



Session : 1

Durée de l'épreuve : 2 heures

Date : 13 janvier 2020

Documents autorisés : tous

Mention Informatique

Matériel utilisé : aucun

Master 2^{ème} année : Administration BD (HMIN328)

1 BD exemple : musées nationaux

1.1 Enoncé

Une interprétation textuelle volontairement simplifiée de l'univers des musées vous est donnée. Un musée, identifié par un code et caractérisé par un nom, un descriptif, un numéro et un nom de rue et une ville, possède un ensemble d'œuvres, qui sont exposées de manière permanente. Une œuvre est identifiée par un code, et est décrite par un titre, une année, un lieu de création ; et un type de création prenant ses valeurs parmi (tableau, dessin, sculpture). Une œuvre peut être anonyme, ou au contraire avoir plusieurs auteurs. Ces auteurs sont rangés parmi les artistes. Un artiste est identifié par son nom et son prénom, et possède également un genre, une année de naissance et le cas échéant, une année de décès, ainsi qu'une nationalité (pays).

Le schéma relationnel, qui retranscrit cette interprétation, vous est donné ci-dessous.

1.2 Schéma Relationnel

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

musée(**codeMusée** varchar(10), nomMusée varchar(12), numRue varchar(5), nomRue varchar(40), descriptif varchar(100), nomVille varchar(20))

artiste(**nomA** varchar(20), **prenomA** varchar(20), genre varchar(1), annee_naissance integer, annee_deces integer, pays varchar(8))

oeuvre(**codeO** varchar(10), titre varchar(50), annee_creation integer, lieu_creation varchar(12), type_creation type varchar(12))

auteur_de(**nomA** varchar(20), **prenomA** varchar(20), **codeO** varchar(10))
avec auteur_de(nomA, prenomA) \subseteq artiste(nomA, prenomA)
avec auteur_de(codeO) \subseteq oeuvre(codeO)

possede(**codeMusée** varchar(10), **codeO** varchar(10))
avec possede(codeMusée) \subseteq musée(codeMusée)
avec possede(codeO) \subseteq oeuvre(codeO)

2 Optimisation de requêtes et index

2.1 Plan d'exécution d'une requête (7 points)

2.1.1 Série de questions

Une série de questions relatives à l'exécution d'une requête, vous est proposée. Vous pouvez répondre à ces questions dans l'ordre que vous souhaitez.

1. Expliquer la sémantique associée à la requête suivante et dessinez l'arbre algébrique correspondant (en tirant parti du plan d'exécution donné). Vous pouvez annoter l'arbre algébrique avec des considérations concernant le plan physique (coût et temps estimé des différentes opérations, nombre de tuples estimés pour chaque étape).
2. Vous décrierez le plan d'exécution obtenu présenté, et les statistiques attenantes. Vous indiquerez si des index sont utilisés pour améliorer les performances du calcul. Si des index sont mobilisés, vous en donnerez le nom, et désignerez les attributs sur lesquels s'appliquent ces index.
3. Quel est l'opérateur physique exclusivement exploité pour le calcul des jointures? Vous expliquerez les raisons qui ont poussé l'optimiseur à faire le choix de cet algorithme/opérateur.
4. Pensez vous que des index supplémentaires pourraient améliorer l'efficacité du calcul, si oui indiquer sur quels attributs pourraient s'appliquer ces index, et définissez en l'ordre de création.
5. Combien de blocs sont parcourus pour satisfaire la requête? Indiquez également si ces blocs sont sur disque, ou bien au contraire dans le cache de données (database buffer cache).
6. rajouter à la requête, les directives nécessaires à l'utilisation de l'opérateur tri-fusion pour chacune des jointures.

```
set autotrace on
```

```
select a.nomA, a.prenomA from artiste a, auteur_de ad, musee m, possede p
where a.nomA = ad.nomA and a.prenomA = ad.prenomA and ad.codeO = p.codeO
and m.codeMusee = p.codeMusee and nomMusee='Musee Fabre' order by a.nomA ;
```

```
Plan hash value: 67542150
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	77	3 (34)	00:00:01
1	SORT ORDER BY		1	77	3 (34)	00:00:01
2	NESTED LOOPS		1	77	2 (0)	00:00:01
3	NESTED LOOPS		1	57	1 (0)	00:00:01
4	NESTED LOOPS		1	48	0 (0)	00:00:01
5	INDEX FULL SCAN	AUTEUR_DE_PK	1	31	0 (0)	00:00:01
* 6	INDEX UNIQUE SCAN	ARTISTE_PK	1	17	0 (0)	00:00:01
* 7	INDEX FULL SCAN	POSSEDE_PK	1	9	1 (0)	00:00:01
* 8	TABLE ACCESS BY INDEX ROWID	MUSEE	1	20	1 (0)	00:00:01
* 9	INDEX UNIQUE SCAN	MUSEE_PK	1		0 (0)	00:00:01

