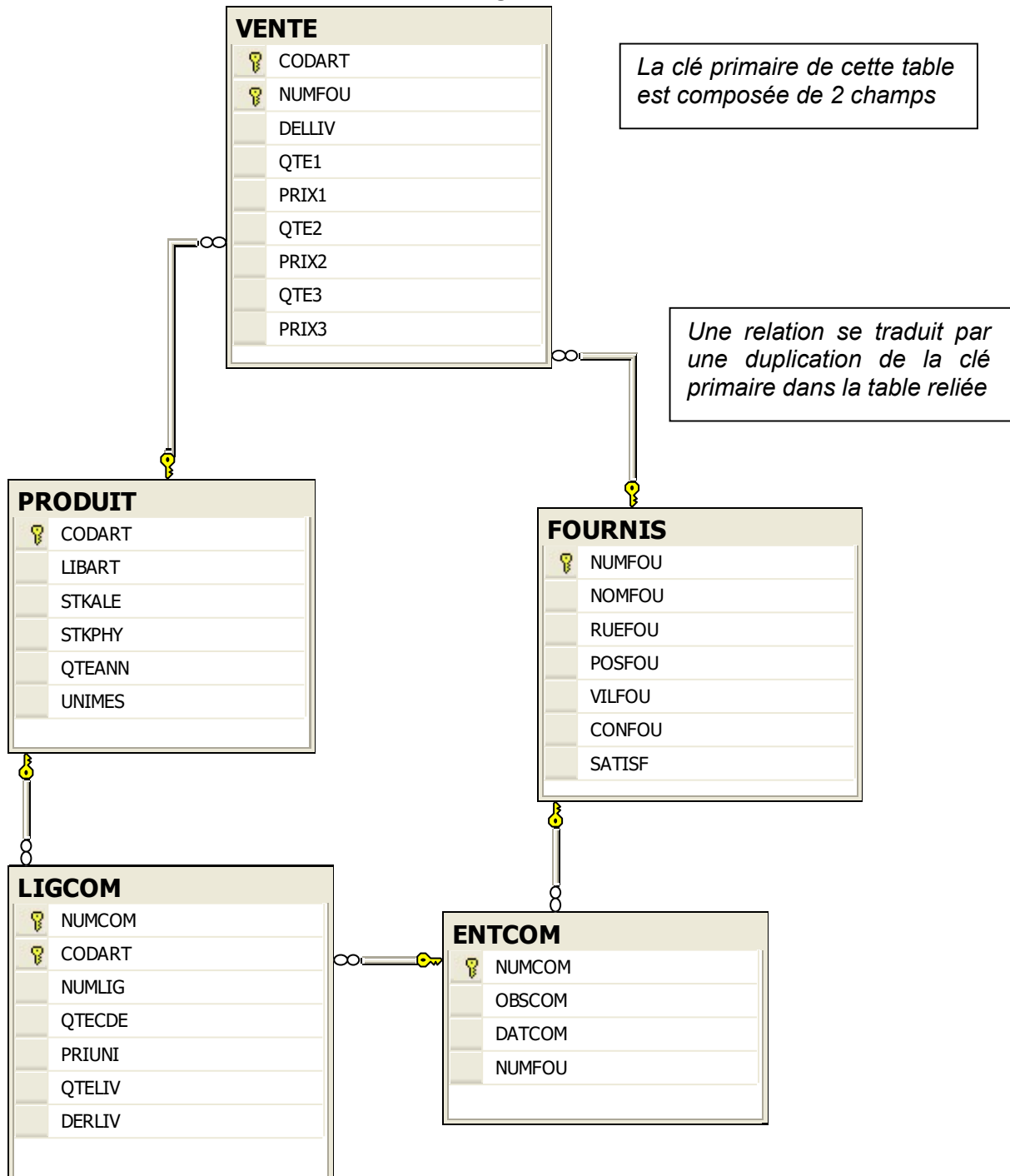
2.1 LE DICTIONNAIRE DES DONNEES

Les champs utilisés dans les différentes tables sont listés dans le tableau ci-dessous

