



Session : 1

Durée de l'épreuve : 2 heures

Date : 11 janvier 2022

Documents autorisés : tous

Mention Informatique

Matériel utilisé : aucun

Master 2^{ème} année : Administration BD (HAI901I)

1 BD exemple : service de vélos en libre accès de la TAM

1.1 Enoncé

Une interprétation textuelle volontairement simplifiée de l'univers du service des déplacements à vélo de l'agglomération de Montpellier. Une station (aussi connue sous le vocable véloparc), est identifiée par un code (codeS) et caractérisée par un nom (nomS), une latitude et une longitude. Une station possède un nombre fixe d'emplacements à vélo qui peuvent être libres ou bien occupés en fonction des allées et venues des usagers. L'agglomération rend disponible en quasi temps réel (toutes les minutes) les informations sur les places libres et les places occupées pour chaque station. Pour des besoins de simplification de l'énoncé, ces informations sont seulement retranscrites à l'échelle de la journée. Une station va donc faire l'objet de différentes occupations au travers du temps. Une occupation est identifiée par le couple (codeS, estampille_temporelle) et est caractérisée par un nombre total de places, un nombre de places libres et un nombre de place occupées. 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.

station(**codeS** varchar(4), nomS varchar(60), latitude float, longitude float)
occupation(**codeS** varchar(4), **estampille_temporelle** date, nbreTotal integer,
nbreLibre integer, nbreOccupe integer)
avec occupation(codeS) ⊆ station(codeS)

2 Optimisation de requêtes et index (9 points)

2.1 Plan d'exécution d'une requête

Une requête SQL vous est donnée, ainsi que le plan d'exécution choisi par l'optimiseur Oracle pour traiter cette requête :

```
set autotrace on
```

```
select s.codeS, nomS from station s join occupation o on s.codeS = o.codeS  
where estampille_temporelle='3/01/2022' and nbreOccupe = 0 ;
```

Plan d'exécution

Plan hash value: 4184542038

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		13	455	6 (17)	00:00:01
1	MERGE JOIN		13	455	6 (17)	00:00:01

	2		TABLE ACCESS BY INDEX ROWID		STATION		57		1140		2	(0)		00:00:01	
	3		INDEX FULL SCAN		ST_PK		57				1	(0)		00:00:01	
*	4		SORT JOIN				13		195		4	(25)		00:00:01	
*	5		TABLE ACCESS FULL		OCCUPATION		13		195		3	(0)		00:00:01	

Predicate Information (identified by operation id):

4 - access("S"."CODES"="0"."CODES")
 filter("S"."CODES"="0"."CODES")
 5 - filter("0"."NBREOCCUPE"=0 AND "0"."ESTAMPILLE_TEMPORELLE"=TO_DATE('2022-01-03 00:00:00', 'syyyy-mm-dd hh24:mi:ss'))

Statistiques

53 recursive calls
 0 db block gets
 36 consistent gets
 11 physical reads
 0 redo size
 851 bytes sent via SQL*Net to client
 460 bytes received via SQL*Net from client
 2 SQL*Net roundtrips to/from client
 4 sorts (memory)
 0 sorts (disk)
 9 rows processed

2.1.1 Série de questions

Une série de questions relatives à l'exécution de la requête, vous est proposée. Vous répondrez à ces questions dans l'ordre.

1. Expliquer la sémantique associée à la requête 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 exploité pour le calcul de la jointure ? Vous expliquerez les raisons qui vous semblent avoir 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. Est ce que le nombre de tuples satisfaisant la requête, estimé par l'optimiseur est correct ?
7. rajouter à la requête, les directives nécessaires à l'utilisation de l'opérateur boucles imbriquées pour la jointure.



8. la requête donnée est une semi-jointure. Vous proposerez en SQL, une requête équivalente qui exploite le prédicat EXISTS.
9. Le plan d'exécution de cette nouvelle requête basée sur EXISTS, est donné. Donnez votre appréciation sur les deux plans d'exécution proposés. L'un vous semble plus efficace, que l'autre ?

