

## TD SEANCE 1 - MODELE CONCEPTUEL

Cette partie est à réaliser sur papier uniquement. Il s'agit de réaliser un modèle conceptuel de données (MCD) suivi du modèle logique de données (MLD) dans le modèle relationnel. Le cas étudié est celui d'une petite entreprise dont les règles de gestion sont énoncées ci-dessous. Dans la suite du TP, une partie de ce MLD sera implémenté sur le SGBD cible.

### Méthodologie :

- Dans un premier temps, nous ferons abstraction des propriétés pour se concentrer sur les entités et les associations.
- Faites un 'brouillon' du MCD contenant uniquement les entités, les associations et essayez de placer certaines propriétés plus complexes que les autres à savoir la prime, la quantité, les remises...
- Rajouter les cardinalités
- Vérifier les règles de gestion ci-dessous (RG 1 à RG 9)
- Vérifier les règles énoncées en cours (règles 1 à 8 du cours)

### Règles de gestion (RG)

- **RG 1** : L'entreprise est organisée en divisions identifiées par un numéro, désignées par un nom. Chaque division est dirigée par un employé de l'entreprise.
- **RG 2** : Un employé est identifié par un matricule, nommé par son nom et prénom. Il travaille dans une seule division où il exerce une fonction qui peut être Président (1 pour l'entreprise), Directeur (1 par division) ou Vendeur. Un employé perçoit un salaire. Les informations suivantes sur les employés seront stockées : adresse complète, téléphone, date de naissance.
- **RG 3** : Un employé perçoit une prime sur les ventes qu'il réalise directement (vendeur) ou indirectement (directeur ou président). La prime directe s'élève à 5% des ventes, les primes indirectes s'élèvent à 1% des ventes.
- **RG 4** : Les clients de l'entreprise sont identifiés par un numéro. Les informations suivantes sur les clients seront stockées : nom, prénom, adresse complète, téléphone.
- **RG 5** : Une vente est identifiée par un numéro de vente. Elle est réalisée par un employé pour un client. De plus les informations suivantes seront stockées : date de la vente, date de livraison, adresse complète de livraison (incluant nom, prénom, adresse et téléphone), frais de port.
- **RG 6** : Une vente concerne un ou plusieurs produits. Pour chaque produit vendu, on indiquera la quantité vendue. Une remise peut être effectuée en fonction de cette quantité. C'est l'employé (vendeur) qui décide au cas par cas du taux de cette remise.
- **RG 7** : Un produit est identifié par un numéro. Il possède un nom et un prix unitaire. Il est fourni par un seul fournisseur.
- **RG 8** : Un fournisseur est identifié par un numéro. Les informations suivantes sur les fournisseurs seront stockées : nom, prénom, adresse complète, téléphone.
- **RG 9** : Une vente est livrée par un transporteur (DHL, la poste, etc.). Un transporteur est identifié par un numéro, possède un nom, un prénom, une adresse complète et un téléphone.

## TD SEANCE 2 - ALGEBRE RELATIONNELLE

Union :  $R \cup S$

Différence :  $R - S$

Projection :  $\prod_{\text{Liste d'attributs}} (R)$

Restriction :  $\sigma_{\text{Critère}} (R)$

Produit cartésien :  $R \times S$

Intersection :  $R \cap S$

Jointure :  $R \bowtie S$

Jointure :  $R \div S$

**Exercice 1.** Soient  $R(X,Y)$  et  $S(X, Z)$  2 relations. Formulez les expressions suivantes à l'aide des opérateurs de base  $\sigma, \pi, \cup, -, \times$  : Jointure  $R \bowtie S$  ; Semi-Jointure  $R \ltimes S$  ; Anti-Jointure  $R \not\bowtie S$  ; Jointure externe gauche  $R \ltimes\ltimes S$  ; Jointure externe complète  $R \ltimes\ltimes\ltimes S$ .