CODART	Code produit	char(4)
CONFOU	Contact chez le fournisseur	varchar(15)
DATCOM	Date de commande	smalldatetime
DELLIV	Délai de livraison	smallint
DERLIV	Date dernière livraison	smalldatetime
LIBART	Libellé Produit	varchar(35)
NUMCOM	Numéro de commande	int
NUMFOU	N° de compte fournisseur	int
NUMLIG	N° de ligne commande	tinyint
NOMFOU	Nom fournisseur	varchar(35)
OBSCOM	Observations	varchar(35)
POSFOU	Code postal fournisseur	char(5)
PRIUNI	Prix unitaire de vente	money
PRIX1	Prix unitaire 1	money
PRIX2	Prix unitaire 2	money
PRIX3	Prix unitaire 3	money
QTE1	Borne quantité livraison 1	smallint
QTE2	Borne quantité livraison 2	smallint
QTE3	Borne quantité livraison 3	smallint
QTEANN	Quantité annuelle	smallint
QTECDE	Quantité commandée	smallint
QTELIV	Quantité livrée	smallint
RUEFOU	Adresse fournisseur	varchar(35)
SATISF	Indice satisfaction	tinyint
STKALE	Stock d'alerte	smallint
STKPHY	Stock physique	smallint
UNIMES	Unité de mesure	char(5)
VILFOU	Ville fournisseur	varchar(30)

2.2 LE MODELE PHYSIQUE

La base de données relationnelle PAPYRUS est constituée des relations suivantes :



PRODUIT (CODART, LIBART, STKLE, STKPHY, QTEANN, UNIMES)

ENTCOM (NUMCOM, OBSCOM, DATCOM, NUMFOU)

LIGCOM (NUMCOM, CODART, NUMLIG, QTECDE, PRIUNI, QTELIV, DERLIV)

FOURNIS (NUMFOU, NOMFOU, RUEFOU, POSFOU, VILFOU, CONFOU, SATISF)

VENTE (CODART, NUMFOU, DELLIV, QTE1, PRIX1, QTE2, PRIX2, QTE3, PRIX3)

3. LE TRAVAIL A EFFECTUER

3.1 IMPLEMENTATION DE LA BASE

Si ce n'est déjà fait lors d'une étude précédente, implémentez la base de données Papyrus en modélisant le MCD pour en générer au final le script LDD SQL, ou en manipulant l'outil interactif d'administration du SGBD ou encore en exécutant le script fourni correspondant à votre SGBD. Pour cette étude, il n'est pas nécessaire de se focaliser sur les contraintes autres que celles de clés primaires et clés étrangères ni sur les index ; il sera nécessaire de saisir un petit jeu d'essai significatif.

3.2 DEMARCHE PROPOSEE

Quelques conseils pour l'écriture d'une commande SELECT :

- ◆ Décider quels sont les attributs à visualiser, les inclure dans la clause SELECT.
- ◆ Les expressions présentes dans la liste de sélection d'une requête (clause SELECT) avec la clause GROUP BY doivent être des fonctions d'agrégation ou apparaître dans la liste GROUP BY.
- ◆ Déterminer les tables à mettre en jeu, les inclure dans la clause FROM.
- ◆ Déterminer les conditions de jointure quand plusieurs tables sont en jeu.
- ◆ Déterminer les conditions limitant la recherche : les conditions portant sur les groupes doivent figurer dans une clause HAVING, celles portant sur des valeurs individuelles dans une clause WHERE.
- ◆ Préciser l'ordre d'apparition des lignes de résultat dans une clause ORDER BY.

On peut formaliser la démarche en remplissant le tableau suivant pour aider à construire chaque requête :

<i>Question ?</i>	<i>Rép</i>	<i>Requête</i>
Quelles sont les colonnes à afficher en résultat (select) ?		SELECT...
De quelles tables sont issues ces colonnes (from) ?		FROM
Y-a-t-il jointures ?		
N'y-a-t-il que certaines lignes à prendre en compte (where) ?		WHERE
Veut-on un résultat par « paquets » de lignes (group by) ?		GROUP BY
Veut-on voir apparaître les résultats selon un ordre précis (order by) ?		ORDER BY
N'y-a-t-il que certaines lignes résultats à prendre en compte (having) ?		HAVING

