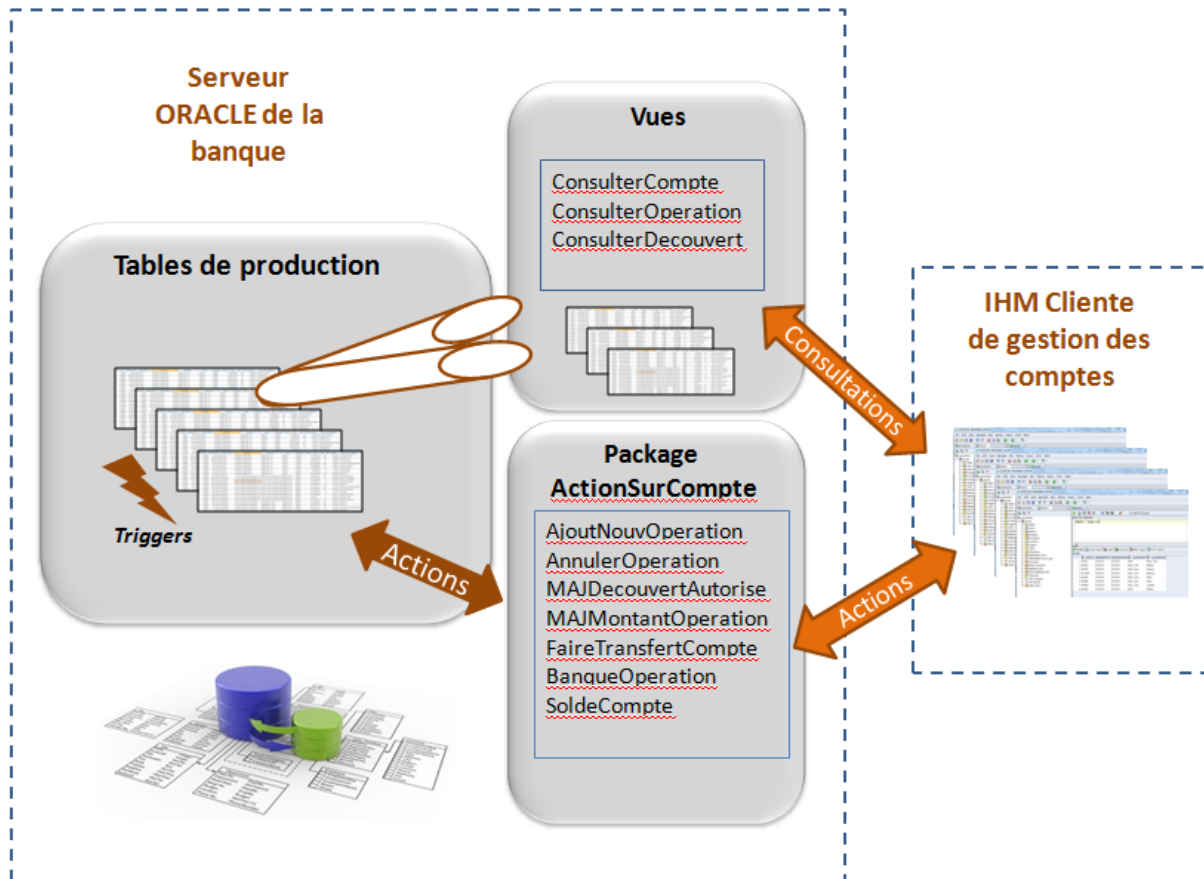


PROJET

Application Bancaire de gestion de comptes

Ce projet porte sur la mise en exploitation de la partie serveur d'une application de gestion de comptes bancaires. Une IHM cliente de gestion de compte a été développée par une entreprise prestataire (hors cadre du projet). Votre travail consiste à mettre en place tous les composants du serveur ORACLE (les tables, les Packages et les vues) pour que cette IHM cliente puisse interagir avec la base de données. Le schéma fonctionnel général de l'application est le suivant :



Cette application contient :

- **Les tables de production** qui contiennent toutes les données bancaires des clients
- **Les vues** qui sont les pseudo-tables consultables par les IHM clientes. Elles permettent de cibler les informations utiles.
- **Le package** qui contient des procédures et fonctions stockées appelables par l'IHM cliente pour réaliser des actions sur les tables de productions.