**Exercice 2.** Soient les relations suivantes :

Res. (R)	Cours	Etudiant	Note
	Math	Toto	A
	Math	Lulu	B
	Latin	Toto	A
	Physique	Toto	A

Ens. (S)	Cours	Prof
	Math	Martin
	Physique	Dupont
	Latin	Martin

Calculer et exprimer en Français :  $\sigma_{\text{Cours} = \text{Math}} (R)$  ;  $\prod_{\text{Cours}} (S) - \prod_{\text{Cours}} (R)$  ;

$\prod_{\text{Cours}} (R) - \prod_{\text{Cours}} (S)$  ;  $\prod_{\text{Etudiant}} (R \div (\prod_{\text{Cours}} (S)))$  ;  $\prod_{\text{Prof}} (\sigma_{\text{Note} = \text{A et Etudiant} = \text{Toto}} (R \bowtie S))$ .

**Exercice 3.** Soient le schéma relationnel suivant :

SALLE	(Nom, Horaire, Titre)	FILM	(Titre, Description, Réalisateur)
PRODUIT	(Producteur, Titre)	VU	(Spectateur, Titre)
AIME	(Amateur, Titre)	JOUE	(Acteur, Titre)

Ecrire les requêtes suivantes en algèbre relationnelle :

1. Où et à quelle heure peut-on voir le film Alien ?
2. Quels sont les films réalisés par Eric Rohmer ?
3. Quels sont les acteurs du film Brazil ?
4. Où peut-on voir un film avec Fernandel (acteur) ?
5. Où peut on voir Binoche (actrice) après 16h ?
6. Quels sont les acteurs qui ont produit un film ?
7. Quels sont les acteurs qui produisent un film dans lequel ils jouent ?
8. Quels acteurs jouent dans les films de Blier (réalisateur) ?
9. Quels acteurs ne jouent que dans les films de Blier (réalisateur) ?
10. Quels acteurs jouent dans tous les films de Blier (réalisateur) ?
11. Quels producteurs voient tous les films de Coline Serreau (réalisatrice) ?
12. Quels sont les spectateurs qui aiment tous les films qu'ils voient ?
13. Quels films ne passent dans aucune salle ?
14. Qui produit un film qui ne passe dans aucune salle ?
15. Quels sont les producteurs qui voient tous les films qu'ils produisent ?
16. Quels spectateurs aiment un film qu'ils n'ont pas vu ?
17. Qui n'aime aucun film ?
18. Quel producteur ne produit aucun film de Alan Parker (réalisateur) ?
19. Quels sont les acteurs qui produisent un film qu'ils ont réalisé ?
20. Quels sont les producteurs qui ne voient que les films qu'ils produisent ?

**Exercice 4.** Si  $R(X,Y)$  et  $S(Y)$  sont deux relations, comment exprimer le quotient  $Q(X) = R \div S$  en fonction des opérateurs relationnels de base  $\sigma, \pi, \cup, -, \times$ .

## SEANCE 3 : CREATION DU SCHEMA ET INSERTION

### Schéma et description:

Table client :	<b>CLI</b> (CodeCli, NomCli, Pays)
Table commande :	<b>COM</b> (NumCom, CodeCli, FraisPort, AnCom)
Table détail commande :	<b>DET</b> (NumCom, NumPro, Qte, Remise)
Table produit :	<b>PRO</b> (NumPro, NomPro, TypePro)

Code du client : CodeCli, Nom du client : NomCli, Pays du client : Pays, Numéro de la commande : NumCom, Frais de port : FraisPort, Année de commande (ex 1992) : AnCom, Numéro de produit : NumPro, Quantité : Qte, Remise effectuée : Remise, Nom du produit : NomPro, Type de produit : TypePro.

### Partie 1: Création du schéma de la base

**Question 1.** Dessiner sur papier le schéma de la base en faisant ressortir les clés primaires (en souligné) et étrangères (en souligné et pointillé)

**Question 2.** Créer les relations de la base ci-dessus (pensez aux clés primaires et étrangères)

**Exemple.** create table COM (  
    NumCom number(5) constraint Cle\_Com primary key,  
    NumCli [number(5)] constraint Com\_Ref\_Cli references CLI,  
    FraisPort number(3,2) constraint FraisPort\_Not\_Null not null  
    AnCom number(4));

Les types disponibles sont: number(n), char(n), date, number(m,n)

Faire attention à la cohérence de type des attributs de jointures (même domaine).

Consulter la liste des types sur le manuel Oracle.

Modifier le type des attributs : ALTER TABLE <nom table> MODIFY ( <attr> newtype, ...)

Ajouter un attribut à une relation : ALTER TABLE <nom table> ADD ( <newattr> type, ...)

