

Feuilles de Cours

Cours Bases de Données 2 - X32I040

Feuille de cours 1 : Rappel Algèbre Relationnelle et SQL
Feuille de cours 2 : Contrôle d'accès
Feuille de cours 3 : Evaluation de requêtes
Feuille de cours 4 : Transactions

Feuille de cours 1

Sur le modèle relationnel et l'algèbre relationnelle

1. Le modèle Entité-Association (EA) ou Entité-Rélation (ER) a plusieurs concepts de base. Lier le concept de gauche avec sa définition à droite.

- | | |
|----------------------------------|---|
| 1. Entité | a. Attribut(s) qui permet de retrouver une instance d'entité unique à tout instant. |
| 2. Attribut | b. Lien entre classes ou instances. Il peut contenir des attributs. |
| 3. Clé | c. Le nombre d'instances d'une entité pour chaque instance de l'autre entité. |
| 4. Association | d. Objet du monde réel que l'on souhaite représenter et qui peut être identifié. |
| 5. Cardinalité d'une association | e. Description des propriétés des entités. |
| 6. Domaine | f. Ensemble nommé de valeurs. |

2. Le modèle relationnel a plusieurs concepts de base. Lier le concept de gauche avec sa définition à droite.

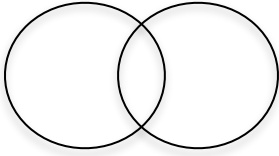
- | | |
|-------------------------------|--|
| 1. Relation | a. Le nom d'une relation et ses attributs. |
| 2. Tuple | b. Entité dans le modèle EA. |
| 3. Degré d'une relation | c. Instance d'une relation qui doit être unique. |
| 4. Schéma | d. Nombre d'attributs d'une relation notée $\delta(R)$. |
| 5. Cardinalité d'une relation | e. Nombre de tuples notée : $\text{Card}(R)$ ou $ R $. |

3. En algèbre relationnelle il a des opérations de base et des opérations dérivées. Souligner les opérations de base à partir desquelles on peut en dériver d'autres opérations.

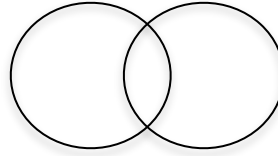
- | | |
|-------------------------------|-----------------------|
| a. Projection π | g. Complément |
| b. Sélection σ | h. Jointure naturelle |
| c. Union \cup | i. Jointure interne |
| d. Différence $-$ | j. Auto jointure |
| e. Produit cartésien \times | k. Semi jointure |
| f. Intersection \cap | l. Jointure externe |

4. Soit $R(A,B,C)$ et $S(A, B, C)$. Remplir les cercles avec le résultat correct des opérations suivantes.

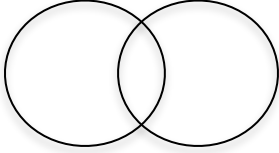
$R-S$



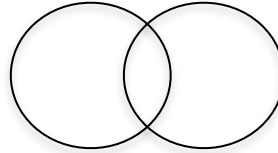
$S-R$



$R-(R-S)$

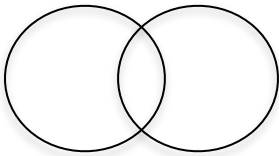


$R \cap S$

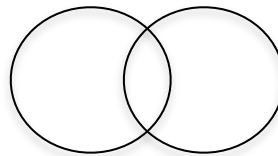


Soit $R(A,B,C)$ et $S(A,C)$

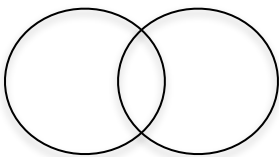
$R \bowtie S$



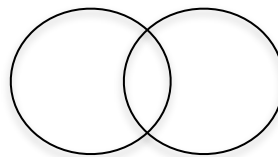
$R \bullet \bowtie_{A=A} S$



$R \bowtie \bullet_{A=A} S$



$R \bullet \bowtie \bullet_{A=A} S$



5. En algèbre relationnelle, comment peut-on simplifier les expressions suivantes en utilisant les jointures.

a) $\pi_{R.A,R.B,R.C} \cup (S.A, S.C) (\sigma_{R.A=S.A \wedge R.C=S.C} (R \times S)) \equiv$

b) $\pi_{R.A,R.B,R.C} (\sigma_{R.A=S.A \wedge R.C=S.C} (R \times S)) \equiv$

Sur les dépendances fonctionnelles et les formes normales.

Soit $R(U)$ un schéma de relation.
Soit r une instance de R .
Soient X et Y deux sous-ensembles de U .

1. Lier le coté gauche avec le coté droit pour compléter les définitions.

- | | |
|---|--|
| 1. $X \rightarrow Y$ est vraie sur r ssi | a. l'ensemble de toutes les df déductibles à partir de DF avec les règles de dérivation. Elle comprend toutes les dfs déductibles, y compris les plus triviales. |
| 2. $X \rightarrow A$ est élémentaire ssi | b. il n'existe pas de sous-ensemble de X qui détermine A . |
| 3. La fermeture DF^+ de DF est | c. a partir d'un ensemble de df élémentaires on peut composer par transitivité d'autres df élémentaires. |
| 4. La fermeture transitive veut dire que | d. tous les tuples de R qui ont même valeur pour X ont même valeur pour Y . |
| 5. La couverture minimale est | e. l'ensemble minimal d'attributs qui détermine tous les autres attributs. |
| 6. Une df est trivial si | f. tous les attributs déterminés par X . |
| 7. La fermeture d'un ensemble d'attributs X^+ est | g. le coté droit est un sous-ensemble du coté gauche. |
| 8. La clé d'une relation est | h. le sous-ensemble minimal de df permettant de générer toutes les autres. |
| 9. Une dépendance est trivial ssi | i. le coté droit est un sous ensemble du coté gauche. |

2. Compléter les dérivations suivantes en choisissant la dernière partie de la définition parmi les choix de droite :

- | | |
|--|---|
| Réflexivité : si $X \subset Y$ | a) alors $Y \rightarrow X$ |
| Transitivité : si $X \rightarrow Y$ et $Y \rightarrow Z$ | b) alors $X \rightarrow Z$ |
| Augmentation : si $X \rightarrow Y$ | c) alors $X, Z \rightarrow YZ$ |
| Union : si $X \rightarrow Y$ et $X \rightarrow Z$ | d) alors $X \rightarrow Y, Z$ |
| Pseudo-transitivité : si $X \rightarrow Y$ et $W, Y \rightarrow Z$ | e) alors $XW \rightarrow Z$ |
| Décomposition : si $X \rightarrow Y$ et $Z \subset Y$ | f) alors $X \rightarrow Z$ |
| Split : si $A \rightarrow B, C$ | g) alors $A \rightarrow B$ et $A \rightarrow C$ |
| Trivial : si $X \rightarrow Y$ et $Y \subset X$ | h) alors $X \rightarrow X \cup Y$ |

3. Analyser l'instance suivante de la relation Notes et cocher les dépendances fonctionnelles qui s'y appliquent.

Notes								
noetu	nom	prenom	diplôme	codemat	titre	resp	cc	exam
27845E	Dupont	Isabelle	L3 info	X6I0050	BD2	E512	15	12
27845E	Dupont	Isabelle	L3 info	X6I0010	Prog fonct	E860	10	11
27845E	Dupont	Isabelle	L3 info	X6I0030	RO	E100	11	13,5
34561C	Legarec	Marc	L3 info	X6I0050	BD2	E512	10,2	11
34561C	Legarec	Marc	L3 info	X6I0010	Prog fonct	E860	10	13
34561C	Legarec	Marc	L3 info	X6I0030	RO	E100	12	10
45678D	Martin	Robert	L3 info	X6I0050	BD2	E512	10	8
45678D	Martin	Robert	L3 info	X6I0030	RO	E100	5	5
64289C	Dupont	Letitia	L3 miage	X6IM020	IHM	E350	15	10
64289C	Dupont	Letitia	L3 miage	X6IM010	Gestion	E670	12,5	10
64289C	Dupont	Letitia	L3 miage	X6I0050	BD2	E512	13,1	12
23456E	Robert	Christine	L3 miage	X6IM010	Gestion	E670	7	2
23456E	Robert	Christine	L3 miage	X6I0080	Fichier	E900	14	13,8
98076E	Robert	Christine	L3 miage	X6IM020	IHM	E350	12	12,8

- | | |
|---|---|
| a. <input type="checkbox"/> noetu -> nom, prenom,diplome; | g. <input type="checkbox"/> nom -> prénom |
| b. <input type="checkbox"/> codemat ->titre,resp; | h. <input type="checkbox"/> resp -> cc, exam |
| c. <input type="checkbox"/> titre ->resp; | i. <input type="checkbox"/> diplome -> noetu |
| d. <input type="checkbox"/> noetu,codemat ->cc,exam; | j. <input type="checkbox"/> titre, codemat -> noetu |
| e. <input type="checkbox"/> Diplome,titre ->codemat, resp; | k. <input type="checkbox"/> cc -> diplome |
| f. <input type="checkbox"/> noetu,codemat,nom,prenom ->cc,exam,titre; | l. <input type="checkbox"/> noetu -> nom |

4. Cocher « vraie » si l'affirmation est vraie, « faux » autrement.

1. Une relation est en 1FN si tout attribut contient une valeur atomique.
Vraie Faux
2. Une relation est en 2FN ssi elle est en première forme normale et tout attribut n'appartenant pas à une clé ne dépend pas d'une partie de la clé.
Vraie Faux
3. La 2FN élimine les anomalies créées par des dépendances entre parties de la clé et attributs non clé.
Vraie Faux
4. Une relation est en 3FN ssi elle est en deuxième forme normale et tout attribut n'appartenant pas à une clé ne dépend pas d'un autre attribut non clé.
Vraie Faux
5. La 3FN élimine les anomalies créées par des dépendances entre les attributs non clés.
Vraie Faux
6. Une décomposition d'une relation R avec l'ensemble DF de dépendances fonctionnelles en R_1, R_2, \dots, R_n avec DF_1, DF_2, \dots, DF_n est sans perte de dépendances fonctionnelles (spdf) ssi $(DF_1 \cup DF_2 \cup \dots \cup DF_n)^+ = DF^+$
Vraie Faux
7. Une décomposition d'une relation R en R_1, R_2, \dots, R_n est sans perte d'information (spi) ssi pour toute extension de R, $R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_n$.
Vraie Faux
8. Toute relation a au moins une décomposition en 3FN telle que la décomposition (a) préserve les DF (spdf) et (b) est sans perte d'information (spi).
Vraie Faux
9. La 3FN permet des dépendances de parties de clés entre elles.
Vraie Faux
10. La 3FN permet des dépendances d'attributs non clé vers une partie de la clé.
Vraie Faux
11. Une relation est en FNBACK ssi les seules dépendances fonctionnelles élémentaires sont celles dans lesquelles une clé entière détermine un attribut.
Vraie Faux
12. Toute relation a au moins une décomposition en FNBACK telle que la décomposition est sans perte d'information (spi).
Vraie Faux

Feuille de cours 2

Sur le contrôle d'accès

Considérez les tables suivantes et les utilisateurs Joe, Michel, Yuppy, Bill, etc.