Pour réaliser cette application, un découpage par lot des tâches est proposé :

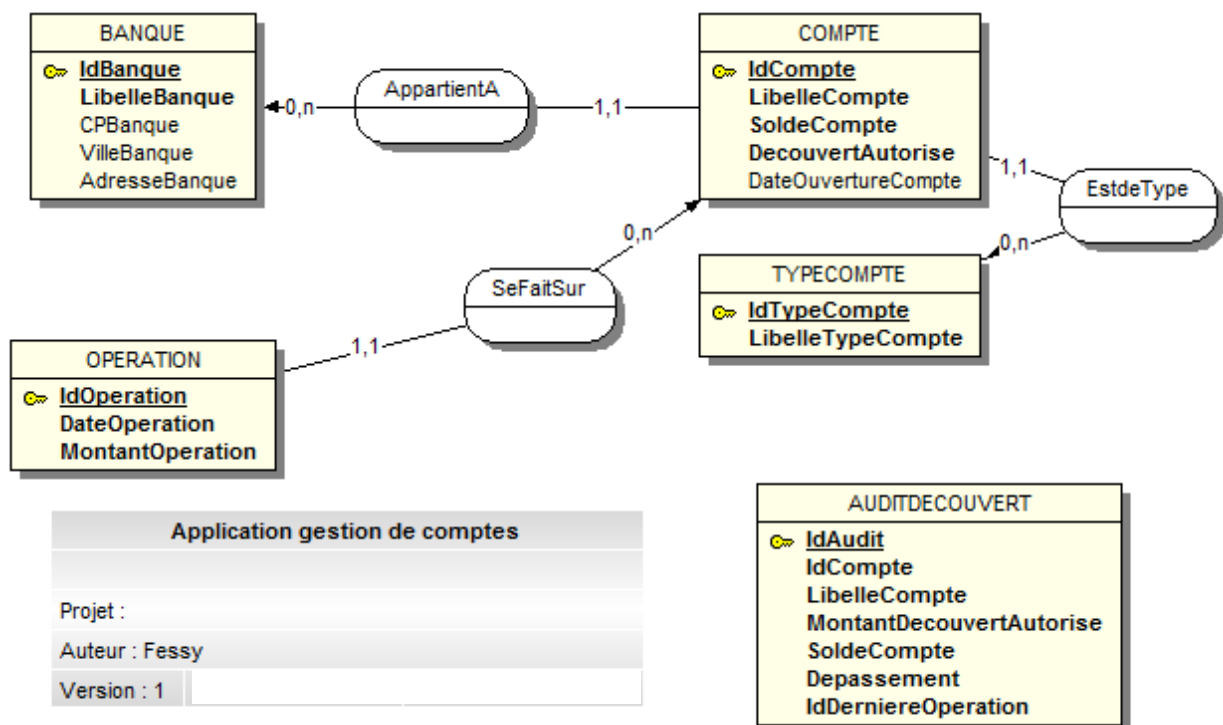
- Lot 1 : Mise en place des tables de production
- Lot 2 : Mise en place du package
- Lot 3 : Mise en place du trigger de calcul de solde
- Compléments : Mise en place des Vues

■ Lot 1 : Mise en place des tables de production

Objectif : Mise en place l'ensemble des tables utiles à l'exploitation

Résultat : Le script de commandes SQL Oracle de création des tables et séquences ainsi qu'un jeu de données de test.

L'équipe de conception vous fournit le modèle conceptuel à implanter :



Dictionnaire de données :

Nomenclature Type de Contraintes

PK : Primary Key	U : Unique	NN : NOT NULL
FK : FOREIGN KEY	CK : CHECK	

« **BANQUE** »

Attribut	Type	Contrainte	Commentaire
IdBanque	INTEGER	PK	Généré par une séquence
LibelleBanque	VARCHAR(50)	U, NN	
CPBanque	CHAR(5)	NN	Code Postale
AdresseBanque	VARCHAR(50)	NN	
VilleBanque	VARCHAR(30)	NN	

« **AUDITDECOUVERT** »

Cette table de travail est alimentée par le trigger du LOT3. Elle identifie les comptes à découvert.

Attribut	Type	Contrainte	Commentaire
IdAudit	INTEGER	PK	Généré par une séquence
IdCompte	INTEGER	NN	Identifiant du compte ayant dépassé son découvert autorisé
LibelleCompte	VARCHAR(30)	NN	Libelle du compte
SoldeCompte	NUMBER(10,2)	NN	Solde de compte. Valeur par défaut 0
DecouvertAutorise	NUMBER(10,2)	NN CK	Decouvert autorisé du compte. Valeur positive
Depassement	NUMBER(10,2)	NN CK	Montant en valeur absolue du dépassement au-delà du découvert autorisé. Valeur positive
IdDerniereOperation	INTEGER	NN	Identifiant de la dernière opération ayant conduit au dépassement du découvert autorisé.

« **TYPECOMPTE** »

Attribut	Type	Contrainte	Commentaire
IdTypeCompte	INTEGER	PK	Généré par une séquence
LibelleTypeCompte	VARCHAR(30)	NN	Libelle du type de compte : LIVRET A, Compte Courant, Livret Développement Durable, PEL, CEL, ...

« **OPERATION** »

Attribut	Type	Contrainte	Commentaire
IdOperation	INTEGER	PK	Généré par une séquence
DateOperation	DATE	NN	Valeur par défaut SYSDATE
MontantOperation	NUMBER(10,2)	NN	Montant de l'opération. Positif pour les crédits, négatif pour les débits.

