



Session : 1

Durée de l'épreuve : 2 heures

Date : 14 janvier 2025

Documents autorisés : papier

Mention Informatique

Matériel utilisé : aucun

Master 2<sup>ème</sup> année : Administration BD (HAI9011)

## 1 BD exemple : les prix Nobel

### 1.1 Énoncé

Nous nous intéresserons aux personnalités qui sont distinguées par un prix Nobel, en raison de leurs travaux ou de leurs engagements au bénéfice de tous. Un prix Nobel est décerné chaque année, et depuis 1901, dans les disciplines suivantes : paix ou diplomatie, littérature, chimie, physique, et physiologie ou médecine. Un prix Nobel peut être partagé une même année par plusieurs personnes en raison de leurs travaux communs ou connexes sur une thématique donnée (appartenant aux disciplines sus-citées).

Les personnalités (personne par souci de simplicité) sont identifiées par un numéro (numP) et sont caractérisées par un nom (nomP), un prénom (prenomP), un genre et une date de naissance (dateN). L'attribut genre n'admet donc que deux valeurs : M (pour Masculin), et F (pour Féminin). Un prix Nobel (prix par souci de simplicité) est identifié par un code (codeP), une année (annee) et une discipline (categorie). Un prix distingue (relation distingue) chaque année et pour chaque catégorie, au moins une personne et jusqu'à 4 personnes à la fois, et ce pour la même raison ou bien pour des raisons connexes. Un pays (pays) est identifié par un nom, et est caractérisé par une capitale et une population. Une personne peut posséder une nationalité voire plusieurs nationalités, elle peut être également apatride (sans nationalité). La relation aPrNationalite indique les nationalités de chacune des personnes.

La base de données associée à cet énoncé, va permettre de répondre à des questions telles que :

- quels sont les personnes qui ont reçu un prix Nobel d'économie en 2019 ?
- quels sont les pays pour lesquels toutes les catégories de prix Nobel sont représentées (qui ont des ressortissants "Nobélisés" dans toutes les catégories) ?
- quelle est la catégorie de prix Nobel ayant récompensé le plus de femmes ?
- quels sont les Français.es qui ont reçu un prix Nobel pour leurs travaux concernant la lutte contre la pauvreté ?

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.

- personne(**numP** integer, nomP varchar(20), prenomP varchar(20), genre char(1), dateN date)
- prix(**codeP** varchar(12), annee integer, categorie varchar(14))
- pays(**nom** varchar(20), capitale varchar(20), population integer)
- distingue(**codeP** varchar(12), **numP** integer, raison varchar(50))  
avec distingue(codeP)  $\subseteq$  Prix(codeP)

- et  $\text{distingue}(\text{numP}) \subseteq \text{Personne}(\text{numP})$
- $\text{aPrNationalite}(\text{numP integer, nom varchar(20)})$   
avec  $\text{aPrNationalite}(\text{numP}) \subseteq \text{Personne}(\text{numP})$   
et  $\text{aPrNationalite}(\text{nom}) \subseteq \text{Pays}(\text{nom})$

## 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 de l'instance de base de données sont en annexe).

set autotrace on

```
select nomP, prenomP from personne ps join distingue d
on ps.numP = d.numP join prix px
on d.codeP = px.codeP where categorie = 'Economie' ;
```

| NOMP     | PRENOMP |
|----------|---------|
| Banerjee | Abhijit |
| Duflo    | Esther  |
| Kremer   | Michael |

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

Plan d'exécution

Plan hash value: 1121308535

| Id  | Operation                   | Name     | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-----------------------------|----------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT            |          | 1    | 75    | 0 (0)       | 00:00:01 |
| 1   | NESTED LOOPS                |          | 1    | 75    | 0 (0)       | 00:00:01 |
| 2   | NESTED LOOPS                |          | 1    | 58    | 0 (0)       | 00:00:01 |
| 3   | INDEX FULL SCAN             | DISTPK   | 1    | 21    | 0 (0)       | 00:00:01 |
| 4   | TABLE ACCESS BY INDEX ROWID | PERSONNE | 1    | 37    | 0 (0)       | 00:00:01 |
| * 5 | INDEX UNIQUE SCAN           | PERSPK   | 1    |       | 0 (0)       | 00:00:01 |
| * 6 | TABLE ACCESS BY INDEX ROWID | PRIX     | 1    | 17    | 0 (0)       | 00:00:01 |
| * 7 | INDEX UNIQUE SCAN           | PRIXPK   | 1    |       | 0 (0)       | 00:00:01 |