Sailors	Boats	Reservs
<ul style="list-style-type: none">• <u>sid</u>: integer• sname: string• rating: integer• age: real	<ul style="list-style-type: none">• <u>bid</u>: integer• bname: string• color: string	<ul style="list-style-type: none">• <u>sid</u>: integer• <u>bid</u>: integer• day: date

L'utilisateur Joe crée l'ensemble des tables et exécute les commandes suivantes. Expliquez ces commandes :

1. GRANT INSERT, DELETE ON Reservs TO Yuppy WITH GRANT OPTION
2. GRANT SELECT ON Reservs TO Michel
3. GRANT SELECT ON Sailors TO Michel WITH GRANT OPTION
4. GRANT INSERT (sid,sname,age) ON Sailors TO Michel
5. GRANT REFERENCES (bid) ON Boats TO Bill
6. GRANT REFERENCES ON Boats TO Yuppy

Quelle est la différence entre les options "WITH GRANT OPTION" et "WITH ADMIN OPTION" ?

Choisir la réponse correcte et lorsqu'elle est négative dire quel droit serait-il nécessaire.

Question	Réponse
1. Est-ce que Yuppy peut exécuter la requête suivante : SELECT * FROM Reservs ;	Oui Non
2. Est-ce que Michel peut exécuter la requête suivante : SELECT * FROM Reservs WHERE day > '01-JAN-14' ;	Oui Non

Question

Réponse

- | | |
|---|-----------|
| 3. Est-ce que Michel peut attribuer le droit suivant :
GRANT SELECT ON Sailors TO Bill ; | Oui Non |
| 4. Est-ce que Yuppy peut attribuer les droits suivants :
GRANT ALL ON Reservas TO Bill ; | Oui Non |
| 5. Est-ce que Michel peut exécuter la requête suivante :
GRANT SELECT ON Reservas TO Bill ; | Oui Non |
| 6. Est-ce que Bill peut créer la table suivante :
CREATE TABLE Reservas (sid INTEGER,
bid INTEGER,
day DATE,
PRIMARY KEY (bid,day),
FOREIGN KEY (sid) REFERENCES Sailors,
FOREIGN KEY (bid) REFERENCES Boats) ; | Oui Non |

Considérer en plus la commande suivante :

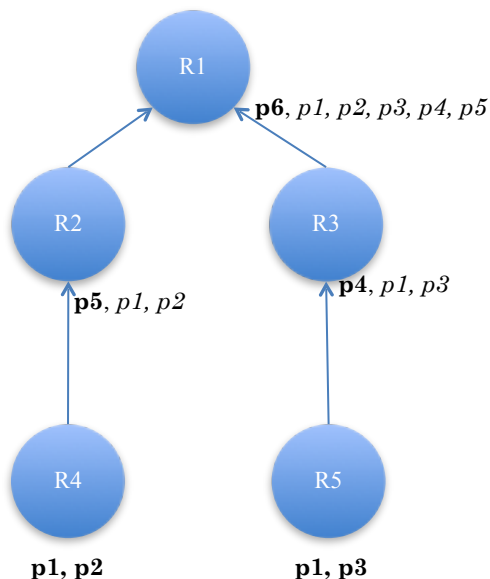
Michel exécute :
GRANT SELECT ON Sailors TO Léa;

Qu'est-ce qui se passe si Joe exécute :

1. REVOKE SELECT ON Sailors FROM Michel;
2. REVOKE SELECT ON Sailors FROM Michel CASCADE ;
3. REVOKE GRANT OPTION FOR SELECT ON Sailors FROM Michel CASCADE ;

Sur les rôles et les graphes de rôles

1. Considérer le graphe A et modifiez-le pour en supprimer le privilège p3 du rôle R5.



Feuille de cours 3 : éval. de requêtes

Les statistiques dans les étapes du traitement des requêtes

1. Le traitement des requêtes est fait en trois étapes, entourez les étapes qui prennent en compte les statistiques collectées par les SGBD (Système de Gestion de Bases de Données)

Analyse/traduction -> Optimisation -> Exécution

2. Citer quelques unes des statistiques collectées par les SGBD sur :

a. les relations :

b. les attributs :

Coût des opérateurs

Lier l'opérateur avec l'estimation de son coût.

- | | |
|--|-------------------------------|
| 1. Sélection sur R (attribut A) avec parcours séquentiel | a. $O(n+m)$ |
| 2. Sélection sur R (attribut A) avec index B+ sur A | b. $O(n)$ |
| 3. Projection de R sans élimination de doublons | c. $O(\log(n))$ |
| 4. Projection de R avec élimination des doublons et tri | d. $O(n)$ |
| 5. Jointure : boucles imbriquées | e. $O(n \log(n))$ |
| 6. Jointure : boucles imbriquées avec un index | f. $O(n*m)$ |
| 7. Jointure : tri fusion (relations pas triées) | g. $O(n*\log(m))$ |
| 8. Jointure : tri fusion (relations triées) | h. $O(n*\log(n) + m*\log(m))$ |

Taille de relations intermédiaires

La taille des relations intermédiaires dépend de l'opération à exécuter. Dans la suite, donner la taille d'une relation intermédiaire pour les opérations suivantes.

1. Projection :

Dans le cas de la sélection, a *selectivité* (S_σ) est le pourcentage de lignes concernées par la sélection.

2. $S_{\sigma(A=valeur)} =$

3. $S_{\sigma(A > valeur)} =$

4. S d'une conjonction (AND) =

5. S d'une disjonction (OR) =

6. Taille d'une jointure R et S

a.

b.

Feuille de cours 4 : Transactions

Concepts ACID

Les transactions sont caractérisées par les propriétés dites ACID. Décrire ces propriétés et dire quelles techniques permettent de garantir chaque propriété.

1. Atomicité
2. Cohérence
3. Isolation
4. Durabilité

Ordonnements

Equivalence entre 2 ordonnancements.

- Deux ordonnancements S1 et S2 sont équivalents s'ils incluent exactement les mêmes transactions et l'effet final des deux est le même indépendamment de l'état initial de la base.
- Soit $T_i T_j$ les transactions participant à deux ordonnancements S1 et S2
 1. Pour tout LIRE(X) fait par T_i dans S1, si la valeur lue est celle modifiée (en dernier) par ECRIRE(X) de T_j dans S1 alors la même opération LIRE(X) de T_i dans S2 doit lire la valeur écrite par ECRIRE(X) de T_j dans S2.
 2. Pour toute entité X sur laquelle il y a un dernier ECRIRE(X) exécutée par T_i dans S1 alors la même opération ECRIRE(X) par T_i doit être la dernière dans S2.

Supposez les 2 ordonnancements suivants et dire de quel type sont-ils.

S1 = L1(A);E1(A);L1(B);E1(B);L2(A); E2(A); L2(B); E2(B)

S2= L1(A);E1(A); L2(A);E2(A); L1(B);E1(B); L2(B);E2(B)

5. Ces ordonnancements sont-ils équivalents (argumenter) ?

Ordonnement conflit-sérialisable

Un ordonnancement est conflit-sérialisable si il est conflit-équivalent à un ordonnancement séquentiel. Les conflits entre deux opérations de transactions distinctes arrivent lors qu'elles s'intéressent aux mêmes données et une des opérations est une écriture :

T1	T2	Lecture T1(X)	Ecriture T1(X)
Lecture T2 (X)			
Ecriture T2(X)			

- Conditions
 - Deux ordonnancements sont conflit-équivalent si tous les deux peuvent être inter-changés l'un par l'autre par la séquence des échanges d'opérations NON conflictuelles adjacentes
 - Deux opérations de la même transaction ne peuvent pas être inter-changées

6. Cet ordonnancement est-il conflit-sérialisable ?

$S2 = L1(A); E1(A); L2(A); \underline{E2(A)}; \underline{L1(B)}; E1(B); L2(B); E2(B)$

Graphe de dépendances

Les graphes de dépendance permettent de vérifier si un ordonnancement est conflit-sérialisable.

- Pour chaque ordonnancement S, on définit une relation de précédence entre les transactions qui indique la précédence dans le temps, d'actions conflictuelles sur un même objet
- Si le graphe est acyclique l'ordonnancement est conflit – sérialisable
- Pour construire le graphe :
 - les sommets sont les transactions
 - on a une arête de T_i vers T_j si dans l'ordonnancement un objet x est :
 - Écrit par T_i puis Écrit par T_j ou
 - Écrit par T_i puis lu par T_j ou
 - Lu par T_i puis Écrit par T_j
 - et aucune écriture (faite par une autre transaction) n'intervient entre les deux opérations

Graphe conflit-sérialisable

Graphe non-conflit sérialisable

Les conflits entre les transactions peuvent entraîner quelques problèmes (on les appelle phénomènes).

7. Expliquer à quel moment peut survenir un problème de :
- a. Perte de mise à jour
 - b. Lecture impropre
 - c. Lecture non-reproductible

D'après les problèmes identifiés 3 niveaux de cohérence existent.

- Niveau 1 : Pas de perte de m à j
Toute T ne doit pas écrire sur un objet dont la valeur a été modifiée par une autre T' qui n'a pas atteint son point de confirmation.
- Niveau 2 : Ni pertes de m à j ni lectures sales
+ Toute transaction T ne doit pas lire des valeurs non confirmées, manipulées par d'autres transactions.
- Niveau 3 : Ni perte de m à j ni lectures sales et non reproductibles
+ Aucune transaction ne doit modifier une valeur lue par une autre transaction avant que cette dernière ne soit terminée.

Les protocoles de contrôle de concurrence permettent d'assurer plus ou moins ces niveaux de cohérence.

Protocole d'exclusion mutuelle

Procéder à une allocation exclusive des objets que chaque transaction manipule.