Valeurs nulles : un attribut d'une table déclaré "NOT NULL" doit nécessairement être renseigné lors de l'insertion d'un tuple.

**Question 3.** Y a-t-il un ordre à respecter lors de la création de ces tables, si oui lequel ? Pourquoi ?

**Question 4.** Vérifier vos créations: sous SQL\*plus, utiliser la commande : desc <nomtable>

**Question 5.** Modifier un type précédemment défini : NomCli devient char(30).

**Question 6.** Ajouter un attribut Téléphone (Tel) pour les clients

**Question 7.** Ajouter le prix unitaire (PrixUnit) pour les produits

**Question 8.** Rajoutez une contrainte de non nullité à l'attribut NomPro

**Question 9.** Insérer un jeu de données cohérent dans vos relations (1 ou 2 tuples par relation)

**Question 10.** Mettre en majuscule les noms des clients (fonction UPPER)

**Question 11.** Vider toutes vos tables. Y a-t-il un ordre à respecter ? Si oui, pourquoi ?

**Question 12.** Supprimer toutes vos tables. Y a-t-il un ordre à respecter? Si oui, pourquoi ?

### Partie 2: Chargement des données

**Question 13.** Lancer le script CREATE.sql (fourni)

**Question 14.** Charger les cinq relations avec SQL loader (à partir d'un terminal).

Syntaxe : sqlldr <login> <chemin\_fichier\_input>.ctl <chemin\_fichier\_output>.log

Remarque : le format du fichier peut poser problème (caractère de fin de chaîne différent entre Windows/Linux). Il peut être nécessaire de modifier les fichiers et de les sauver à nouveau sous Linux, ou de les faire passer par un utilitaire type « dos2unix ».

Exemples :

LOAD DATA	Charger des données
INFILE *	qui se trouvent dans ce fichier
APPEND	On veut les ajouter dans la table...
INTO TABLE CLI	...dans la table CLI
FIELDS TERMINATED BY ","	OPTIONALLY ENCLOSED BY ''''
[ Attention ''''	signifie quote/guillemet/quote ]
Les champs sont terminés par des virgules...	
...encadrés éventuellement par des guillemets	
(CodeCli,NomCli,Pays,Tel)	Liste des attributs indiqués ci-dessous
BEGINDATA	Les données commencent ici
1, "Maria", "Pologne", "4867427275"	1er tuple => codecli = 1, nomcli = Maria, etc.
2, "Ana", "Autriche", "4905219433"	2ème tuple => ...
3, "Antonio", "Espagne", "7062001573"	3ème tuple => ...

**Question 15.** Y a-t-il un ordre à respecter lors du chargement des relations, si oui pourquoi...

**Question 16.** Vérifier que vous avez l'intégralité des données : 91 Clients, 1078 Commandes, 2809 Détails, 77 Produits et 29 Fournisseurs. Il vous en manque encore ?

1. Vérifier les logs créés lors du chargement (exemple : more fichier\_output.log)
2. Trouvez le problème et résolvez-le (le cas échéant en rechargeant les données corrigées après les avoir supprimées).

### Partie 3: Mises à jour des données

**Question 17.** Lancer le script CLEAN.sql.

Multiplier par deux les remises associées aux commandes passées après 2006

**Question 18.** Supprimer les clients français

**Question 19.** Supprimer les clients qui ont commandé avant 2001

## TD-TP DE BASES DE DONNÉES SEANCE 4 : MANIPULATION DES DONNEES

### Préparation:

L'objectif du TD est d'interroger en SQL la base de données créée lors des séances précédentes. Charger le schéma et la base de données en lançant le script CLEAN.sql.

### Répondez aux questions suivantes en interrogeant votre base :

1. Quelle est la liste des clients ?
2. Quelle est la liste des clients français triés par ordre alphabétique ?
3. Quels sont les noms des clients français ou anglais ?
4. Quelle est la liste des produits vendus à "Antonio" ?
5. Quels sont les noms des clients ayant commandé en 2002 ou 2003 ?
6. Quelle est la liste des fournisseurs de produits coutant de moins de 50 euros?
7. Quels fournisseurs offrent à la fois des produits à moins de 50 et à plus de 400 euros ?
8. Quels sont les fournisseurs qui fournissent le produit 'Chef' le moins cher ?
9. Pour chaque fournisseur, quelle est la liste des produits fournis par ordre alphabétique ?
10. Quels sont les fournisseurs qui sont aussi clients (même nom) ?
11. Quels sont les fournisseurs qui ne fournissent aucun produit à moins de 50 euros?
12. Quel est le nombre de produits dans la base ?
13. Quelle est pour chaque client, la liste des produits commandés et leur nombre pour l'ensemble des commandes passées ?
14. Quel est, pour chaque ligne de commande, le prix total à payer (penser à la remise et à la quantité) ?
15. Quel est le total à payer par commande ?
16. Quel est le total à payer par client ?
17. Quels sont les clients qui ont commandé pour plus de 10000 euros?