« **COMPTE** »

Attribut	Type	Contrainte	Commentaire
IdCompte	INTEGER	PK	Généré par une séquence
LibelleCompte	VARCHAR(30)	NN	Numéro de référence du compte

		U	Clé de gestion
SoldeCompte	NUMBER(10,2)	NN	Solde de compte. Valeur par défaut 0
DecouvertAutorise	NUMBER(10,2)	NN	Découvert autorisé pour le compte. Valeur par défaut 0.
DateOuvertureCompte	DATE	NN	Valeur par défaut SYSDATE

■ Lot 2 : Mise en place du package « ACTIONSURCOMPTE »

Objectif : Mise en place l'ensemble des procédures et fonctions stockées accessibles par les applications clientes

Résultat : Le script de commande PLSQL Oracle de création du package

Description du contenu du package :

- ✓ **AJOUTNOUVOPERATION** (*Idcompte INTEGER, Value NUMBER*)
 Cette procédure stockée permet d'ajouter une nouvelle opération dans la table « OPERATION ». Elle prend en paramètre l'identifiant du compte sur lequel l'opération s'effectue et la valeur de l'opération. Celle-ci est positive ou négative selon qu'il s'agisse d'un crédit ou d'un débit.
- ✓ **ANNULEROPERATION** (*IdOpt INTEGER*)
 Cette procédure permet d'annuler une opération de la table « OPERATION ». Elle prend en paramètre l'identifiant de l'opération. Pour annuler une opération, on ne supprime pas l'opération mais on réalise une nouvelle opération d'un montant opposé à celle à annuler.
- ✓ **MAJDECOUVERTAUTORISE** (*Idcompte INTEGER, Value NUMBER*)
 Cette procédure permet de modifier le montant du découvert autorisé d'un compte. Elle prend en paramètre l'identifiant du compte à modifier et la valeur du nouveau montant de découvert autorisé.
- ✓ **MAJMONTANTOPERATION** (*Idoperation INTEGER, Value NUMBER*)
 Cette procédure permet de modifier le montant d'une opération. Elle prend en paramètre l'identifiant de l'opération à modifier et la valeur de son nouveau montant. Pour modifier une opération, on annule l'opération et on insère une nouvelle opération avec le nouveau montant.
- ✓ **FAIRETRANSFERTCOMPTE** (*CptOrig INTEGER, CptDest INTEGER, Value NUMBER*)
 Cette procédure permet de réaliser un transfert d'argent entre comptes. Elle prend en paramètre l'identifiant du compte d'origine, l'identifiant du compte destination et le montant du transfert. Ce montant est toujours positif.
- ✓ **BANQUEOPERATION** (*Idopt INTEGER*) **RETURN VARCHAR**
 Cette fonction permet de retourner le libellé de la banque associée à l'opération dont l'identifiant est passé en paramètre.
- ✓ **SOLDECOMPTE** (*Cpt INTEGER*) **RETURN NUMBER**
 Cette fonction permet de retourner le solde du compte dont l'identifiant est passé en paramètre. Elle retourne un nombre.

■ Lot 3 : Mise en place du Trigger « CALCULSOLDE »

Objectif : Mise en place du déclencheur qui permet de mettre à jour le solde des comptes

Résultat : Le script de commande PLSQL Oracle de création du Trigger

Règles de fonctionnement :

1. A chaque action sur la table opération (ajout d'une opération de débit ou crédit sur un compte, annulation d'une opération ou modification du montant d'une opération) ce traitement calcule automatiquement le nouveau montant du solde du compte associé à l'opération.
2. A chaque action si le découvert autorisé du compte est dépassé, l'opération s'effectue mais le compte est signalé dans la table « AUDITDECOUVERT ». Cette table ne doit contenir qu'une seule ligne par compte. Si le solde d'un compte revient dans la limite du découvert autorisé alors il disparaît de la table « AUDITDECOUVERT ».
3. Les comptes d'épargne ne peuvent pas avoir de solde négatif.

■ Compléments : Mise en place des Vues métier

Objectif : Mise en place des vues accessibles par l'IHM cliente pour visualiser les données

Résultat : Le script de commandes SQL Oracle de création des vues

Description des vues à réaliser :**✓ *CONSULTERCOMPTE***

Cette vue permet d'afficher les informations des comptes : Identifiant du compte, Libelle du compte, libelle du type de compte, solde du compte

✓ *CONSULTERDECOUVERT*

Cette vue permet d'afficher les comptes présents dans la table « AUDITCOMPTE ». Elle affiche les identifiants des comptes, leurs libellés et libellé de type de compte, leur solde et leur dépassement

✓ *CONSULTEROPERATION*

Cette vue permet d'afficher les opérations présentes dans la table « OPERATION ». Elle affiche les identifiants des opérations, leurs montants et dates. Elle trie les opérations depuis les plus récentes au plus anciennes.

Un projet à faire seul pour monter en compétence.

A chaque séance, un suivi de votre avancement sera réalisé par votre chef de projet (M. Fessy) Il permettra de s'assurer de la réalisation de la mission dans les délais.

L'évaluation du projet se fera donc progressivement à chaque séance.