



Session : 1	Durée de l'épreuve : 2 heures
Date : janvier 2018	Documents autorisés : tous
Mention Informatique	Matériel utilisé : aucun
Master 2 ^{ème} année 2018 : Administration BD (HMIN328)	

1 BD exemple : les transactions commerciales

1.1 Enoncé

La base de données considérée, gère des transactions commerciales qui sont conclues de particulier à particulier, pour le compte d'une plateforme d'échange qui en garantit la sécurité.

Le schéma de la base de données est volontairement simplifié. Les acheteurs et vendeurs sont envisagés comme autant de personnes qui vont respectivement proposer et accepter une transaction. Une transaction commerciale ne concerne qu'un et un seul objet qui peut être de type bien matériel, bien immatériel ou service. Un objet peut être vendu à un prix différent de sa valeur marchande. La relation **Personne** est identifiée par un code de personne (codeP), et est caractérisée par un nom (nomP), un prénom (prenomP) et une ville d'origine (villeP). La relation **Objet** est identifiée par un code d'objet (codeO), et contient également les attributs : désignation (designation), valeur marchande en euros (valeurM) et une catégorie d'objet (categorieO) qui puise ses valeurs parmi {bien_materiel, bien_immateriel, service}. La relation **TransactionC** est identifiée par un code de transaction (codeT), et contient une date de transaction (dateT), une valeur en euros correspondant au montant négocié de la transaction (valeurE), un code d'acheteur (codeAcheteur), un code de vendeur (codeVendeur) et un code d'objet cédé (codeO).

1.2 Schéma Relationnel

Les attributs portant les contraintes de clés primaires sont en gras. Les contraintes de clés étrangères sont données sous la forme de contraintes d'inclusion. Les types des attributs sont également indiqués.

- Personne(**codeP varchar(6)**, nomP varchar(20), prenomP varchar(20), villeP varchar(16))
- Objet(**codeO varchar(6)**, designation varchar(20), valeurM float, categorieO varchar(15))
- TransactionC(**codeT varchar(6)**, dateT date, valeurE float, codeAcheteur varchar(6), codeVendeur varchar(6), codeO varchar(6))
avec TransactionC(codeAcheteur) \subseteq Personne(codeP)
avec TransactionC(codeVendeur) \subseteq Personne(codeP)
avec TransactionC(codeO) \subseteq Objet(codeO)

2 Optimisation de requêtes

2.1 Index (6 points)

2.1.1 Index implicite

La contrainte de clé primaire posée sur l'attribut codeO de la relation Objet est nommée OBJET_PK. Des instructions SQL, que vous aurez à commenter, sont données. Le résultat

de la requête de consultation de la vue index_stats est également fourni.

```
analyze index objet_pk validate structure;
SELECT name, btree_space, most_repeated_key, lf_rows, br_rows, height
FROM index_stats;
```

NAME	BTREE_SPACE	MOST_REPEATED_KEY	LF_ROWS	BR_ROWS	HEIGHT
OBJET_PK	7996	1	6	0	1

Vous répondrez aux questions suivantes avec concision :

1. pourquoi l'index porte le même nom que la contrainte de clé primaire ?
2. quelle est la structure de données sur laquelle s'appuie cet index ? Est ce le type d'index le plus fréquemment utilisé sous Oracle ? Précisez en les intérêts.
3. cet index est t'il dit unique ? quelle est l'information dans le résultat de la requête sur la vue index_stats qui peut vous renseigner à ce sujet ?
4. expliquer l'intérêt des attributs BTREE_SPACE, MOST_REPEATED_KEY, LF_ROWS, BR_ROWS, HEIGHT et commentez les valeurs obtenues

2.1.2 Index explicite

1. Donner l'ordre SQL de création d'index (que vous pouvez nommer obj_secondary_idx) sur l'attribut codeO de TransactionC
2. cet index est t'il unique ? Sur quelle structure de données s'appuie t'il ?
3. Donner un exemple de requête de consultation qui pourrait être efficacement servie par ce nouvel index

2.2 Plan d'exécution d'une requête (4 points)

Expliquer la sémantique associée à la requête suivante et décrivez le plan d'exécution obtenu (figure partie 2.2 en fin de sujet). Donner les différents opérateurs physiques disponibles sous Oracle pour le calcul d'une jointure.

```
explain plan for SELECT /*+ use_merge(p t) */ nomp, prenomp
from personne p, transactionC t
where codeP = codeAcheteur;
```

```
select plan_table_output from table (dbms_xplan.display()) ;
```

2.3 Différentes écritures d'une même requête (4 points)

Vous écrirez de deux manières différentes, la requête : donnez le nom et la ville des personnes qui sont soit des vendeurs, soit des acheteurs, soit les deux. Vous en donnerez les expressions algébriques et vous formulerez votre avis sur la requête qui vous semble la plus efficace.