Principe : Avant d'effectuer une opération sur x (lire ou écrire),

- Acquérir le contrôle exclusif : verrouiller(x)
- Abandonner le contrôle : libérer(x)

Règle 1 : Aucune transaction ne peut effectuer une mise à jour ou une lecture d'un objet si elle n'en a pas acquis au préalable le contrôle exclusif par verrouiller

Règle 2 : Si une transaction T2 ne peut acquérir le contrôle exclusif d'un objet X, parce que X est utilisé par une autre transaction T1, T2 doit attendre jusqu'à ce que X soit libéré par T1.

8. Quel niveau d'isolation et quel type d'ordonnancement permet d'assurer ce protocole ?
9. Quels sont les inconvénients de ce protocole ?

Protocole de verrouillage à 2 phases (2PL)

Pour optimiser l'accès concurrent deux types de verrous :

- PARTAGEABLES en LECTURE : V- partagé (A)
- EXCLUSIF en ECRITURE : V- exclusif (A)

Principe :

- Transaction *t bien formée* si :
 - i. Avant de LIRE x elle a au moins un V-partagé (x)
 - ii. Avant d'ECRIRE x elle a un V-exclusif (x)
 - iii. Aucun objet ne reste verrouillé après FIN TRANSACTION T
- Transaction à deux phases (2 Phase Locking) :
 - Règle I : Des transactions différentes ne peuvent posséder simultanément des verrous en conflit
 - Règle II : Toute transaction qui libère un verrou ne peut plus en acquérir d'autres

Théorème : Si dans un ordonnancement toutes les transactions sont à deux phases et bien formées alors tout ordonnancement est équivalent à un ordonnancement séquentiel (qui est donc cohérent)

10. Quel niveau d'isolation et quel type d'ordonnancement permet d'assurer ce protocole ?
11. Quels sont les inconvénients de ce protocole ?

Protocole de verrouillage à deux phases strict (S-2PL)

- Comme 2PL mais la règle II de 2PL est remplacée par :
 - Tous les verrous sont libérés à la fin de la transaction (commit/abort)

12. Quel niveau d'isolation et quel type d'ordonnancement permet d'assurer ce protocole ?
13. Quels sont les inconvénients de ce protocole ?

Donner les définitions suivantes.

14. Ordonnancement séquentiel
15. Ordonnancement sérialisable
16. Ordonnancement conflit-sérialisable

17. Ordonnement recouvrable

Interblocages

Le verrouillage peut entraîner des interblocages lorsque 2 transactions attendent leurs verrous mutuellement.

Les cycles dans le graphe Qui Attend Quoi permettent de détecter les interblocages.

Graphe Qui Attend Quoi (ou Qui)

- Nœuds = Transactions (et Objets)
- Arêtes = de l'objet vers T_i si T_i a verrouillé l'objet, de T_i vers l'objet si T_i attend

18. Quoi doit être fait lorsqu'un interblocage est détecté ?

Graphe de dépendances

Le graphe de dépendance permet de vérifier la sérialisabilité d'un ordonnancement.

Construction du graphe :

- On a une arête de T_i vers T_j si T_i fait $VL(x)$ ou $VE(x)$ et si T_j est la transaction suivante à verrouiller x en écriture $VE(x)$
- Si T_i relâche un verrou exclusif sur x alors il y a une arête de T_i vers toutes les transactions T_m qui posent un verrou en lecture $VL(x)$ sur x jusqu'à ce qu'une transaction pose un verrou en écriture sur x .

S'il n'a pas de cycle on peut produire un ordonnancement séquentiel équivalent (l'ordonnancement est sérialisable)

Feuilles de Travaux Dirigés

- TD1 : Rappels SQL
- TD2 : Dépendances fonctionnelles et formes normales
- TD3 : Normalisation
- TD4 : PL/SQL Blocs
- TD4 bis : PL/SQL Fonctions et SQL dynamique
- TD5 : Triggers et vues
- TD6 : Evaluation de requêtes - Optimisation
- TD7 : Indexes
- TD8 : Transactions

TD1 : Rappels SQL

Exercice 1

Soit le schéma relationnel suivant permettant de mémoriser les informations concernant la gestion des projets d'études de construction de maisons individuelles réalisés par une entreprise. Les clés relationnelles sont soulignées.

Les employés sont identifiés par leur matricule (*matreempl*) et caractérisés par leur nom (*nom*), leur prénom (*prenom*), leur adresse (*adresse*), leur qualification professionnelle (*qualif*) et le numéro de service auquel ils sont affectés (*servaffect*). Un employé n'est affecté qu'à un seul service.

Les services sont identifiés par leur numéro de service (*nuserv*) et caractérisés par leur libelle (*libserv*) et le matricule de l'employé directeur du service (*matrdir*). Un service n'a qu'un seul directeur et un employé ne peut être directeur que d'un seul service.

Les projets sont identifiés par leur numéro de projet (*nuproj*) et caractérisés par leur libelle (*libproj*), le matricule de l'employé chef du projet (*matrchef*) et la date de début de réalisation du projet (*datedebut*) sous la forme JJ-MMM-AAAA où JJ représente le numéro du jour du mois, MMM le mois en abrégé et AAAA l'année. Un projet n'a qu'un seul chef et un employé peut être chef de plusieurs projets.

La relation *compoproj* donne pour chaque projet les lots qui le composent. Pour chaque lot d'un projet, on dispose du temps total en heures prévu pour sa réalisation (*tpstotprevh*). Chaque jour durant lequel on travaille sur le lot d'un projet, on cumule le temps effectivement consacré à ce lot afin que l'on puisse disposer du temps en cours réellement effectué à l'aide des deux attributs *tpsencoursh* et *tpsencoursm*. Le premier exprime les heures et le second les minutes.

La relation *travail* fournit tous les jours le temps de travail effectif de chaque employé sur chacun des lots. L'attribut *tpsjourm* correspond au temps de travail effectué par l'employé sur le lot de la maison concernée le jour de la date *datejour*. Ce temps de travail est exprimé en minutes. Par exemple, un temps de travail de 2h25 est exprimé par 145 mm.

Employe (matreempl char(6), nom varchar2(25), prenom varchar2(25), adresse varchar2(50), qualif varchar2(25), servaffect char(2))

Service (nuserv char(2), libserv varchar2(25), matrdir char(6))

Projet (nuproj char(3), libproj varchar2(25), matrchef char(6), datedebut date)

Compoproj (nuproj char(3), nulot char(3), tpstotprevh number(4), tpsencoursh number(4), tpsencoursm number(2))

Travail (nuproj char(3), nulot char(3), datejour date, matreempl char(6), tpsjourm number(3))

Q1 : Donner les requêtes SQL permettant de créer les relations ci-dessus. On écrira les contraintes de domaines ainsi que celles de clés primaires et étrangères.

Q2 : Pour les recherches suivantes, proposer l'expression relationnelle et l'arbre algébrique permettant d'obtenir le résultat, puis donner son expression en SQL. Vous devrez éliminer les n-uplets dupliqués des résultats de vos requêtes lorsque c'est nécessaire.

1. Liste des employés comportant pour chacun le matricule, le nom, le prénom et le service d'affectation.

2. Liste des projets ayant démarré entre le premier janvier et le premier juillet 2018.
3. Numéro de projets et de lots sur lesquels a travaillé l'employé de matricule 06PR14. Reprendre la question en affichant aussi le libellé du projet et sa date de début. Reprendre la question en affichant aussi le nom et le prénom du chef de projet ainsi que son service d'affectation.
4. Numéros de projet et de lot sur lesquels l'employé de matricule 06PR14 a travaillé aujourd'hui, avec pour chaque lot, le temps qui lui a été globalement consacré jusque là.

Q3 : Proposer une requête SQL pour les recherches suivantes. en utilisant les relations ci-dessus. Vous devrez éliminer les n-uplets dupliqués des résultats de vos requêtes lorsque c'est nécessaire.

1. Liste des employés qui ont travaillé sur le projet de numéro 222 et celui de numéro 333 le 1er mars 2012. Afficher le matricule de l'employé, ses nom, prénom et service d'affectation.
2. Liste des employés qui travaillent sur tous les projets qui ont débuté au 1er janvier 2013. Afficher seulement le matricule des employés.
3. On cherche à établir la liste des employés qui ont travaillé sur au moins deux lots différents d'un même projet, un même jour. On affichera le matricule, le nom et le prénom des employés, le jour, le numéro et le libellé du projet, le nombre de lots.
4. Liste des employés pour lesquels le temps de travail effectué jusqu'à présent pour un projet dépasse 50% du temps en cours passé au total sur le projet (donné par les attributs *tpsencoursh* et *tpsencoursm*). Afficher le matricule de l'employé et le numéro de projet.
5. Afficher chaque service (i.e. son numéro et son libellé) et sa contribution au projet de numéro 222 en termes de temps (via chaque membre du service). Chaque service doit apparaître, même s'il n'a pas contribué. Le temps sera arrondi à l'heure inférieure.

TD2

Dépendances fonctionnelles et formes normales

Exercice 1

$R(\text{id_matiere}, \text{id_groupe}, \text{id_prof_ens}, \text{id_corresp_ele}, \text{id_prof_resp}, \text{no_creneau_h}, \text{jour}, \text{salle})$ est un schéma de relation qui mémorise l'emploi du temps simplifié hebdomadaire d'un collège avec les dépendances fonctionnelles suivantes :

- (1) $\text{id_matiere}, \text{id_groupe} \rightarrow \text{id_prof_ens}$
- (2) $\text{id_groupe} \rightarrow \text{id_prof_resp}$
- (3) $\text{id_prof_resp} \rightarrow \text{id_groupe}$
- (4) $\text{id_groupe}, \text{id_prof_ens} \rightarrow \text{id_matiere}$
- (5) $\text{id_matiere}, \text{id_groupe} \rightarrow \text{id_corresp_ele}$
- (6) $\text{no_creneau_h}, \text{jour}, \text{salle} \rightarrow \text{id_matiere}$
- (7) $\text{no_creneau_h}, \text{jour}, \text{salle} \rightarrow \text{id_groupe}$

Q1. Cet ensemble de dépendances permet-il :

1. Qu'un prof soit responsable de plusieurs groupes ?
2. Qu'un prof puisse enseigner des matières différentes ?

Q2. Indiquer si l'instance de relation suivante satisfait toutes les dépendances fonctionnelles.

id_matiere	id_groupe	id_prof_ens	id_prof_resp	id_corresp_ele	no_creneau_h	jour	salle
math	6A	dupont	lerouge	max6A	8	2	112
math	4C	martin	dupont	art4C	8	3	112
français	6A	dupont	lerouge	max6A	10	4	111
français	5E	martin	lerouge	isa5E	10	4	111

Q3. Calculer $\{\text{id_matiere}, \text{id_groupe}\}^+$, puis indiquer les clés de ce schéma.