## TD-TP DE BASES DE DONNEES SEANCE 5 : PROGRAMMATION SQL – PL/SQL

L'objectif de cette séance est de se familiariser avec la programmation SGBD avec PL/SQL. Pour cela, vous allez créer des scripts PL/SQL (nommés avec une extension en « .sql ») que vous déclencherez avec la commande @nom\_fichier dans SQL\*Plus. Ces blocs PL/SQL peuvent à terme être intégrés au serveur sous forme de procédures stockées.

### Q1 : Phase préliminaire et prise en main

Créer une table RESULTAT suivante : RESULTAT (CODE number, MESSAGE char(50)). Compléter le programme PL/SQL ci-dessous pour qu'il génère le résultat suivant.

PROGRAMME PL/SQL	Contenu table RESULTAT après exécution										
<pre> DELETE FROM RESULTAT ; PROMPT Nombre de lignes à produire ACCEPT n DECLARE     x NUMBER:= ????????; BEGIN     FOR i IN 1..&amp;n LOOP         IF MOD(x, ??????) = 0 THEN -- x is even             INSERT INTO RESULTAT VALUES (????????????);         ELSE             INSERT INTO RESULTAT VALUES (????????????);         END IF;         x := x ????????????;     END LOOP;     COMMIT; END; / SELECT * FROM RESULTAT ; </pre>	<table> <thead> <tr> <th>CODE</th><th>MESSAGE</th></tr> </thead> <tbody> <tr> <td>1</td><td>100 is even</td></tr> <tr> <td>2</td><td>101 is odd</td></tr> <tr> <td>3</td><td>102 is even</td></tr> <tr> <td>...</td><td>etc</td></tr> </tbody> </table>	CODE	MESSAGE	1	100 is even	2	101 is odd	3	102 is even	...	etc
CODE	MESSAGE										
1	100 is even										
2	101 is odd										
3	102 is even										
...	etc										

### Q2 : Bloc PL/SQL, expression conditionnelle

Charger les scripts CREATE-bis.sql et CREATE-bis-data.sql. Ecrire un programme PL/SQL qui :

- a) demande un numéro de client
- b) insère un tuple dans la table RESULTAT et le visualise. Ce tuple est tel que :
  - s'il n'y a pas de commandes pour ce client, il a comme valeur 'pas de commande'
  - sinon, il a comme valeur 'n commandes pour le client X', avec n le nombre de commandes pour X.

### Q3 : Bloc PL/SQL, curseurs et boucles

Ecrire un programme qui permette d'afficher les nième et n+1ième commandes les plus récentes (afficher le numéro de la commande, la date, et le prix total) et d'insérer les informations correspondantes dans la table RESULTAT. Le prix total de la commande sera calculé à partir des détails et des prix des produits commandés. Le nombre n est un paramètre saisi par l'utilisateur.

### Q4 : Bloc PL/SQL, curseurs paramétrés

Ecrire un programme qui utilisera un curseur paramétré et qui permette de vérifier que le montant payé (COM.Payment) pour une commande donnée (renseignée par l'utilisateur) est égal à la somme des lignes de commandes correspondantes. Pour la commande demandée, une ligne sera insérée dans la table résultat. Cette ligne aura la forme suivante : « NumCom : 100 - Payement : 5000 - Prix : 5000 ».

### Q5 : Bloc PL/SQL, exceptions

Compléter le programme de la question 3 pour gérer l'erreur survenant dans le cas où le nombre N serait strictement supérieur au nombre de commandes dans la table Commandes. Dans ce cas, un message d'erreur sera inséré dans la table résultat.