3.3 REQUETES SQL COMPLEXES A REALISER

1. Quelles sont les commandes dont le numéro fournisseur est 9120 ?
2. Combien de commandes (nommées en « nombre de commandes ») a fait chaque fournisseur ?
3. Combien y a-t-il eu de commandes (nommées en « nombre de commandes ») et combien de fournisseurs (nommés en « nombre de fournisseur ») ?
4. Quels sont les articles dont la quantité en stock est inférieure au stock alerte et dont la quantité annuelle est inférieure à 1000 ? On souhaite avoir le code article et son nom ainsi que ses stocks.
5. Quels sont les fournisseurs situés dans les départements 75 78 92 77 ?
L'affichage (département (et uniquement département), nom fournisseur) sera effectué par département décroissant, puis par ordre alphabétique.
6. Quelles sont les commandes passées et les dates de commandes aux mois de mars et avril ?
7. Quelles sont les commandes et leurs dates ayant une observation lors de l'année 2007 ?
8. Donnez les commandes ainsi que leur « Total » de commande par ordre décroissant de chiffre d'affaire.
9. Lister les commandes dont le total est supérieur à 10 000€ ; on exclura dans le calcul du total les articles commandés en quantité supérieure ou égale à 1000.
(Affichage numéro de commande et total)
10. Lister les numéros de commande avec leur date et leur nom de fournisseur.
11. Lister les articles et leur prix de vente (« sous total ») provenant d'une commande urgente.
On affichera également le numéro de la commande et le nom du fournisseur.
12. Lister le nom des fournisseurs susceptibles de livrer au moins un article.
13. Lister les commandes et leurs dates du fournisseur ayant passé la commande 5

14. Dans les articles susceptibles d'être vendus, lister les articles moins chers (basés sur Prix1) que le moins cher des rubans (article dont le premier caractère du code article commence par 'R'). On affichera le libellé de l'article et prix1.
15. Editer la liste des fournisseurs susceptibles de livrer les produits dont le stock est inférieur ou égal à 150 % du stock d'alerte. La liste est triée par produit puis fournisseur.
16. Éditer la liste des fournisseurs susceptibles de livrer les produits dont le stock est inférieur ou égal à 150 % du stock d'alerte et un délai de livraison d'au plus 30 jours. La liste est triée par fournisseur puis produit.
17. Avec le même type de sélection que ci-dessus, sortir un total des stocks par fournisseur trié par total décroissant.
18. En fin d'année, sortir la liste des produits dont la quantité annuelle prévue est inférieure de 10 % à la quantité réellement commandée.
19. Existe-t-il des lignes de commande non cohérentes avec les produits vendus par les fournisseurs ?
20. Lister les chiffres d'affaires par fournisseur en ordre décroissant
(CA Fournisseur = somme des quantités d'articles en commande * prix unitaires des articles présents dans les tables des commandes passées aux fournisseurs).
21. Mettre à jour le champ obscom en positionnant '*****' pour toutes les commandes dont le fournisseur a un indice de satisfaction <5
22. Augmenter les prix des articles du fournisseur 9180 de 4% sur le PRIX1 et de 2% sur le PRIX2
23. Tous les PRIX2 non renseignés prennent le PRIX1.
24. Supprimer l'article '1110'

3.4 TRIGGERS A PROGRAMMER

3.4.1 Création d'un déclencheur AFTER DELETE

Créer un déclencheur AFTER DELETE sur la table VENTE, pour empêcher l'utilisateur de supprimer plusieurs lignes de la table VENTE à la fois.

3.4.2 Création d'un déclencheur AFTER UPDATE

a)Créer une table ARTICLES_A_COMMANDER de structure :

ARTICLES_A_COMMANDER (CODART, QTE, DATE)

Dans cette table seront stockés les articles à commander.

Cette table temporaire n'a pas de clé primaire.

b) Créer un déclencheur UPDATE sur la table PRODUIT.

Fonctions de ce déclencheur :

- 1) Lorsque le stock physique est égal à 0, toute décrémentation est rejetée : la transaction est annulée et un message est envoyé dans les journaux.
- 2) Lorsque le stock physique augmenté de la somme des quantités de l'article dans la table ARTICLES_A_COMMANDER devient inférieur ou égal au stock d'alerte, une nouvelle ligne est insérée dans la table ARTICLES_A_COMMANDER avec les champs code article, qte et date du jour

NB : $qte = stock\ alerte - (stock\ physique + somme\ des\ quantités\ commandées\ pour\ cet\ article\ dans\ la\ table\ ARTICLES_A_COMMANDER)$

3.4.3 Création d'un trigger de contrôle d'évolutivité

L'indice de satisfaction d'un fournisseur évolue, mais lentement, dans le temps. On considère qu'un saut de valeur de plus de 3 points est une erreur de saisie.

Ecrire un trigger qui refuse une mise à jour de l'indice de satisfaction d'un fournisseur quand il évolue de plus de trois points, en positif comme en négatif.

CREDITS

ŒUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

E. Catttanéo – Formatrice

B. Hézard - Formateur

Ch. Perrachon – Ingénieure de formation

Date de mise à jour : 08/02/16

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »

Cas Papyrus SQL Server – Programmer des Triggers

Afpa © 2016 – Section Tertiaire Informatique – Filière « Etude et développement »