Q4. Quelles seraient les clés dans un schéma $R'(\text{id_matiere}, \text{id_groupe}, \text{id_prof_ens}, \text{id_corresp_ele}, \text{id_prof_resp})$ muni des dépendances (1) (2) (3) (4) et (5).

Exercice 2

Une entreprise veut maintenir une base de données concernant son approvisionnement. L'ensemble des attributs identifiés est noté U . Il contient les attributs suivants : **codef** : le code du fournisseur ; **nomf** : le nom du fournisseur ; **codep** : le code du produit ; **designation** : la désignation du produit ; **prix_unit**, **tva** : le prix unitaire (respectivement le taux de tva) auquel un fournisseur fournit un produit ; **jourlivr**, **qte** : jour de livraison d'un produit et quantité livrée ; **adressef** : adresse du

fournisseur.

Soit le schéma de relation $R(U)$ muni de l'ensemble DF des dépendances fonctionnelles suivantes :

- (1) $codef \rightarrow nomf$
- (2) $codep \rightarrow designation$
- (3) $codep, codef \rightarrow prix_unit$
- (4) $codef \rightarrow adressef$
- (5) $codep, codef \rightarrow tva$
- (6) $codep, codef, jourlivr \rightarrow qte$

Q1. Répondre (simplement) par OUI ou NON. Cet ensemble de dépendances permet-il :

1. qu'un même fournisseur applique à l'entreprise un taux de tva différent pour deux produits différents ?
2. que deux produits de code différent aient la même désignation ?
3. que deux fournisseurs livrent le même produit, le même jour en quantités différentes ?

Q2. Illustrer l'anomalie de redondance d'informations sur une instance bien choisie de $R(U)$, DF et expliquer. Il n'est pas nécessaire de spécifier une valeur pour tous les attributs.

Q3. Donner toutes les clés de $R(U)$ muni de l'ensemble DF . *Justifier par rapport à la définition.*

Q4. Montrer, en utilisant la définition, que $R(U)$, DF n'est pas en 3e forme normale (3FN).

Exercice 3

Soient le schéma $R(A, B, C, D, E, F, G, H, I, J)$ et l'ensemble des dépendances fonctionnelles de R suivantes $DF = \{A, B, D \rightarrow E ; A, B \rightarrow G ; B \rightarrow F ; C \rightarrow J ; C, J \rightarrow I ; G \rightarrow H\}$

1. Calculer les fermetures : $\{A, C, G\}^+$, $\{A, B, C\}^+$.
2. Donner toutes les clés de R muni de DF .
3. Ce schéma est-il en 3 FN ?

Exercice 4

On considère un laboratoire qui réalise le suivi de différents végétaux. Chaque plant est identifié par un numéro (noplant). Chaque mois il est installé dans un emplacement spécifique qui ne contient que lui. Cet emplacement, pour lequel on désigne un responsable chaque jour, ne contient qu'un seul type de plante. Chaque plant est suivi selon un certain protocole qui est appliqué avec un kit spécifique. Ce dernier peut être ajusté en fonction des données climatiques disponibles. Les dépendances fonctionnelles identifiées sont les suivantes :

- (1) mois, noplant \rightarrow emplacement
- (2) emplacement, mois \rightarrow noplant
- (3) jour, emplacement \rightarrow responsable
- (4) emplacement \rightarrow typeplante

(5) noplant \rightarrow idprotocole

(6) idprotocole \rightarrow nokit

(7) jour \rightarrow temperaturej, soleilj, ventj

(8) idprotocole, temperaturej, soleilj, ventj \rightarrow qtéeau, qtéengrais, typeengrais

(9) responsable, qtéeau \rightarrow modeadministration

Q1. On considère le schéma de relation $R(\text{mois}, \text{noplant}, \text{emplacement})$ muni des dépendances (1) et (2). Donner toutes les clés du schéma et indiquer en quelle forme normale il est.

Q2. On considère maintenant le schéma défini par l'ensemble des dépendances exprimées ci-dessus et l'ensemble des attributs qu'elles utilisent.

1. Calculer $\{\text{jour}, \text{noplant}\}^+$
2. Quelles sont toutes les clés de ce schéma ? Justifier.
3. Montrer que ce schéma n'est pas en 3FN. Illustrer le phénomène de redondance sur un exemple d'instance bien choisi.

Exercice 5

1. Soit $R(A, B, C)$ et $DF = \{B \rightarrow C\}$. Si A est une clé, R peut-elle être en FNBCCK ?
2. Soit $R(A, B, C, D, E)$ et $DF = \{A \rightarrow B ; B, C \rightarrow E ; E, D \rightarrow A\}$. La relation est-elle en FNBCCK ?
3. Soit $R(A, B, C, D)$ et $DF = \{\}$. La relation est-elle en FNBCCK ?

TD3 : Normalisation

Exercice 1

On considère un schéma de relation $R(\text{nopers}, \text{nomevt}, \text{datevt}, \text{salle})$ relatif à la gestion de salles où se déroulent des événements culturels.

- nopers : représente le numéro d'une personne gérant une salle
- nomevt : est le nom d'un événement
- datevt : indique le jour où a lieu un événement.
- salle : représente le nom d'une salle où a lieu un événement.

Les dépendances fonctionnelles sont les suivantes :

- (1) $\text{nopers} \rightarrow \text{salle}$
- (2) $\text{salle}, \text{datevt} \rightarrow \text{nomevt}$
- (3) $\text{nomevt}, \text{datevt} \rightarrow \text{salle}$

Q1. Donner les clés du schéma $R(\text{nopers}, \text{nomevt}, \text{datevt}, \text{salle})$, $DF = \{(1), (2), (3)\}$. Justifier. Le schéma est-il en 3FN ? Justifier.

Q2. La décomposition suivante est proposée :

$R_1(\text{nopers}, \text{salle})$ $DF_1 = \{(1)\}$; $R_2(\text{salle}, \text{datevt}, \text{nomevt})$ $DF_2 = \{(2), (3)\}$.

Montrer, à l'aide d'une instance de R bien choisie, que cette décomposition n'est pas spi (sans perte d'information). Préciser quelle information est perdue.

Q3. Proposer une décomposition spi et spdf où toutes les relations sont en 3FN. Expliquer.

Exercice 2

Soit $R(A, B, C, D)$. Pour chaque ensemble de df suivant : (a) Donner les clés de R et (b) commenter la qualité de la décomposition proposée (spi, spdf).

$DF_1 = \{B \rightarrow C ; D \rightarrow A\}$; décomposition : $R_1(B, C)$ et $R_2(A, D)$.

$DF_2 = \{A, B \rightarrow C ; C \rightarrow A ; C \rightarrow D\}$; décomposition : $R_1(A, C, D)$ et $R_2(B, C)$.

$DF_3 = \{A \rightarrow B ; B \rightarrow C ; C \rightarrow D\}$; décomposition : $R_1(A, B)$, $R_2(A, D)$ et $R_3(C, D)$.

Exercice 3

On reprend la base de données pour le suivi des végétaux de la feuille de TD2. On note U l'ensemble de tous les attributs utilisés dans les dépendances fonctionnelles identifiées suivantes:

- (1) $\text{mois}, \text{noplant} \rightarrow \text{emplacement}$
- (2) $\text{emplacement}, \text{mois} \rightarrow \text{noplant}$
- (3) $\text{jour}, \text{emplacement} \rightarrow \text{responsable}$
- (4) $\text{emplacement} \rightarrow \text{typeplante}$
- (5) $\text{noplant} \rightarrow \text{idprotocole}$
- (6) $\text{idprotocole} \rightarrow \text{nokit}$
- (7) $\text{jour} \rightarrow \text{temperaturej}, \text{soleilj}, \text{ventj}$
- (8) $\text{idprotocole}, \text{temperaturej}, \text{soleilj}, \text{ventj} \rightarrow \text{qtéeau}, \text{qtéengrais}, \text{typeengrais}$
- (9) $\text{responsable}, \text{qtéeau} \rightarrow \text{modeadministration}$

Q1. Utiliser l'algorithme de Bernstein pour proposer une décomposition spi et spdf de la relation $R(U)$ muni de l'ensemble des dépendances.

Exercice 4

On considère les attributs suivants : TITRE (titre d'un film), DATE (date de sortie du film), REAL (réalisateur), NOMFEST (nom du festival où le film a concouru), PRIX (prix obtenu par le film au festival), NOMACT (nom d'un acteur primé au festival pour ce film), NAT (nationalité de l'acteur).

Soit $R(\text{TITRE, DATE, REAL, NOMFEST, PRIX, NOMACT, NAT})$ munie des df :

- TITRE \rightarrow DATE, REAL (1),
- TITRE, NOMFEST \rightarrow PRIX (2),
- NOMACT \rightarrow NAT (3).

Q1. Montrer que R n'est pas en 3FN et décomposer R en un ensemble de schémas 3FN de manière spi et spdf en utilisant **(i)** l'algorithme de synthèse de Bernstein **(ii)** l'algorithme de décomposition.

Exercice 5

Soit le schéma de relation R qui représente le personnel d'une entreprise et son affectation :

$R(\text{noe number, nome varchar(10), noms varchar(15), chef number, nobu number(2), bat char(3), projet char(5), exp char(3)})$

où les attributs désignent respectivement le numéro de l'employé, le nom de l'employé, le nom du service auquel il est affecté, le numéro d'employé du chef de ce service, le numéro de bureau de l'employé, le bâtiment dans lequel travaille l'employé, un projet auquel il participe, une expérience du projet sur laquelle il doit travailler. Notons qu'un projet peut être constitué de plusieurs expériences. Il peut y avoir des expériences de projets différents dans un même bâtiment.

Il ressort de l'analyse les dépendances fonctionnelles suivantes :

- noe \rightarrow nome, nobur (1)
- noms \rightarrow chef (2)
- nobur \rightarrow bat (3)
- noe, nobur \rightarrow noms (4)
- bat, projet \rightarrow exp (5)

La dépendance (5) signifie que la seule connaissance du projet et du bâtiment dans lequel travaille un employé permet de déterminer à quelle expérience il travaille.