```
Predicate Information (identified by operation id):
```

```

6 - access("A"."NOMA"="AD"."NOMA" AND "A"."PRENOMA"="AD"."PRENOMA")
7 - access("AD"."CODEO"="P"."CODEO")
  filter("AD"."CODEO"="P"."CODEO")
8 - filter("NOMMUSEE"='Musee Fabre')
9 - access("P"."CODEMUSEE"="M"."CODEMUSEE")
```

```
Statistics
```

```

0 recursive calls
0 db block gets
```



```

13 consistent gets
0 physical reads
0 redo size
572 bytes sent via SQL*Net to client
384 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
4 rows processed

```

2.1.2 Questions complémentaires (3 points)

Une nouvelle requête SQL est donnée :

```

select nomA, prenomA from artiste where (nomA, prenomA) in (select nomA, prenomA
from auteur_de where codeO in (select codeO from possede where codeMusee =
(select codeMusee from Musee where nomMusee='Musee Fabre')))) order by nomA;

```

avec le plan d'exécution qu'en propose Oracle :

Execution Plan

Plan hash value: 4086681405

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	41	5 (40)	00:00:01
1	SORT ORDER BY		1	41	5 (40)	00:00:01
2	NESTED LOOPS		1	41	4 (25)	00:00:01
3	VIEW	VW_NSO_1	1	24	3 (0)	00:00:01
4	HASH UNIQUE		1	40		
5	NESTED LOOPS		1	40	0 (0)	00:00:01
6	INDEX FULL SCAN	AUTEUR_DE_PK	1	31	0 (0)	00:00:01
* 7	INDEX UNIQUE SCAN	POSSEDE_PK	1	9	0 (0)	00:00:01
* 8	TABLE ACCESS FULL	MUSEE	1	20	3 (0)	00:00:01
* 9	INDEX UNIQUE SCAN	ARTISTE_PK	1	17	0 (0)	00:00:01

Predicate Information (identified by operation id):

```

7 - access("CODEMUSEE"= (SELECT "CODEMUSEE" FROM "MUSEE" "MUSEE" WHERE
"NONMUSEE"='Musee Fabre') AND "CODEO"="CODEO")
8 - filter("NONMUSEE"='Musee Fabre')
9 - access("NOMA"="$nso_col_1" AND "PRENOMA"="$nso_col_2")

```

Statistics

```

1 recursive calls
0 db block gets
12 consistent gets
0 physical reads
0 redo size
572 bytes sent via SQL*Net to client
384 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
4 rows processed

```

1. Dites en le justifiant si cette requête est équivalente à la requête précédente.
2. Donnez votre appréciation sur les deux plans d'exécution proposés. Lequel vous semble le plus efficace ?

3 Index (2 points)

3.1 Construction d'un index de type arbre B+

Construire un arbre B+ avec pour valeurs de clé d'accès, les codes des œuvres (code0), en supposant qu'il y a au plus 2 enregistrements par bloc (ordre 1), et en prenant les enregistrements dans l'ordre donné ci-dessous.

S1, B2, P2, P1, P3, S2

Le placement des valeurs de clé doit respecter l'ordre alphabétique, soit $B1 < B2 < MN1 < P1 < P2 < P3 < S1 < S2 < S3$

4 Architecture Oracle (5 points)

4.1 Éléments mis en jeu

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.

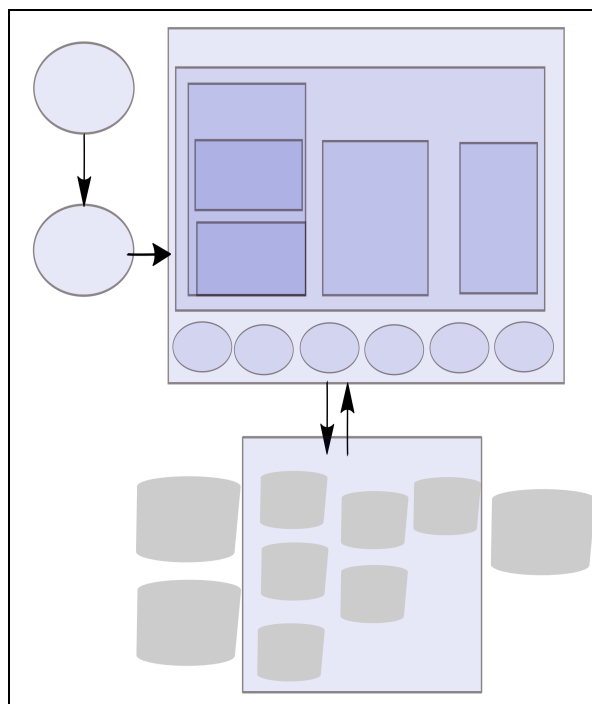


FIGURE 1 – Architecture physique Oracle (structures et processus)



4.1.1 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, dba_data_files

4.2 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

5 Transaction (3 points)

5.1 Questions

1. définissez le rôle général des ordres commit et rollback, et expliquer la notion de commit implicite.
2. quel est le mode d'isolation par défaut exploité par Oracle ?
3. une séquence d'ordres SQL vous est fournie. Quels sont les effets des ordres qui seront rendus permanents au sein de la base de données ?
4. Donnez le nombre de transactions passées dans le cours de cette session.

