



Session : 1

Durée de l'épreuve : 2 heures

Date : 16 janvier 2024

Documents autorisés : tous

Mention Informatique

Matériel utilisé : aucun

Master 2<sup>ème</sup> année : Administration BD (HAI901I)

## 1 BD exemple : les spectacles et leurs artistes

### 1.1 Enoncé

La base de données considérée, se propose de gérer de manière très partielle les spectacles produits par une grosse compagnie spécialisée dans l'évènementiel. Seuls les artistes, et les spectacles dans lesquels ils se produisent, sont pris en charge par la modélisation. Un Artiste est identifié par un code (codeA), et est caractérisé par un nom (nomA), un prénom (prenomA), un genre (genre), un âge (age) et un pays d'origine (pays). Un Spectacle est identifié par un code (codeS), et est caractérisé par un titre (titre), une localisation (localisation), une date de début (debut) et une date de fin (fin) de mise en production. Un Artiste peut se produire dans un ou plusieurs spectacles (relation seProduitDans), et peut même jouer plusieurs rôles (nomRole) dans le même spectacle. Chaque rôle relève d'une catégorie de rôle (categorieRole) qui permet de savoir si un rôle est au premier plan (valeur A pour categorieRole), au second plan (valeur B pour categorieRole), ou seulement un rôle de figurant (valeur F pour categorieRole). L'attribut categorieRole n'admet donc que ces trois valeurs suivantes : A, B et F.

La base de données va permettre de répondre à des questions telles que :

- quels sont les artistes qui jouent dans le spectacle de "Cendrillon" ?
- quels sont les artistes qui sont partenaires dans le spectacle de "Cendrillon" ?
- Quel est le spectacle qui réunit le plus grand nombre d'artistes ?

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.

- Artiste(**codeA** varchar(8), nomA varchar(16), prenomA varchar(16), age number(3), genre varchar(1), pays varchar(18))
- Spectacle(**codeS** varchar(6), titre varchar(30), localisation varchar(10), debut Date, fin Date)
- seProduitDans(**codeA** varchar(8), **codeS** varchar(6), nomRole varchar(12), categorieRole varchar(1))  
avec seProduitDans(codeA) ⊆ Artiste(codeA)  
et seProduitDans(codeS) ⊆ Spectacle(codeS)

## 2 Optimisation de requêtes et index (10 points)

### 2.1 Plan d'exécution d'une requête

Une requête SQL portant sur le schéma précédent, et son résultat vous sont donnés (les tuples sont en annexe).

```
set autotrace on
```

```
select distinct nomA, prenomA from Artiste A, seProduitDans SPD  
where A.codeA=SPD.codeA and categorieRole='B' ;
```

NOMA	PRENOMA
Dupont	Lea
Berretini	Linda
Dupont	Boris

Le plan d'exécution choisi par l'optimiseur Oracle (plus statistiques) pour traiter cette requête est également fourni.

Plan d'execution

Plan hash value: 832725622

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	136	7 (15)	00:00:01
1	HASH UNIQUE		4	136	7 (15)	00:00:01
* 2	HASH JOIN		4	136	6 (0)	00:00:01
* 3	TABLE ACCESS FULL	SEPRODUITDANS	4	32	3 (0)	00:00:01
4	TABLE ACCESS FULL	ARTISTE	7	182	3 (0)	00:00:01

Predicate Information (identified by operation id):

```
2 - access("A"."CODEA"="SPD"."CODEA")  
3 - filter("CATEGORIEROLE">'B')
```

Statistiques

```
41 recursive calls  
0 db block gets  
46 consistent gets  
0 physical reads  
0 redo size  
724 bytes sent via SQL*Net to client  
52 bytes received via SQL*Net from client  
2 SQL*Net roundtrips to/from client  
0 sorts (memory)  
0 sorts (disk)  
3 rows processed
```

#### 2.1.1 Série de questions

Différentes questions relatives à la requête étudiée et à son exécution, sont posées. 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 décrirez 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. Si au contraire, aucun index n'est mobilisé, expliquez en la raison.
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. 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).
5. Est ce que le nombre de tuples satisfaisant la requête, estimé par l'optimiseur est correct ?
6. Rajouter à la requête, les directives nécessaires à l'utilisation de l'opérateur boucles imbriquées pour la jointure.
7. La requête donnée est une semi-jointure. Vous proposerez en SQL, une requête équivalente qui exploite le prédictat EXISTS.
8. Le plan d'exécution de cette nouvelle requête basée sur EXISTS, est donné. Donnez votre appréciation sur les deux plans d'exécution proposés. Est ce le même plan ou bien deux plans différent. L'un vous semble t'il plus efficace, que l'autre ? Vous pouvez vous aider d'un nouvel arbre algébrique pour justifier vos réponses.
9. Pensez vous q'un index est exploité dans ce nouveau plan d'exécution, et si oui lequel ? Justifiez votre réponse.
10. 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.