Q1. La relation $R(DF)$ est-elle en FNBCCK ? Justifier. Illustrer le phénomène de redondance sur une instance judicieusement choisie de la relation.

Q2. Soit la décomposition suivante de R :

$R_1(\text{noe, nome, nobur, noms})$

$R_2(\text{noms, chef})$

$R_3(\text{nobur, bat})$

$R_4(\text{bat, projet, exp})$

1. Montrer que la décomposition est sans perte de dépendances fonctionnelles (spdf) et avec perte d'information (non spi).
2. Rappeler la *définition* d'une décomposition spi et illustrer la perte d'information sur une instance bien choisie. Indiquer par une phrase claire quelle information a été perdue.

Q3. Utiliser l'algorithme de normalisation par décomposition pour proposer une décomposition de $R(DF)$ dont vous indiquerez les propriétés (spi ? spdf ? forme normale des relations obtenues ?).

TD4 PL/SQL - Blocs

Exercice 1

On suppose les créations de tables suivantes :

```
create table situation(
mois number,
noplant number,
emplact varchar2(30),
primary key(mois, noplant));

create table plantes(
noplant number,
type varchar2(30),
primary key(noplant));
```

Q1. Ecrire un bloc PL/SQL qui affiche le nombre d'emplacement distincts utilisés pour une plante dont le numéro est passé en paramètre. Un message d'erreur doit s'afficher si le numéro de plante donné n'existe pas.

Q2.

1. Ecrire un bloc PL/SQL qui affiche pour chaque numéro de plante se trouvant dans la table *situation*, le nombre d'emplacements différents utilisés.
2. Modifier le bloc pour qu'il écrive les résultats dans une table de nom *resultats*.

TD4 : PL/SQL – Fonctions et SQL dynamique

Le langage PL/SQL permet l'automatisation certains processus accédant une base de données. Dans ce TD nous allons utiliser ce qui est appelé du SQL dynamique. La commande EXECUTE IMMEDIATE parse et exécute du code SQL dynamique. Le paramètre de cette commande est une chaîne de caractères contenant les commandes SQL ou PL/SQL à exécuter.

Voici un block PL/SQL contenant plusieurs commandes SQL dynamiques.

```
DECLARE
    sql_stmt      VARCHAR2(200);
    plsql_block  VARCHAR2(500);
    emp_id       NUMBER(4) := 7566;
    salary       NUMBER(7,2);
    dept_id     NUMBER(2) := 50;
    dept_name   VARCHAR2(14) := 'PERSONNEL';
    location    VARCHAR2(13) := 'DALLAS';
    emp_rec     emp%ROWTYPE;
BEGIN
    EXECUTE IMMEDIATE 'CREATE TABLE bonus (id NUMBER, amt NUMBER)';
    sql_stmt := 'INSERT INTO dept VALUES (:1, :2, :3)';
    EXECUTE IMMEDIATE sql_stmt USING dept_id, dept_name, location;
    sql_stmt := 'SELECT * FROM emp WHERE empno = :id';
    EXECUTE IMMEDIATE sql_stmt INTO emp_rec USING emp_id;
    plsql_block := 'BEGIN emp_pkg.raise_salary(:id, :amt); END;';
    EXECUTE IMMEDIATE plsql_block USING 7788, 500;
    sql_stmt := 'UPDATE emp SET sal = 2000 WHERE empno = :1
        RETURNING sal INTO :2';
    EXECUTE IMMEDIATE sql_stmt USING emp_id RETURNING INTO salary;
    EXECUTE IMMEDIATE 'DELETE FROM dept WHERE deptno = :num'
        USING dept_id;
    EXECUTE IMMEDIATE 'ALTER SESSION SET SQL_TRACE TRUE';
END;
/
```

La création de vos comptes utilisateur en SQL peut être fait avec le code SQL suivant :

```
CREATE USER L3_10 IDENTIFIED BY L3
DEFAULT TABLESPACE L3Info
TEMPORARY TABLESPACE L3Info_TEMP
QUOTA 1M ON L3info;
```


La destruction des vos comptes utilisateur entraînant la destruction de tous les objets créés sous ces comptes (la destruction du schéma) est fait par la commande :

```
DROP USER L3_10 CASCADE;
```

Les droits système attribués à vos comptes sont donnés par la commande suivante :

```
GRANT CONNECT, RESOURCE TO  
L3_1, L3_2, L3_3, L3_4, L3_5, L3_6, L3_7, L3_8, L3_9, L3_10, etc. ;
```

Q1. Faire 3 procédures PL/SQL contenant du SQL dynamique pour automatiser :

- a)** la création de comptes. La procédure doit prendre en paramètre le préfixe des comptes, le nombre de comptes à créer, les tablespaces (défaut et temporaire) et le quota.
- b)** la suppression des comptes. La procédure devra prendre en paramètre le préfixe des comptes ainsi que le nombre de comptes. Prenez soin à lever une exception si les comptes n'existent pas.
- c)** l'attribution de droits système aux comptes. La procédure devra prendre en paramètre les droits à attribuer (une chaîne de caractères), le préfixe des comptes et le nombre de comptes.

Q2. Prenez le temps de réfléchir à la proposition de processus pouvant se réaliser avec de procédures et fonctions PL/SQL. Prenez par exemple les différentes relations vues dans les TD précédents. Développer une discussion autour du sujet avec vos collègues et le chargé de TD.

TD Triggers et vues

L'objectif du TD est d'écrire de triggers et créer de vues en utilisant la base de données suivante.

Description de la base

Une association étudiante nantaise regroupe pour ses membres des offres de promotions sur des séjours de groupe. Le schéma de base de données comprend les tables suivantes.

Promos (idpromo varchar2(5), nbplaces number, *idlog* varchar2(5), cout number, deb timestamp, fin timestamp)

Mémoire les promotions obtenues pour un nombre de places donné, dans un logement précis, avec prix du séjour par personne, hors frais de transport, la date de début et de fin du séjour. L'attribut *idlog* référence la table Logts.

Logts (idlog varchar2(5), type varchar2(20), nom varchar2(30), adresse varchar2(50), capacité number, idzone varchar2(15))

Mémoire les différents logements connus de l'association. Pour chaque logement, sont indiqués son type (hôtel, gîte, auberge de jeunesse), son nom, son adresse, sa capacité maximale (nombre de places maximal) et la zone dans laquelle il se trouve. L'attribut *idzone* fait référence à la table Zones.

Zones (idzone varchar2(15), dpt number, type varchar2(20), caract varchar2(20), dist number, couTr number)

L'association mémorise un certain nombre de caractéristiques des zones. Pour identifiant de zone (par exemple, Brière, Vercors), elle mémorise le numéro de département concerné, son type (côté atlantique, montagne,...) et une caractéristique particulière (par exemple Parc régional). La table mémorise aussi une évaluation de la distance en kilomètres qui sépare Nantes de la zone, ainsi qu'une évaluation du coût du transport.

Reservs (idresa varchar2(5), idpromo varchar2(5), noetu varchar2(8), nb number)

Mémoire les réservations pour une promotion, faite par un étudiant. L'attribut *nb* indique le nombre de places réservées. *idPromo* référence la table Promos.

Membres (noetu varchar2(8), nome varchar2(20), prenom varchar2(20), diplome varchar2(10))

Pour chaque étudiant identifié par son numéro, mémorise ses nom et prénom ainsi que le diplôme principal qu'il prépare.

Histo_reservs (username varchar2(20), date_modif timestamp, operation varchar2(20), idresa varchar2(5), idpromo varchar2(5), noetu varchar2(8), nb number)

Cette table maintient un historique des réservations pour savoir quel utilisateur a fait à quel moment, quelle opération (insert, delete ou update) sur la table Reservs.

Les triggers

Q1. Proposez deux solutions pour assurer que dans la relation promotion, on empêche l'insertion ou la modification d'un tuple si la date de début de séjour est supérieure à celle de fin de séjour.

Q2. Ecrire un déclencheur qui, à l'insertion d'un tuple, remplit automatiquement le champ *couTr* (coût théorique du transport) de la table Zones. Ce coût est calculé à partir de la distance indiquée et d'un coefficient quelconque représentant le coût en kilomètres, il doit être fait dans une fonction qui

sera appelée à partir du déclencheur.

Q3. Ecrire un déclencheur qui provoque une erreur à l'insertion d'un tuple dans la table reservs si le nombre de places réservées par la personne additionné au nombre de places déjà réservées sur cette promo dépasse le nombre de places (nbplaces) de la promo.

Q4. Ecrire un déclencheur qui provoque une erreur à l'insertion d'un tuple dans la table Promos si le nombre de places du logement dépasse la capacité du logement défini dans la table Logts.

Q5. Créer un déclencheur qui mémorise dans la table histo_reservs quel utilisateur a fait, à quelle heure, une modification (insert, delete ou update) et quelles nouvelles valeurs d'attributs il a insérées dans la table Reservs.

Q6. Proposer d'autres déclencheurs (au moins 3)

Vues

Q1. Créer la vue Zonnes44 à partir de la table Zones pour avoir dans cette vue uniquement les zones du département 44. Les attributs de la vue seront : idzone, type, caract, dist.

Cette vue est-elle modifiable ?

Si vous pensez qu'elle n'est pas modifiable, comment la rendre modifiable ?

Q2. Créer la vue StatusPromos qui permet d'avoir par promotion, le nombre de places de la promotion et le nombre de places disponibles. Les attributs de la vue seront idpromo et placesDispo.

Cette vue est-elle modifiable ?

Si vous pensez qu'elle n'est pas modifiable, comment la rendre modifiable ?

Evaluation de requêtes - Optimisation

Exercice 1

Soit la requête :

```
select titre
from livres
where annee = 2009 or (annee = 2009 and id_auteur = 'Gard');
```

posée sur la relation de schéma suivant :

```
livres (cote, isbn, titre, id_auteur, annee)
```

On suppose qu'il existe un index B+ sur l'attribut `annee` et que l'optimiseur connaît deux implantations de l'opérateur de sélection : un algorithme de parcours séquentiel et un algorithme utilisant un index B+, et une seule implantation de l'opérateur de projection, sous sa forme la plus simple (algorithme sans élimination de doublons ni tri).

Q1. Expliquer ce que produirait la phase « analyse/traduction » de l'évaluation de cette requête.

Q2. Donner tous les plans possibles pour cette requête.