Plan d'exécution

Plan hash value: 3697785705

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		12	420	6 (17)	00:00:01
1	MERGE JOIN SEMI		12	420	6 (17)	00:00:01
2	TABLE ACCESS BY INDEX ROWID	STATION	57	1140	2 (0)	00:00:01
3	INDEX FULL SCAN	ST_PK	57		1 (0)	00:00:01
* 4	SORT UNIQUE		13	195	4 (25)	00:00:01
* 5	TABLE ACCESS FULL	OCCUPATION	13	195	3 (0)	00:00:01

Predicate Information (identified by operation id):

- 4 - access("S"."CODES"="0"."CODES")
filter("S"."CODES"="0"."CODES")
- 5 - filter("NBREOCCUPE"=0 AND "ESTAMPILLE_TEMPORELLE"=TO_DATE(' 2022-01-03 00:00:00', 'syyy-mm-dd hh24:mi:ss'))

Statistiques

9 recursive calls
0 db block gets
14 consistent gets
10 physical reads
0 redo size
851 bytes sent via SQL*Net to client
483 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
9 rows processed

3 Index (3 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 stations (codeS), 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.

001, 005, 038, 037, 050, 004

Le placement des valeurs de clé doit respecter l'ordre alphabétique, soit $001 < 004 < 005 < 037 < 038 < 050$

4 Architecture Oracle (4 points)

4.1 Eléments 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 complétez 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 (les processus avant-plan : user process et server process ne sont pas à indiquer) sont listées, vous positionnerez également ces vues sur la figure.

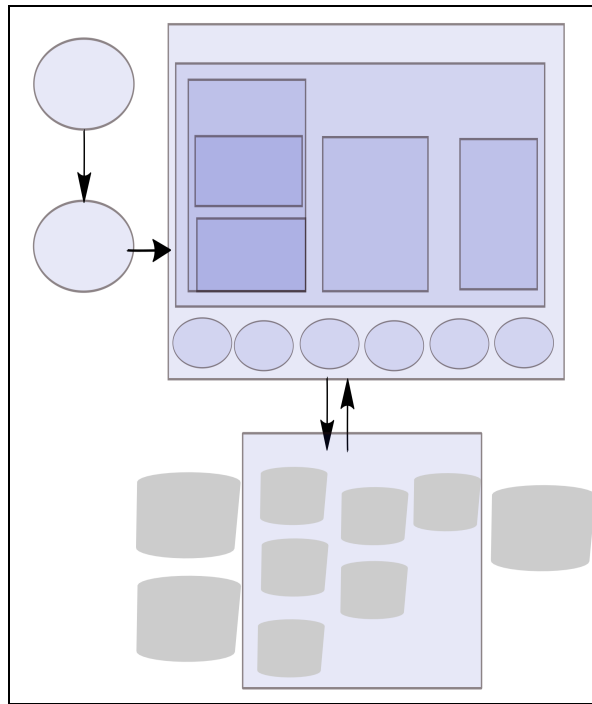


FIGURE 1 – Architecture physique Oracle (structures et processus)

4.1.1 Vues à positionner sur la figure

1. v\$instance, v\$database, v\$bgprocess, v\$sga, v\$sgastat, v\$sql, v\$sqlarea, 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 (4 points)

5.1 Questions

1. définissez le rôle général des ordres commit et rollback, et expliquer la notion de commit implicite.
2. quels sont les deux modes d'isolation exploités par Oracle? donnez en les grands principes
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 occupation values ('036', '04/01/2022',32,28,4);
rollback;
insert into occupation values ('037', '04/01/2022',24,13,11);
create index codeS_idx on occupation(codeS);
rollback;
```

6 Annexe

```
insert into station values ('036','Occitanie',43.634242,3.849128);
insert into occupation values ('036', '02/01/2022',32,23,9);
insert into occupation values ('036','03/01/2022',32,29,3);
insert into station values ('037','FacdesSciences',43.631018,3.860697);
insert into occupation values ('037', '02/01/2022',24,4,20);
insert into occupation values ('037','03/01/2022',24,15,9);
insert into station values ('038','Fac de Lettres',43.630665,3.870230);
insert into occupation values ('038', '02/01/2022',16,16,0);
insert into occupation values ('038','03/01/2022',16,16,0);
insert into station values ('039','Aiguelongue',43.626163,3.882492);
insert into occupation values ('039', '02/01/2022',8,5,3);
insert into occupation values ('039','03/01/2022',8,6,2);
insert into station values ('040','Jeu de Mail des Abbes',43.619701,3.883831);
insert into occupation values ('040', '02/01/2022',8,6,2);
insert into occupation values ('040','03/01/2022',8,6,2);
insert into station values ('041','Euromedecine',43.639119,3.828199);
insert into occupation values ('041', '02/01/2022',8,8,0);
insert into occupation values ('041','03/01/2022',8,4,4);
```