3 Architecture Oracle

3.1 Eléments mis en jeu (4 points)

Une figure incomplète de l'organisation physique d'un serveur de données Oracle vous est donnée (vous la reproduirez sur votre copie). Vous complèterez les informations manquantes en ce qui concerne les structures et les processus mis en jeu. Les principales vues



du méta-schéma qui servent à renseigner les différentes structures mémoire et processus d'arrière plan (ou autres processus) sont listées, vous positionnerez également ces vues sur la figure.

3.1.1 Vues à positionner sur la figure

1. v\$instance, v\$database, v\$bgprocess, v\$sga, v\$sgastat, v\$sql, v\$sql_plan, v\$datafile, v\$logfile, v\$bh, v\$librarycache

3.2 Structures mémoire à détailler

Vous expliquerez de manière très brève les rôles principaux dévolus aux structures mémoires listées ci-dessous

1. Database buffer cache
2. Redo log buffer
3. Data dictionary cache
4. Library cache

4 Transaction (2 points)

Une séquence d'ordres SQL vous est fournie. Vous définirez le rôle général des ordres commit et rollback. Vous en expliquerez les effets sur la séquence proposée.

```
alter table TransactionC add constraint codeDifferent check (codeAcheteur != codeVendeur);
rollback;
insert into Personne values ('A17-3','Heric','Zoe',null);
insert into Personne values ('V17-3','Saunier','Elise','Uzes');
commit;
update Personne set villeP = 'Mars' where codeP = 'A17-3';
exit;
```

5 Annexe

```
insert into Personne values ('A17-1','Martien','Eloi','Uzes');
insert into Personne values ('A17-2','Calixte','Marthe','Montpellier');
insert into Personne values ('V17-1','Martinez','Joan','Nimes');
insert into Personne values ('V17-2','Sorlin','Agathe','Montpellier');
```

```
insert into Objet values ('2017-1','Tricycle',134,'bien_materiel');
insert into Objet values ('2017-2','image jpg',100,'bien_immateriel');
insert into Objet values ('2017-3','poncho',10,'bien_materiel');
insert into Objet values ('2017-4','brushing',39,'service');
insert into Objet values ('2017-5','cuisine',59,'service');
```

```
insert into TransactionC values ('T1-17','19-apr-2017', 140, 'A17-1','V17-1','2017-1');
insert into TransactionC values ('T2-17','19-jul-2017', 40, 'A17-1','V17-2','2017-4');
insert into TransactionC values ('T3-17','19-jun-2017', 40, 'V17-2','V17-1','2017-3');
insert into TransactionC values ('T4-17','19-sep-2017', 41, 'A17-1','V17-1','2017-5');
```

Plan hash value: 1517166987

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	130	6 (17)	00:00:01
1	MERGE JOIN		5	130	6 (17)	00:00:01
2	TABLE ACCESS BY INDEX ROWID	PERSONNE	6	120	2 (0)	00:00:01
3	INDEX FULL SCAN	PERSONNE_PK	6		1 (0)	00:00:01
* 4	SORT JOIN		5	30	4 (25)	00:00:01
5	TABLE ACCESS FULL	TRANSACTIONC	5	30	3 (0)	00:00:01

Predicate Information (identified by operation id):

4 - access("CODEP"="CODEACHETEUR")
filter("CODEP"="CODEACHETEUR")

FIGURE 1 – Plan d'exécution de la requête

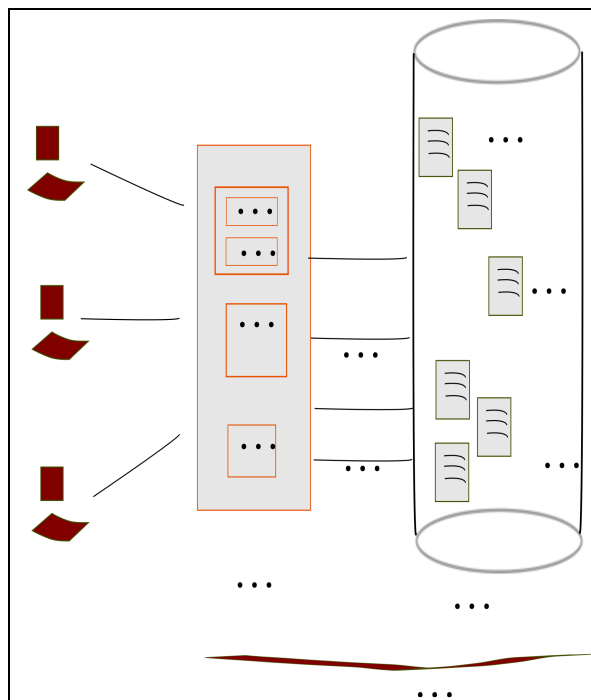


FIGURE 2 – Architecture physique Oracle (structures et processus)