Q3. Quel plan semble le « meilleur » intuitivement ? Évaluez sa complexité en fonction du nombre n de pages disque utilisées pour stocker la relation `livres`.

Exercice 2

On considère les relations :

```
livres (cote, isbn, titre, id_auteur, annee)
auteurs (id_auteur, nom, prenom, labo, email)
```

et la requête :

```
R1 : select * from livres natural join auteurs
where annee = 2009 and labo = 'LINA';
```

On suppose que :

- les relations `livres` et `auteurs` occupent respectivement n et m pages sur disque.
- l'optimiseur connaît trois implantations de la jointure : sous la forme de 2 boucles imbriquées ; sous la forme de 2 boucles imbriquées utilisant un index sur la table intérieure ; sous la forme d'un tri-fusion.
- qu'il n'y a qu'une seule implantation de la sélection.

Q1. Donner tous les plans possibles pour cette requête, en rappelant l'ordre de grandeur du nombre d'accès disque pour chacun des opérateurs élémentaires.

Q2. Quelles informations manque-t-il pour pouvoir calculer de manière plus précise le nombre total de pages accédées ?

Exercice 3

(d'après TD, Université Pierre et Marie Curie) On considère de nouveau les relations :

livres (cote, isbn, titre, id_auteur, annee)

auteurs (id_auteur, nom, prenom, labo, email)

où id_auteur est une clé étrangère dans la relation livres. Les hypothèses sont :

- le nombre de tuples de la relation livres est $T_l = 200000$; le nombre de tuples de la relation auteurs est $T_a = 10000$.
- la relation livres mémorise des tuples dont la valeur de l'attribut annee est comprise entre 2000 et 2010.
- la taille en octets des attributs de la relation livres est de 2, 3, 4, 2 et 1 octets respectivement pour les attributs cote, isbn, titre, id_auteur, annee.
- la taille en octets des attributs de la relation auteurs est de 2, 4, 4, 3 et 4 octets respectivement pour les attributs id_auteur, nom, prenom, labo, email.
- les valeurs des attributs sont indépendantes et la distribution de ces valeurs dans leur domaine est uniforme.
- le nombre de valeurs distinctes des attributs de la relation livres est noté nd1, nd2, nd3, nd4, nd5 respectivement. Il est noté nd'1, nd'2, nd'3, nd'4, nd'5 pour les attributs de auteurs.

Q1. Proposer des valeurs pour les paramètres nd1, nd2, nd3, nd4, nd5, nd'1, nd'2, nd'3, nd'4 et nd'5.

Q2. Pour les requêtes suivantes, donner la cardinalité (nombre de tuples) et la taille en octets du résultat :

1. `select id_auteur, nom, prenom from auteurs ;`
2. `select * from auteurs where labo = 'LINA';`
3. `select * from auteurs a, livres l where a.id_auteur = l.id_auteur;`
4. `select * from livres where annee >= 2007 ;`

Q3. Quelles opérations produisent un résultat de taille supérieure à la taille de leurs opérandes ?

Q4. On considère les deux formes suivantes de la même recherche :

- `select a.nom from auteurs a, livres l where a.id_auteur = l.id_auteur and annee = 2009;`
- `select a.nom from auteurs a, livres l where annee = 2009 and a.id_auteur = l.id_auteur;`

a. Donner leur arbre syntaxique en supposant que les opérateurs disponibles sont la sélection, la projection, la jointure, l'union et la différence et que les arbres sont construits en respectant l'ordre des prédicats de la clause `where`.

b. Calculer le volume en octet de l'information manipulée pendant leur exécution si celle-ci suit l'arbre syntaxique. Conclure.

Q5. Une des heuristiques utilisée pour la recherche d'un plan d'exécution consiste à effectuer le plus tôt possible les sélections puis les projections, avant les jointures. Pour cela, donner les règles de transformation que l'on peut utiliser.

TD : INDEXES

Exercice 1

Répondez aux questions suivantes, et donnez un exemple.

- 1) Qu'est-ce qu'un index primaire ?
- 2) Qu'est-ce qu'un index secondaire ?
- 3) Qu'est-ce qu'un index dense ?
- 4) Qu'est-ce qu'un index clairsemé ?
- 5) Qu'est-ce qu'un index multiniveaux ?

Répondez aux questions suivantes selon la définition des indexes B+ donnée en cours et selon le livre de Bases de Données de Garcia-Molina, Ullman et Widom.

Supposez un arbre B+ d'uniquement 2 niveaux et avec des blocs (nœuds) pouvant stocker n clés.

- 1) Combien de pointeurs peut contenir chaque bloc ?
- 2) Combien de pointeurs *au minimum* doivent être utilisés dans un nœud:
 - a) racine ?
 - b) intermédiaire ?
 - c) feuille ?
- 3) Combien de clés en totale peut indexer cet arbre ?
- 4) Combien de clés au minimum doit stocker cet arbre ?
- 5) Quel est l'intérêt que la structure d'un index B+ soit équilibrée ?
- 6) Quel est l'intérêt que chaque nœud ait un nombre minimal de pointeurs ?
- 7) Supposez un block intermédiaire avec $n=3$ et clés = 14, 52 et 78. Quel est le rang des clés k sur lequel va pointer chaque pointeur de ce bloc ?

Exercice 2

Supposez un index B+ d'uniquement 2 niveaux (le niveau racine ne compte pas) où chaque bloc peut contenir 5 clés de type entier.

- 8) Quel est le nombre maximal de clés que cet index peut stocker ?
- 9) Quel est le nombre minimal de clés que cet index peut stocker ?

Supposez une taille de bloc de 8192 octets, les clés et les pointeurs sont des entiers de 4 octets.

- 10) Quel est le nombre de clés que peut stocker un bloc ?

Considérez l'arbre B+ de la Figure 1.

- 11) Donner les arbres correspondants à l'insertion successive des clés 28, 20 et 8.
- 12) Donner les arbres correspondants à la suppression successive des clés 5, 19 et 7.
- 13) Donner une *liste* de clés de telle manière que :
 - a) L'insertion des clés dans l'ordre donné (par votre liste) et la suppression des clés dans l'ordre inverse (e.g., insert a, insert b, delete b, delete a) donne en résultat l'arbre original.

- b) L'insertion des clés dans l'ordre donné (par votre liste) et la suppression des clés dans l'ordre inverse (e.g., insert a, insert b, delete b, delete a) donne en résultat un arbre différent.
- 14) Décrire un algorithme pour un arbre B+ acceptant des doublons (plusieurs clés avec même valeur).

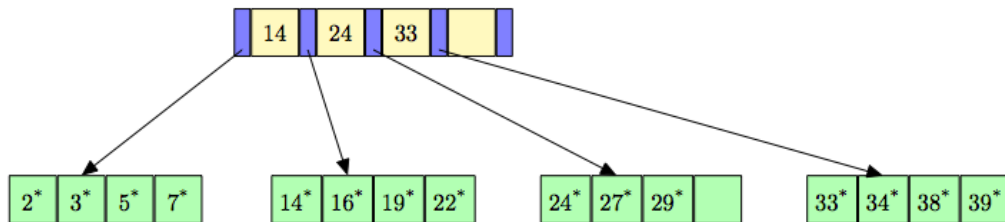


Figure 1. Arbre B+ initial.

TD : TRANSACTIONS

Exercice 1

- 1) Considérez les opérations suivantes de la transaction T1 : $l_1(x)$, $e_1(x)$, $l_1(y)$, $e_1(y)$, $l_1(z)$.
Donnez un exemple de transaction T2 laquelle exécutée de manière concurrente avec T1 sans un contrôle de la concurrence :
 - a) fait interférence à T1.
 - b) ne fait pas d'interférence à T1.

- 2) Supposez les ordonnancements suivants et dites s'il est équivalent à un ordonnancement séquentiel en utilisant la méthode d'échange d'opérations. Si la réponse est oui, donner l'ordonnancement séquentiel correspondant :
 - a) O1 = $l_3(y)$, $l_4(x)$, $l_4(y)$, $e_4(x)$, $e_3(y)$.
 - b) O2 = $l_1(x)$, $l_2(y)$, $e_1(y)$, $l_3(y)$, $e_2(z)$, $e_1(x)$, $l_3(x)$, $e_3(z)$.

- 3) Construisez les graphes de dépendances des ordonnancements suivants et dites s'ils sont conflit-sérialisables :
 - a) O3 = $l_2(y)$; $l_2(x)$; $e_2(y)$; $l_3(z)$; $l_3(y)$; $l_1(x)$; $e_2(x)$; $l_1(z)$; $e_3(z)$
 - b) O4 = $l_4(x)$; $e_4(x)$; $l_5(y)$; $l_5(x)$; $l_5(z)$; $l_4(y)$; $e_4(y)$; $e_3(z)$
 - c) O1 et O2

- 4) Considérez les transactions suivantes :
 $T1 = l_1(x)$; $l_1(y)$; $e_1(x)$
 $T2 = l_2(x)$; $l_2(y)$; $e_2(x)$; $e_2(y)$; $l_2(x)$
 Donnez un ordonnancement des actions T1 et T2 où :
 - a) un conflit écriture-lecture (lectures impropres) pourrait exister.
 - b) un conflit écriture-écriture (perte de mise à jour) pourrait exister.
 - c) un conflit lecture-écriture (lectures non reproductibles) pourrait exister.

Exercice 2

- 1) Montrez comment le protocole de concurrence 2PL ne permet pas l'existence des trois ordonnancements précédents (question 4 de l'exercice 1).
- 2) Donnez un ordonnancement conflit-sérialisable en utilisant le protocole 2PL et en rajoutant les opérations de verrouillage exclusif ($ve_1(x)$), de verrouillage partagé ($vp_1(x)$), de libération de verrous ($lib_1(x)$) et de validation (c_1) :
 - a) des dernières transactions T1 et T2 (question 4 de l'exercice 1).
 - b) des transactions participant aux ordonnancements O3 et O4. Pour cette question, identifiez préalablement les transactions participant aux ordonnancements avec leur liste d'opérations et proposez un ordonnancement conflit-sérialisable ensuite. Rappelez vous que l'ordre des

opérations au sein d'une transaction ne doit pas être modifié. Vos ordonnancements seront appelés O3' et O4'.

3) Est-ce que l'ordonnancement suivant peut être le résultat d'une exécution avec S-2PL ?