```
INSERT INTO oeuvre VALUES('P4','Child with dove',1901,null,'tableau');
```

```
ALTER TABLE oeuvre ADD CONSTRAINT domType  
CHECK (type_creation in ('tableau','dessin','sculpture'));
```

```
ROLLBACK;
```

```
INSERT INTO musee VALUES('ARL_2','Musee Reattu','10','Rue du Grand Prieure',  
'peinture et architecture','Arles');
```

```
ROLLBACK;
```

```
UPDATE oeuvre SET lieu_creation = 'France' WHERE codeO = 'P3';
```

```
EXIT;
```

6 Annexe

```
INSERT INTO oeuvre VALUES('ROD_1','Musee Soulages','','Jardin du foirail',  
'arts contemporains','Rodez');  
INSERT INTO musee VALUES('BAR_1','Musee Picasso','15-23','Carrer de Montcada.',  
'arts contemporains','Barcelone');  
INSERT INTO musee VALUES('MPL_1','Musee Fabre','13','rue Montpellieret',  
'arts contemporains','Montpellier');
```

```
INSERT INTO artiste VALUES('Soulages','Pierre','h',1919,null,'France');  
INSERT INTO artiste VALUES('Bioules','Vincent','h',1938,null,'France');  
INSERT INTO artiste VALUES('Bazille','Frederic','h',1841,1870,'France');  
INSERT INTO artiste VALUES('Mc Nicoll','Helen','f',1879,1915,'Canada');  
INSERT INTO artiste VALUES('Picasso','Pablo','h',1881,1973,'Espagne');  
INSERT INTO artiste VALUES('Modigliani','Amedeo','h',1884,1920,'Italie');
```

```

INSERT INTO oeuvre VALUES('S1','Polyptyque I',1986,null,'tableau');
INSERT INTO oeuvre VALUES('S2','Peinture 202 x 143 cm',1967,null,'tableau');
INSERT INTO oeuvre VALUES('S3','Peinture, 55 x 46 cm',1960,null,'tableau');
INSERT INTO oeuvre VALUES('B1','La Place d'Aix',1977,'Aix en P.','tableau');
INSERT INTO oeuvre VALUES('B2','Jeune femmes aux pivoines',1870,null,'tableau');
INSERT INTO oeuvre VALUES('MN1','Sous la tente',1914,'sud France','tableau');
INSERT INTO oeuvre VALUES('P1','Etreinte VIII',1963,null,'dessin');
INSERT INTO oeuvre VALUES('P2','Picador et taureau',1959,null,'sculpture');
INSERT INTO oeuvre VALUES('P3','Mousquetaire assis',1969,null,'tableau');

```

```

INSERT INTO possede VALUES('ROD_1','S1');
INSERT INTO possede VALUES('ROD_1','S2');
INSERT INTO possede VALUES('MPL_1','S3');
INSERT INTO possede VALUES('MPL_1','B1');
INSERT INTO possede VALUES('MPL_1','B2');
INSERT INTO possede VALUES('MPL_1','MN1');

```

```

INSERT INTO auteur_de VALUES('Soulages','Pierre','S1');
INSERT INTO auteur_de VALUES('Soulages','Pierre','S2');
INSERT INTO auteur_de VALUES('Soulages','Pierre','S3');
INSERT INTO auteur_de VALUES('Bioules','Vincent','B1');
INSERT INTO auteur_de VALUES('Bazille','Frederic','B2');
INSERT INTO auteur_de VALUES('Mc Nicoll','Helen','MN1');
INSERT INTO auteur_de VALUES('Picasso','Pablo','P1');
INSERT INTO auteur_de VALUES('Picasso','Pablo','P2');
INSERT INTO auteur_de VALUES('Picasso','Pablo','P3');

```