Base de données - Achats

*Théo Lorio / Gabriel Bailleul / Louis Iwinski*

Une image contenant texte, intérieur, scène, marché

Description générée automatiquement

06/10/2021

Base de données - Achats

Théo Lorio / Gabriel Bailleul / Louis Iwinski

Sommaire

Étape 1 pages 2-5

* Diagramme E/R de la base de données
* Description sommaire de la base de données
* Tests des contraintes et affichages des erreurs

Étape 2 page 6

* Diagramme E/R de la base de données sans contraintes d’intégrités

Étape 3 pages 7-8

* Un trigger et une fonction PL/SQL portant sur une table contenant une clé étrangère

Étape 1

Diagramme E/R

Une image contenant texte

Description générée automatiquement

Diagramme E/R de notre base de données Achats réalisé sur QuickDBD

Légende :

 Unique  Primary Key  Type des données

Description sommaire de la base de données

Durant ce premier semestre, en groupe de 3 (Louis Iwinski, Théo Lorio et Gabriel Bailleul), on nous a demandé de réaliser une activité, dans le cadre du cours Base de données avancées, qui permet de créer une base de données. Nous avons décidé de la nommer « Achats » et elle comporte trois tables (« Client », « Produit » et « Achete ») et une vue (« view\_1 »).

Une image contenant table

Description générée automatiquement

Tableau des trois tables de la base de données « Achats »

Cette base de données pourrait s’apparenter à une base qui pourrait être utilisé dans un magasin de sport et de vêtements. On peut le voir grâce aux tuples insérés dans les différentes tables de base de données.

Par exemple :

Une image contenant texte

Description générée automatiquement

Tableau de tuples de la table Produit

Les tables de cette base de données sont créées avec plusieurs contraintes. Notamment, avec des « NOT NULL » ou encore des « UNIQUE ». Ce qui permet de ne pas faire d’erreurs quand on insère de nouvelles valeurs dans les tables en question.

Une image contenant texte

Description générée automatiquement Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Capture de la création des tables de la base de données

On a donc trois tables. La première se nomme Client est regroupe l’identifiant (id\_client), le nom (nom), le prénom (prenom) ainsi que l’âge (age) du client. La deuxième se nomme Produit et regroupe l’identifiant (id\_produit), le nom du produit (nom\_produit) et le prix en euros (prix\_en\_euros) du produit. Pour finir, on a la table Achète qui regroupe les deux clés primaires des deux autres tables (id\_client et id\_produit).

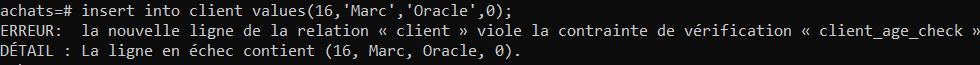
Chaque table comporte quinze tuples différentes.

Tests des contraintes et affichages des erreurs

* Test numéro 4 Client– sur la contrainte NOT NULL sur nom de Client



* Test numéro 6 Client – sur la contrainte de l’age>0 sur age de Client



* Test numéro 3 Produit – Contrainte UNIQUE sur nom\_produit de Produit



* Test numéro 8 Produit – Contrainte varchar(50) sur nom\_produit de Produit



* Test numéro 1 Achete – Contrainte UNIQUE sur (id\_client,id\_produit) de Achete

Une image contenant texte, orange

Description générée automatiquement

Étape 2

Diagramme E/R sans contrainte d’intégrité

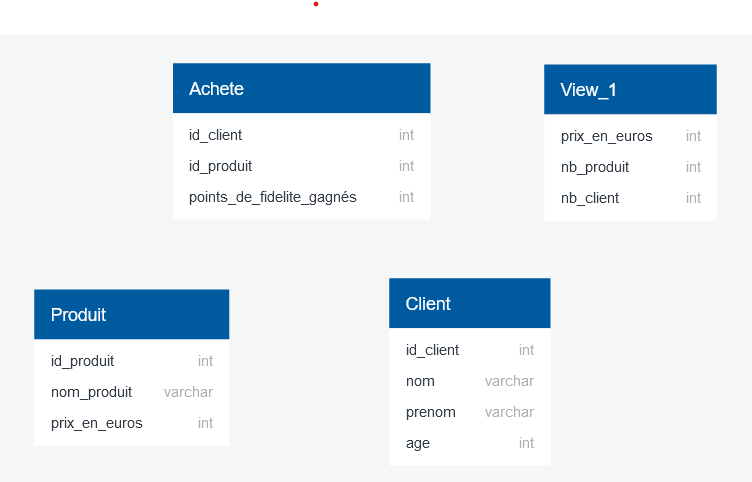


Diagramme E/R de notre base de données Achats sans contraintes d’intégrité réalisé sur QuickDBD

Étape 3

Un trigger et une fonction PL/SQL portant sur une table contenant une clé étrangère

Une image contenant texte

Description générée automatiquement



Sur la capture d’écran au-dessus, nous pouvons voir une fonction PL/SQL qui se nomme test\_achete\_after\_iut(). La fonction permet de tester une insertion ou une mise à jour d’un tuple de la table ACHETE. Celle-ci retourne un trigger. On commence à coder cette fonction par un DECLARE pour déclarer une variable nb en INTEGER. Par la suite, nous faisons une requête SQL pour prendre le tuple qu’on choisit avec un WHERE. Au passage, nous faisons aussi un COUNT(\*) pour savoir le nombre d’id qui sont identiques.

De plus, nous réalisons une boucle « if » qui parcourt la table ACHETE et regarde si le nombre d’id identique est supérieur à 1. Si cela est vrai, alors la fonction retourne un message d’exception.

Par la suite, nous réalisons un PERFORM COUNT(\*) pour exécuter la requête SQL sans récupérer le résultat. On regarde si on le trouve avec un NOT FOUND, s’il ne le trouve pas on envoie un message d’exception aussi.

Le trigger trigger\_achete\_after\_iu permet d'exécuter la procédure test\_achete\_after\_iut() après l'insertion ou une mise à jour d'un tuple de la table ACHETE. À l’aide, notamment, d’AFTER et du FOR EACH ROW EXECUTE PROCEDURE (qui permet de parcourir les lignes de la table une seule fois et de lancer la procédure).

Résultats des commandes :





Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Les deux tuples s’insèrent normalement. Quand on saisit la commande une seconde fois, cela lance l’exception personnalisée grâce au trigger et à la fonction PL/SQL.