O5 = $vp_1(x) ; l_1(x) ; ve_1(z) ; e_1(z) ; lib_1(x) ; ve_2(x) ; e_2(x) ; lib_1(z) ; c1 ; vp_2(z) ; l_2(z) ; lib_2(x) ; lib_2(z) ; c2$

4) Donnez le graphe Qui Attend Quoi (QAQ) des ordonnancements O3, O4 et O5 et dites à quel moment un interblocage pourrait avoir lieu.

1. Quelles sont les avantages et inconvénients de 2PL et S-2PL ?

Liste non-exhaustive :

Feuilles de Travaux Pratiques

Projet

TP1 : Définition de relations sous Oracle

TP2 : Reporting SQL Plus

TP3 : PL/SQL

TP4 : Plans d'exécution

TP5 : Transactions

Cours Bases de Données 2 - X32I040

Projet Base de Données 2

Travail à faire en quadrinôme selon les groupes de TP.

Définir le schéma d'une base de données que vous allez développer tout au long du cours. Au fur et à mesure que le cours se déroulera, vous allez pouvoir compléter votre base de données avec des contraintes, du contrôle d'accès, des déclencheurs, etc. Vous allez pouvoir améliorer les performances de votre base en définissant des requêtes performantes, en introduisant des vues, des indexes et des transactions.

Votre base de données doit être normalisée en 3FN et en FNBC.

Le sujet de votre base de données est libre, cependant, l'originalité et le rapprochement avec la réalité compteront pour la note finale.

Consignes

1. Définir une seule table avec tous les attributs de votre base de données.
2. Définir des tuples de votre table (e.g., une dizaine) pour mettre en évidence les dépendances fonctionnelles.
3. Définir l'ensemble de dépendances fonctionnelles de votre table.
4. Décomposer votre table en plusieurs tables à l'aide des 2 algorithmes de normalisation; synthèse (3FN) et décomposition (FNBC). Faites si nécessaire plusieurs essais jusqu'à trouver l'ensemble de schémas qui vous satisfait selon l'usage préconisé de la base (requêtes les plus fréquentes). Argumenter votre choix final.
5. Tester la normalisation de votre schéma avec un outil comme : http://uisacad5.uis.edu/cgi-bin/mcrem2/database_design_tool.cgi

1er rendu sur les points 1-5 : Rendre à votre chargé de TD (sur madoc) votre schéma pour vérification : **un seul fichier en format pdf** avec la démarche suivie. Votre rendu doit commencer à prendre la forme du compte rendu final avec les points a, b, c du point 9 ici-bas.

6. Créer les tables de votre base de données. De préférence en Oracle.
7. Ajouter des triggers, des fonctions et des vues intéressantes à votre projet.
8. Attribuer des droits (simulant différents utilisateurs de différent type) aux comptes de vos collègues.
9. Rédiger un compte rendu court avec :
 - a. Introduction.
 - b. Déroulement de votre processus et comment vous vous êtes distribué les tâches.
 - c. Modèle Relationnel de votre base de données (tables, clés, types, cardinalités, etc).
 - d. Explication des fonctions, triggers, vues, etc. de votre base de données.
 - e. Atouts, améliorations possibles et critiques de votre travail.
 - f. Conclusion.

Démonstration de votre base de données à votre chargé de TP : **date à préciser**.

Rendu de votre compte rendu (sur madoc) à votre chargé de TP : **date à préciser**.

Le compte rendu devra prendre en compte toutes les modifications suggérées par vos enseignants.

TP1 : Définition de relations sous Oracle

Modalités de connexion

Les noms de « user » Oracle et les mots de passe seront donnés en séance.

Utilisation du client Oracle local (sqlplus)

Pour le nom de user Oracle *usr3*, dans un terminal taper la commande :

```
sqlplus usr3@cienetdb
```

Il faut ensuite fournir le mot de passe de ce compte. On peut alors travailler avec sqlplus.

Utilisation de SQL Developer

Dans un terminal, taper la commande : `sqldeveloper &`

La fenêtre SQL Developer apparaît. Créer alors une nouvelle connexion (*new... db connection*).

Paramètres à indiquer :

1. Le nom de la machine hôte est : *oracle.ensinfo.sciences.univ-nantes.prive*
2. le nom de l'instance (sid) : *cienetdb*
3. le port est : *1521*

Relations à définir

Dans le but de mettre en place un système de recommandation dans une librairie, la base de données est constituée des relations suivantes :

Clients(idcl number, nom varchar2(20), pren varchar2(15), adr varchar2(30), tel varchar2(12))

Les clients ont un identifiant ; on mémorise leur nom, prénom, adresse et numéro de téléphone.

Livres(refl varchar2(10), titre varchar2(20), auteur varchar2(20), genre varchar2(15))

Les livres ont une référence qui les identifie. On mémorise leur titre, leur auteur, leur genre.

Achats(idcl number, refl varchar2(10), dateachat date)

On mémorise tous les achats effectués par les clients, avec la date d'achat. Cette date d'achat est comprise entre le 1er janvier 2008 et le 31 décembre 2013.

Avis(idcl number, refl varchar2(10), note number(4,2), commentaire varchar2(50))

Les clients peuvent indiquer s'ils ont aimé un ouvrage, en donnant une note, comprise entre 1 et 20 (cela correspond à un nombre d'étoiles) ; ils peuvent aussi laisser un commentaire.

Q1. Compléter la création des tables du fichier *create_tables_TP1.sql* en implantant le code pour la création de la table *Avis*. Veiller à indiquer les clés, les clés étrangères et les autres contraintes qui peuvent être indiquées. Exécuter le fichier .sql pour créer les tables dans votre espace de travail.

Q2. Exécuter le fichier *insert_tables_TP1.sql* pour insérer de tuples.

Q3. Ecrire les requêtes SQL permettant d'afficher :

1. Les meilleures ventes : les livres (titre, auteur, genre) qui ont été achetés en plus de 10000 exemplaires.
2. Les livres qui obtiennent une note moyenne supérieure à 16.
3. Les clients qui n'ont pas renseigné l'attribut **commentaire** d'un livre (quel client, livre, note).

TP2 Reporting SQL Plus

Définition d'une séquence

Une séquence est un objet de la base qui génère des entiers, tous différents. Cela peut être pratique pour définir les valeurs d'une clé. Exemple de définition :

```
CREATE SEQUENCE maSequence START WITH 1 INCREMENT BY 1 ;
```

Utilisation dans une insertion :

```
INSERT INTO maTable VALUES (maSequence.nextval, 'toto', 5) ;
```

Q1. Créez une séquence pour l'attribut clé idcl de la relation Clients.

Un truc pratique

Nous nous plaçons sous SqlPlus. Supposons que l'on veuille effacer le contenu de toutes les tables d'un utilisateur. Il faut à priori connaître le nom de ses différentes tables et effacer au fur et à mesure dans chacune des tables. On se propose ici d'automatiser ce processus en mémorisant automatiquement dans un fichier l'ensemble des commandes SQL nécessaires.

Q2. Tester la requête suivante :

```
SELECT 'DELETE FROM ' || table_name || ';'
FROM USER_TABLES ;
```

Q3. L'objectif est maintenant de mémoriser les différentes lignes obtenues, par une suite de commandes, dans le fichier *effacer.sql*. Exécuter les commandes suivantes :

```
SET ECHO OFF
SPOOL effacer.sql
SET ECHO OFF
SET FEEDBACK OFF
SET HEADING OFF
SET PAGESIZE 0
<placer ici la requête de la question Q2.>
SPOOL OFF
SET ECHO ON
```

On peut ensuite lancer l'exécution du fichier par *@ effacer.sql* (ou *start effacer.sql*) et voir si toutes les tables sont effacées.

Note : il est possible sous SQL Developer de « exécuter une sortie de script en tant que script » mais il semble difficile de supprimer par exemple les noms de colonnes.

Q4. Proposer un autre exemple sur le même modèle.

Rapports sous sqlplus

Sous sqlplus on peut formater le résultat de requêtes pour être présentées et analysées avant de faire de la programmation dans un langage applicatif. Dans cette partie vous allez faire le réglages nécessaires dans un fichier .sql pour enregistrer le résultat d'une requête sur la base de données du TP1 dans un fichier .lst.

Q5. Ajouter l'attribut *prix* de type numérique à la relation Achats du TP1 avec la commande alter

table.

Q6. Ecrire la requête sql et le code sqlplus permettant d'enregistrer automatiquement dans le fichier 2013-01-28-achats.lst les numéros de client ainsi que chacun de leurs achats avec le genre et le prix payé. On calculera par client la moyenne des prix payé et l'addition des achats. Le résultat sera ordonné par numéro de client et date d'achat. Le résultat doit rassembler à ceci :

```
          Achats des clients au 28 janvier 2013
Idcl      Date d'achat      Genre      Prix
1         30-JAN-10         ROMAN      20.30
          12-MAY-10         HISTORIQUE  49.99
          20-FEB-11         ROMAN      25
*****
avg                                              31.76
sum                                              95.29

2         21-JUN-90         POESIE     15
*****
avg                                              15
sum                                              15
```

TP3 : PL/SQL

On conserve les relations définies dans le TP1, en apportant les modifications et ajouts suivants :

* A la relation *livres*, on ajoute un attribut *note_moy* : pour un livre donné, cet attribut est calculé en faisant la moyenne des notes qui lui ont été attribuées dans la relation *Avis*.

* On ajoute des relations. La librairie propose des parcours sur différents thèmes, chaque parcours étant constitué d'événements ayant lieu à la librairie (exposés d'auteurs, discussions, etc). Tout client peut s'inscrire aux parcours de son choix. Les relations proposées sont les suivantes :

parcours(idp varchar2(10), intitulep varchar2(15), genre varchar2(15), date_deb date)

Les parcours ont un intitulé et peuvent concerner un genre (ex: policier) ; si ce n'est pas le cas, le champ correspondant prend la valeur indéfinie. Un parcours commence à une certaine date.

compo_parcours(idp varchar2(10), id_evt varchar2(10))

Cette relation mémorise la composition des parcours ; un parcours contient 4 événements. Un même événement peut être proposé pour différents parcours.

inscrip_parcours(idcl number, idp varchar2(10))

Cette relation mémorise l'inscription d'un client à un parcours.

inscrip_evt(idcl number, idp varchar2(10), id_evt varchar2(10))

Pour des raisons de logistique (capacité de la salle, etc), on maintient une relation mémorisant les inscriptions des clients aux différents événements.