Predicate Information (identified by operation id):

- 5 - access("D"."NUMP"="PS"."NUMP")
- 6 - filter("PX"."CATEGORIE"='Economie')
- 7 - access("D"."CODEP"="PX"."CODEP")

Statistics

```
1 recursive calls
0 db block gets
24 consistent gets←
0 physical reads
0 redo size
547 bytes sent via SQL*Net to client
384 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 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. Si au contraire, aucun index n'est mobilisé, expliquez en la raison.
- ✓ 3. Quels sont les opérateurs physiques exploités pour le calcul des jointures ? Vous expliquerez les raisons qui vous semblent avoir poussé l'optimiseur à faire ces choix d'algorithmes/opérateurs.
- ✓ 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 "tri fusion" (sort merge) pour la jointure entre les tables personne et distingue.
7. La requête donnée est une semi-jointure. Vous proposerez en SQL, une requête équivalente qui exploite le prédicat EXISTS par deux fois.
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 que des index soient exploités dans ce nouveau plan d'exécution, et si oui lesquels ? 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.

#### Execution Plan

Plan hash value: 3103407001

| Id  | Operation                   | Name     | Rows | Bytes | Cost (%CPU) | Time     |
|-----|-----------------------------|----------|------|-------|-------------|----------|
| 0   | SELECT STATEMENT            |          | 1    | 50    | 1 (100)     | 00:00:01 |
| 1   | MERGE JOIN SEMI             |          | 1    | 50    | 1 (100)     | 00:00:01 |
| 2   | TABLE ACCESS BY INDEX ROWID | PERSONNE | 1    | 37    | 0 (0)       | 00:00:01 |
| 3   | INDEX FULL SCAN             | PERSPK   | 1    |       | 0 (0)       | 00:00:01 |
| * 4 | SORT UNIQUE                 |          | 1    | 13    | 1 (100)     | 00:00:01 |
| 5   | VIEW                        | VW_SQ_1  | 1    | 13    | 0 (0)       | 00:00:01 |
| 6   | NESTED LOOPS SEMI           |          | 1    | 38    | 0 (0)       | 00:00:01 |
| 7   | INDEX FULL SCAN             | DISTPK   | 1    | 21    | 0 (0)       | 00:00:01 |

8

Table access

Prix

|   |   |  |                             |  |        |  |   |  |    |  |   |     |  |          |  |
|---|---|--|-----------------------------|--|--------|--|---|--|----|--|---|-----|--|----------|--|
| * | 8 |  | TABLE ACCESS BY INDEX ROWID |  | PRIX   |  | 1 |  | 17 |  | 0 | (0) |  | 00:00:01 |  |
| * | 9 |  | INDEX UNIQUE SCAN           |  | PRIXPK |  | 1 |  |    |  | 0 | (0) |  | 00:00:01 |  |

-----

Predicate Information (identified by operation id):

-----

```

4 - access("PS"."NUMP"="NUMP")
    filter("PS"."NUMP"="NUMP")
8 - filter("CATEGORIE"='Economie')
9 - access("D"."CODEP"="PX"."CODEP")

```

Statistics

-----

```

1 recursive calls
0 db block gets
12 consistent gets
0 physical reads
0 redo size
547 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)
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 prix Nobel (codeP), 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.

nEco\_2020, nChim\_2003, nLitt\_2016, nEco\_2019, nPeace\_2018, nPeace\_2002

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

nChim\_2003 < nEco\_2019 < nEco\_2020 < nLitt\_2016 < nPeace\_2002 < nPeace\_2018

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



^ update pays set population = population + 102 where nom='France' ;  
 2 insert into pays values ('Belgique','Bruxelles',11492641);  
 3 alter table personne add constraint domGenre check (genre in ('F','M')) ;  
 insert into pays values ('Luxembourg','Luxembourg',645397);  
 rollback;  
 create index idx\_codeP on distingue(codeP);  
 rollback;