### Q6 : Bloc PL/SQL, packages

Ecrire le package comportant une procédure de tri des clients dans l'ordre décroissant de leur montant total de commandes qui fournit la liste triée des identifiants clients et du montant total de leurs commandes dans la table résultat, et une fonction qui renvoie l'identifiant du client ayant le montant de dépenses le plus élevé.

## TD-TP DE BASES DE DONNEES - IN207

### EXPERIENCES SUR LA CONCURRENCE

#### Préparation

Ajouter un attribut numérique à votre table CLI, que vous nommerez « Solde », et insérer un client nommé 'Joe' (attention au majuscules/minuscules). Récupérer ensuite les 3 fichiers nommés Solde.sql, Depot.sql et Retrait.sql.

Ouvrez deux fenêtres SQL\*PLUS. Chacune sera considérée par Oracle comme pilotant une session (une transaction) différente, en situation de concurrence l'une avec l'autre.

Dans tout ce qui suit, on note INSTRi l'exécution de l'instruction INSTR dans la session i. Par exemple Depot1 correspond à l'exécution du fichier Depot dans la première fenêtre par la commande START Depot. On note de même ROLi et COMi l'exécution des commandes rollback; et commit; dans la session i.

#### Question 1 : Observez le fonctionnement du verrouillage d'Oracle par défaut

Vous réaliserez les séquences suivantes en notant ce qu'Oracle affiche comme résultat.

**Scénario 1.** L'utilisateur 1 effectue un retrait sur le compte de 'Joe', l'autre ne fait que consulter les soldes :

COM1, COM2, Solde1, Solde2, Retrait1, Solde2, ROL1, Solde1, Solde2.

**Scénario 2.** Idem, mais avec un commit :

COM1, COM2, Solde1, Solde2, Retrait1, Solde2, COM1, Solde1, Solde2.

**Scénario 3.** L'utilisateur 1 fait un retrait sur le compte de 'Joe', alors que l'utilisateur 2 crédite le compte :

COM1, COM2, Solde1, Solde2, Retrait1, Solde2, Depot2, Solde1, Solde2, COM1, COM2.

#### Question 2 : Analyse du comportement d'Oracle

Concluez sur le fonctionnement du verrouillage d'Oracle dans le mode par défaut.

Pour y parvenir, vous pourrez répondre dans un premier temps aux questions suivantes :

1. ORACLE garantit-il la sérialisabilité des transactions ?
2. A quel niveau d'isolation de la norme SQL ce comportement peu correspondre?
3. Oracle utilise-t-il un protocole multi-versions ?

#### Question 3 : Niveau d'isolation SERIALIZABLE

Expérimentez les exécutions précédentes dans le niveau d'isolation "serializable". Pour passer dans ce niveau, la première instruction de la transaction doit être : SET TRANSACTION ISOLATION LEVEL SERIALIZABLE.

**ATTENTION** : non seulement cette instruction de changement du degré d'isolation doit être la première de la transaction, mais sa portée n'est que la transaction en cours. Il faut donc la

relancer systématiquement après le COMMIT, autrement une nouvelle transaction démarre en degré d'isolation par défaut.

Comparer et expliquer comment Oracle verrouille les données dans ce mode.

#### Question 4 : Inter-blocage (DEADLOCK)

Proposer un scénario, le plus simple possible, permettant de générer un inter-blocage entre deux transactions.

#### Question 5 : Niveau SQL REPEATABLE READ et READ UNCOMMITTED

Ces niveaux sont-ils disponibles dans Oracle ? D'après vous, pourquoi ?

#### Question 6 : Niveau d'isolation READ ONLY

Un niveau d'isolation supplémentaire existe dans Oracle, il s'agit du niveau READ ONLY. Pour passer dans ce niveau, la première instruction de la transaction doit être : SET TRANSACTION READ ONLY;

Proposer une expérience vous permettant de tester le fonctionnement de ce niveau d'isolation. Peut-on voir (i.e., faire apparaître) des données "sales" ? Des lectures "non répétables" ? Des "fantômes" ?

A votre avis, quel est l'intérêt de ce niveau d'isolation ?

#### Question 7 : Clause FOR UPDATE

Modifier le script Solde.sql en ajoutant la clause FOR UPDATE. Expérimentez à nouveau les exécutions précédentes, et conclure sur la nouvelle stratégie appliquée par Oracle en présence de la clause FOR UPDATE.