Mise à jour de la note moyenne

Q1. Modifier la table *livres* en ajoutant l'attribut *note_moy*. (a) Ecrire un bloc PL/SQL qui met à jour cet attribut à partir de la relation *Avis*, pour un livre demandé à l'utilisateur ; (b) Ecrire un bloc PL/SQL qui met à jour cet attribut pour tous les livres. (c) Reprendre la question précédente en écrivant une procédure.

Q2. Ecrire un déclencheur (trigger) qui à l'ajout ou à la modification d'un avis, met à jour l'attribut *note_moy* de la référence de livre concernée. Que constatez-vous ? Commentez.

Cohérence Avis-Achat

Q1. Ecrire un déclencheur qui assure qu'un client qui insère un avis a bien acheté le livre.

Q2. Ecrire un bloc PL/SQL de mise à jour de la table *Avis* (la note et/ou le commentaire) qui assure qu'un client ne peut modifier que les tuples qu'il a insérés (on suppose que l'identifiant de client et la référence du livre sont demandés à l'utilisateur).

Traitement d'une inscription à un parcours

Implanter le schéma des relations : *parcours*, *compo_parcours*, *inscrip_parcours*, *inscrip_evt*.

Q1. Ecrire un code PL/SQL qui réalise l'inscription d'un client à un parcours (clients et parcours demandés à l'utilisateur) : le tuple correspondant est inséré dans la relation *inscrip_parcours*. La relation *inscrip_evt* est remplie automatiquement (cf *compo_parcours*). Transformer en procédure.

Q2. Ecrire un déclencheur pour la relation *Achats* qui propose des parcours (qui n'ont pas encore commencé) de même genre que celui du livre acheté.

TP4 – Plans d'exécution

L'objectif de ces travaux pratiques est de réaliser quelques manipulations visant à illustrer le choix de plans d'exécution de requêtes par l'optimiseur par défaut (Cost Based Optimizer).

La base considérée concerne un extrait de la base de DBLP des publications du domaine informatique (<http://www.dblp.org/db/>). Les noms des relations sont: facts, authors, collaborations, dates, publications, squads, supports. Vous avez le droit uniquement de lecture sur ces tables.

Prise en main de la base

Q1. Consultez le schéma de chacune des tables, identifiez les clés primaires, les clés étrangères et les possibilités de jointures.

Q2. Quel est le nombre de lignes de chaque table ?

Visualisation du plan d'exécution d'une requête

Pour visualiser le plan d'exécution d'une requête, on peut utiliser le code SQL suivant (aussi bien sous SQLPLUS que sous l'interface SQL Developer) :

- `explain plan for <ecrire ici la requête>;`
- `select * from table(dbms_xplan.display);`

Sous SQL Developer, on peut aussi afficher le plan d'exécution avec la touche F10.

Dans madoc vous pouvez voir de l'information sur l'outil explain :

<http://madoc.univ-nantes.fr/mod/resource/view.php?id=142034>

Et bien sur, sur la documentation Oracle :

http://docs.oracle.com/cd/B28359_01/server.111/b28274/ex_plan.htm

Q1. Exécuter et visualiser le plan des requêtes suivantes. Pour chacune noter les différentes informations fournies et comparer le plan pour R2 et celui pour R3.

- R1 : `select title from publications where nb_pages > 20 ;`
- R2 : `select title from publications natural join facts where date_id = 2008 ;`
- R3 : `select title from publications where publication_id in (select publication_id from facts where date_id = 2008);`
- R4 : `select title from publications natural join facts natural join collaborations natural join authors where last_name= 'Rosenthal' ;`

Q2. On considère maintenant la table publications1 au lieu de publications. Que constatez-vous dans les plans utilisés ?

Q3. Visualisez le plan d'exécution d'autres requêtes de votre choix.

Les index

Q1. Faire le bilan des index définis sur les différentes tables de la base (y compris publications1) avec leurs caractéristiques. Proposez une explication pour les résultats de la question **Q2** précédente.

Vous pouvez avoir de l'information sur les indexes en interrogeant le dictionnaire ALL_INDEXES.

Q2. Il est possible d'utiliser l'interface de SQL Developer pour définir un index. On peut aussi définir un index par la commande SQL simplifiée :

```
create index <nom_index> on <relation> (<nom_index>) ;
```

Tous les détails liés au stockage de l'index sont alors définis par défaut par Oracle. Voir : http://download.oracle.com/docs/cd/B28359_01/server.111/b28310/indexes003.htm#i1006525)

En utilisant les tables que vous avez définies dans les premiers TPs,

1. Visualiser le plan d'exécution d'une requête du type :
2. `select * from avis where note > 10;`
3. et du type :
4. `select * from avis where round(note,2) > 10;`
5. Créer un index sur l'attribut note, puis visualiser de nouveau le plan d'exécution des requêtes précédents. Conclure.

Travail Pratique 1

Les transactions sous Oracle 10g

La norme SQL92 propose quatre niveaux d'isolation (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ et SERIALIZABLE). Oracle implémente uniquement deux : READ COMMITTED et SERIALIZABLE.

Le niveau d'isolation d'une transaction est spécifié comme suit : SET TRANSACTION ISOLATION LEVEL { SERIALIZABLE | READ COMMITTED };

Le niveau d'isolation peut être défini pour la durée d'une session active : ALTER SESSION SET ISOLATION_LEVEL { SERIALIZABLE | READ COMMITTED };

Sous Oracle, une instruction DML (SELECT, INSERT, UPDATE, DELETE), DDL (CREATE, ALTER, DROP) ou DCL (GRANT, REVOKE, COMMIT, ROLLBACK) démarre une transaction qui se termine par un COMMIT si tout s'est bien passé.

Dans Oracle on peut définir le mode d'une transaction (lecture seule ou lecture écriture) par l'instruction : SET TRANSACTION { READ ONLY | READ WRITE };

Par défaut, le mode d'une transaction est READ WRITE et son niveau d'isolation est READ COMMITTED.

Dans la suite vous allez réaliser des expériences qui vous permettront de comprendre et d'analyser la manière comment Oracle gère la concurrence.

Initialisation de l'environnement

Chaque binôme utilisera 2 comptes oracle (chaque étudiant utilisera le compte qui lui a été attribué). Le TP est à faire avec une session par compte.

Ne pas ajouter d'autres commandes pendant les expériences. Entre chaque expérience vous pouvez faire un SELECT * FROM T suivi d'un COMMIT.

Session 1	Session 2
<pre>Drop table T cascade constraints; create table T (A integer, B integer); grant select, update, insert, delete on T to compte2; insert into T values (0,0); insert into T values (2,3); insert into T values (0,1); commit;</pre>	<pre>Drop synonym T ; create synonym T for compte1.T;</pre>

Expérience 1

Session 1	Session 2
<pre>select * from T; select * from T; select * from T; commit;</pre>	<pre>update T set A=A+1; commit ;</pre>

Question. Quel problème est mis en évidence ?

Expérience 2

Session 1	Session 2
<pre>update T set A=A+1; select * from T; commit ;</pre>	<pre>insert into T values(3,7); select * from T; delete from T where A=2; delete from T where A=1; commit ;</pre>

Qu'est-ce que vous pouvez déduire de cette exécution ? Est-ce qu'elle est sérialisable ?

Expérience 3

Session 1	Session 2
<pre>update T set A=3 where A=2; select * from T; update T set B=2 where B=3 ; commit ;</pre>	<pre>update T set B=2 where B=3; select * from T; update T set A=3 where A=2; commit ;</pre>

A quel niveau de granularité sont posés les verrous ? Comment se terminent ces transactions ? Pourquoi ?

Expérience 4

Session 1	Session 2
<pre>Select * from T where B=7 for update ; update T set B=6 where B=7 ; commit ;</pre>	<pre>update T set B=10 where B=0; select * from T; update T set A=5 where A=3; commit ; select * from T; commit ;</pre>

Quelle a été l'utilité de la clause select for update ?

Expérience 5

Session 1	Session 2
<pre>Set transaction read only; Select * from T ; Select * from T ; Select * from T ; commit ; Select * from T ; commit ;</pre>	<pre>select * from T; update T set A=A+1; commit ; select * from T; update T set A=A+1; commit ;</pre>

Qu'est-ce que vous avez remarqué ?

Expérience 6

Session 1	Session 2
<pre>lock table T in share mode; commit; lock table T in exclusive mode; commit;</pre>	<pre>select * from T for update; update T set A=A+1; commit ; select * from T ; select * from T for update; commit ;</pre>

Expliquez les résultats obtenus.

Expérience 7

Session 1	Session 2
<pre>alter session set isolation_level=serializable; select * from T ; select * from T ; select * from T ; commit; select * from T ; commit ;</pre>	<pre>alter session set isolation_level=serializable; update T set A=A+1; commit ; insert into T values (0,0) ; commit ; select * from T ; commit ;</pre>

Expliquez les résultats obtenus.

Expérience 8

Session 1	Session 2
<pre>select * from T ; update T set A=A+1 where A=8; commit ;</pre>	<pre>select * from T; delete from T where A=9; delete from T where B=2;</pre>

Expliquez le problème rencontré.

Expérience 8 bis

Session 1	Session 2
<pre>Alter table T add constraint PK primary key (B) ; Grant references on T to compte2 ; Select * from T ; Delete from T where A=0 ; Commit ;</pre>	<pre>Create table S(B number references T, C number) ; Select * from T ; Insert into S values(0,1) ; commit ;</pre>

Qu'est-ce que vous avez remarqué dans cette exécution ? Comment résoudre le problème rencontré ?

Expérience 9

Maintenant, c'est à vous de jouer. Il y a d'autres caractéristiques transactionnelles que nous n'avons pas analysé, par exemple, les variations de verrouillage d'une table (lock table) à savoir « share mode » et « exclusive mode » avec « no wait », ou « in row share mode », « in row exclusive mode ». Vous pouvez également exécuter les premières expériences du TP avec le mode serialisable.

Expérience 10

Pour terminer, dans cette expérience vous allez constater l'utilité des clauses ROLLBACK et SAVEPOINT

Session 1	Session 2
<pre>Insert into T values (0,0) ; Insert into T values (0,0) ; Rollback ; Select * from T ; Insert into T values (0,0) ; Insert into T values (0,0) ; Commit ; Rollback ;</pre>	<pre>update T set B=B+1; savepoint A ; insert into T values(10,10) ; savepoint B ; select * from T ; rollback to savepoint B select * from T ;</pre>

Expliquez les résultats obtenus.