## 6 Annexe

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

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';
```

```

insert into personne values (1, 'Van't Hoff','Jacobus Henricus','M','1852-08-30');
insert into personne values (1026,'Mukwege','Denis','M','1955-03-01');
insert into personne values (1027,'Murad','Nadia','F','1993-03-10');
insert into personne values (1035,'Banerjee','Abhijit','M','1961-02-21');
insert into personne values (1036,'Duflo','Esther','F','1972-10-25');
insert into personne values (1037,'Kremer','Michael','M','1964-11-12');
insert into personne values (770,'Carter','Jimmy','M','1924-10-14');
insert into personne values (964,'Dylan','Bob','M','1941-05-24');
insert into personne values (775,'MacKinnon','Roderick','M','1956-02-19');

```

```

insert into prix values('nPeace_2002',2002,'Paix');
insert into prix values('nPeace_2018',2018,'Paix');
insert into prix values('nEco_2019',2019,'Economie');
insert into prix values('nLitt_2016',2016,'Litterature');
insert into prix values('nEco_2020',2020,'Economie');
insert into prix values('nChim_2003',2003,'Chimie');

```

```

insert into distingue values ('nPeace_2018',1026,'droits de l''Homme');
insert into distingue values ('nPeace_2018',1027,'droits de l''Homme');
insert into distingue values ('nEco_2019',1036,'lutte contre la pauvreté');
insert into distingue values ('nEco_2019',1035,'lutte contre la pauvreté');
insert into distingue values ('nEco_2019',1037,'lutte contre la pauvreté');
insert into distingue values ('nPeace_2002',770,'diplomatie');
insert into distingue values ('nLitt_2016',964,'paroles chanson');
insert into distingue values ('nChim_2003',775,'canaux ioniques');

```

```

insert into pays values ('Rep. Dem. Congo','Kinshasa',86026000);
insert into pays values ('Irak','Baghdad',38124182);
insert into pays values ('France','Paris',64300821);
insert into pays values ('Espagne','Madrid',47400798);
insert into pays values ('USA','Washington',331449281);
insert into pays values ('India','Delhi',1210854977);
insert into pays values ('Grande Bretagne','Londres',66980000);

```

```

insert into aPrNationalite values (1026,'Rep. Dem. Congo');
insert into aPrNationalite values (1036,'France');
insert into aPrNationalite values (1036,'USA');
insert into aPrNationalite values (1035,'USA');
insert into aPrNationalite values (1035,'India');
insert into aPrNationalite values (1037,'USA');
insert into aPrNationalite values (770,'USA');
insert into aPrNationalite values (1027,'Irak');
insert into aPrNationalite values (964,'USA');
insert into aPrNationalite values (775,'USA');

```

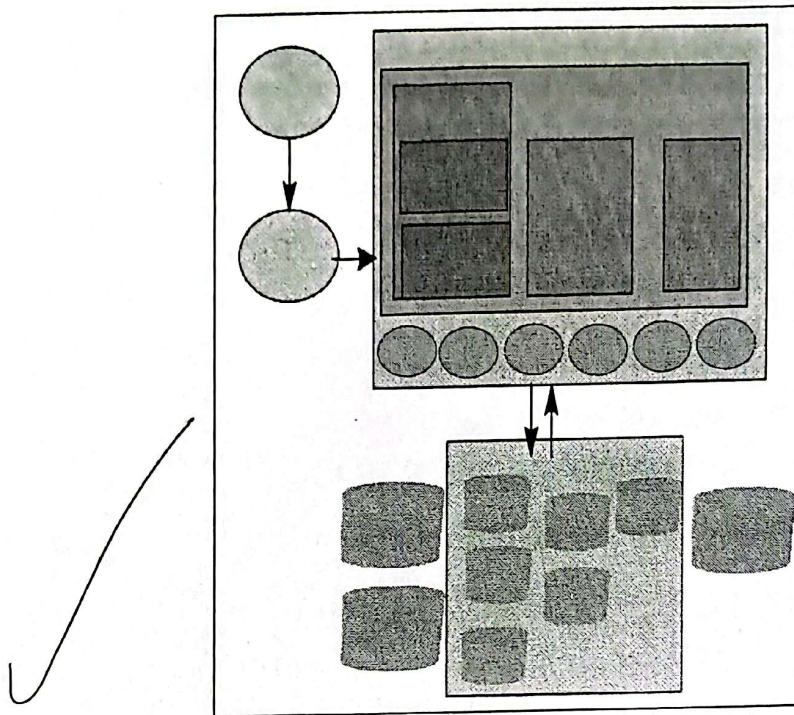


FIGURE 1 – Architecture physique Oracle (structures et processus)

#### 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?
4. Donnez, en le justifiant, le nombre de transactions passées dans le cours de cette session.