Plan d'exécution

Plan hash value: 3533515601

Id   Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0   SELECT STATEMENT		4	136	6 (17)	00:00:01
1   NESTED LOOPS		4	136	6 (17)	00:00:01
2   NESTED LOOPS		4	136	6 (17)	00:00:01
3   SORT UNIQUE		4	32	3 (0)	00:00:01
/* 4   TABLE ACCESS FULL	SEPRODUITDANS	4	32	3 (0)	00:00:01
/* 5   INDEX UNIQUE SCAN	ARTISTE_PK	1		0 (0)	00:00:01
6   TABLE ACCESS BY INDEX ROWID  ARTISTE		1	26	1 (0)	00:00:01

Predicate Information (identified by operation id):

```
4 - filter("CATEGORIEROLE"='B')
5 - access("A"."CODEA"="SPD"."CODEA")
```

Statistiques

```
14 recursive calls
0 db block gets
18 consistent gets
0 physical reads
0 redo size
724 bytes sent via SQL*Net to client
52 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
1 sorts (memory)
0 sorts (disk)
3 rows processed
```

### 3 Index (3 points)

#### 3.1 Construction d'un index de type arbre B+

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

PA\_18, BDD\_23, FS\_24, LPP\_23, BD\_21, C\_20

Le placement des valeurs de clé doit respecter l'ordre alphabétique, soit :

BDD\_23 < BD\_21 < C\_20 < FS\_24 < LPP\_23 < PA\_18

### 4 Architecture Oracle (3 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 compléterez les informations manquantes en ce qui concerne les structures et les processus mis en jeu (les processus avant-plan : user process et server process ne sont pas à indiquer). Les principales vues du métaschéma qui servent à renseigner les différentes structures mémoire et processus d'arrière plan sont listées, vous positionnerez également ces vues sur la figure.

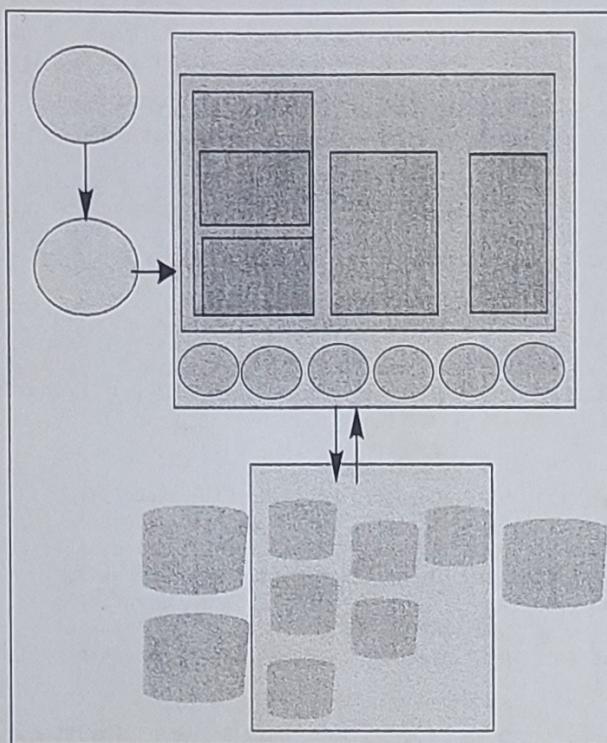


FIGURE 1 – Architecture physique Oracle (structures et processus)

#### 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 ; aidez vous d'un exemple de deux transactions concurrentes
3. Une séquence d'ordres SQL vous est fournie plus bas. Quels sont les effets des ordres qui seront rendus permanents au sein de la base de données (on considère que toutes les équipes référencées ont été au préalable saisies dans la table Equipe) ?
4. Donnez le nombre de transactions passées dans le cours de cette session.

```

X update Artiste set age = age + 1 ;
X insert into Spectacle values ('CB_24','Crin Blanc','ParisLand','20/01/24','20/01/25') ;
✓ create table Compagnie (nomC varchar(20), ville varchar(40), nomResponsable varchar(20)) ;
X insert into Spectacle values ('LB_24','Le Bossu','ParisLand','23/04/24','30/06/25') ;
✓ rollback;
✓ create unique index Compagnie_pk on Compagnie(nomC);
✓ rollback;

```

## 6 Annexe

Un ensemble d'ordres d'insertion de tuples sont fournis à titre d'exemple.

```

prompt insertion des tuples
prompt Artiste
insert into Artiste values ('A1_2023','Dupont','Lea',26,'F','France');
insert into Artiste values ('A2_2023','Dupont','Boris',46,'M','France');
insert into Artiste values ('A3_2023','Botero','Pedro',29,'M','Colombie');
insert into Artiste values ('A4_2023','Diouf','Hilda',22,'F','Senegal');
insert into Artiste values ('A5_2023','Hanani','Kamil',21,'M','Algérie');
insert into Artiste values ('A6_2023','Berretini','Linda',52,'F','Italie');
insert into Artiste values ('A7_2023','Martinez','John',62,'F','Angleterre');

prompt Spectacle
insert into Spectacle values ('BBD_23','La Belle au Bois Dormant','ParisLand','03/11/23','31/12/24');
insert into Spectacle values ('BB_21','La Belle Et la Bete','ParisLand','22/06/21','31/12/22');
insert into Spectacle values ('C_20','Cendrillon','ParisLand','20/04/20','31/12/21');
insert into Spectacle values ('LPP_23','Le Petit Poucet','ParisLand','20/04/23','31/08/24');
insert into Spectacle values ('PA_18','Peau d''Ane','ParisLand','19/02/18','31/08/18');
insert into Spectacle values ('FS_24','Femmes Savantes','ParisLand','19/08/24','31/08/25');

prompt seProduitDans
insert into seProduitDans values ('A1_2023','BBD_23','Aurore','A');
insert into seProduitDans values ('A3_2023','BBD_23','Philippe','A');
insert into seProduitDans values ('A6_2023','BBD_23','Malefique','A');
insert into seProduitDans values ('A6_2023','BB_21','Foule','F');
insert into seProduitDans values ('A6_2023','C_20','Maratre','B');
insert into seProduitDans values ('A4_2023','C_20','Cendrillon','A');
insert into seProduitDans values ('A5_2023','C_20','Le Prince','A');
insert into seProduitDans values ('A1_2023','C_20','Javotte','B');
insert into seProduitDans values ('A2_2023','C_20','Grand Duc','B');
insert into seProduitDans values ('A6_2023','C_20','Anastasie','B');

commit;

```