

1 BD exemple : le monde estudiantin

1.1 Enoncé

La base de données considérée, gère les étudiants inscrits dans des formations. Ces étudiants en outre nouent des relations d'amitié. Cette gestion est très partielle, et porte sur les types d'entité suivants. Un type d'entité **Etudiant** est identifié par un numéro INE, et est caractérisé par un nom, un prénom, une date de naissance et un genre qui prend ses valeurs parmi l'ensemble {homme, femme}. Une **Formation** est identifiée par un code de formation et est caractérisée par un intitulé, un niveau et un nom de responsable. Un **Etudiant** est inscrit à une **Formation** pour une année donnée. De même un **Etudiant** apprécie d'autres étudiants qui peuvent à leur tour l'apprécier ou ne pas l'apprécier.

1.2 Enoncé 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.

- Etudiant(**numINE** varchar(10), nom varchar(20), prenom varchar(15), dateNaissance date, genre varchar(5))
- Formation(**codeF** varchar(9), nomF varchar(20), niveau varchar(3), nomResponsable varchar(20))
- InscritDans(**numINE** varchar(10), **codeF** varchar(9), **anneeInscription** number(4))
avec InscritDans(numINE) \subseteq Etudiant(numINE)
avec InscritDans(codeF) \subseteq Formation(codeF)
- Apprecie(**numEtudiant** varchar(10), **numAmi** varchar(10))
avec Apprecie(numEtudiant) \subseteq Etudiant(numINE)
avec Apprecie(numAmi) \subseteq Etudiant(numINE)

2 Exercice optimisation

2.1 Index (4 points)

Vous répondrez (avec une justification courte) aux questions suivantes :

1. Quels sont les index déjà présents au sein du schéma de la base de données, et sur quels attributs, sont ils implicitement posés ?

Il y a exactement quatre présents au sein du schéma de la base de données. Nommons les Etudiant_PK, Formation_PK, InscritDans_PK et Apprecie_PK. Ils sont respectivement posés sur Marque_PK est posé sur l'attribut numINE, sur l'attribut codeF, sur le triplet d'attributs (numINE, codeF, anneeInscription), et sur le couple d'attributs (numEtudiant, numAmi).

2. Sur quels autres attributs, poseriez vous d'autres index ? Donner un exemple d'ordre de création d'index correspondant.

Nous pourrions poser des index sur les attributs portant une contrainte de clé étrangère.

```
CREATE INDEX InscritDans_FK_Etudiant
ON InscritDans (numINE);

CREATE INDEX InscritDans_FK_Formation
ON InscritDans (codeF);

CREATE INDEX Apprecie_FK_Etudiant_numEtudiant
ON Apprecie(numEtudiant);

CREATE INDEX Apprecie_FK_Etudiant_numAmi
ON Apprecie (numAmi);
```

3. Expliquez les ordres suivants et les résultats obtenus. Quel est le type d'index le plus fréquemment utilisé sous Oracle ? Précisez en les intérêts.

```
analyze index etudiant_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
ETUDIANT_PK	7996	1	6	0	1

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

Expliquez la sémantique associée à la requête suivante et décrivez le plan d'exécution obtenu.

```
EXPLAIN PLAN FOR
SELECT /*+ use_merge(e i) */ nom, prenom
FROM etudiant e, inscritDans i
WHERE anneeInscription = '2015' and e.numINE = i.numINE;
```

```
SELECT plan_table_output
FROM table (dbms_xplan.display()) ;
```

PLAN_TABLE_OUTPUT

Plan hash value: 2806399230

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
1						

```

-----
| 0 | SELECT STATEMENT | | 6 | 192 | 4 (25) |
00:00:01 |
| 1 | MERGE JOIN | | 6 | 192 | 4 (25) |
00:00:01 |
| 2 | TABLE ACCESS BY INDEX ROWID | ETUDIANT | 6 | 126 | 2 (0) |
00:00:01 |
| 3 | INDEX FULL SCAN | ETUDIANT_PK | 6 | | 1 (0) |
00:00:01 |
| * 4 | SORT JOIN | | 6 | 66 | 2 (50) |
00:00:01 |
| * 5 | INDEX FULL SCAN | INSCRITDANS_PK | 6 | 66 | 1 (0) |
00:00:01 |
-----

```

PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

- ```

4 - access("E"."NUMINE"="I"."NUMINE")
 filter("E"."NUMINE"="I"."NUMINE")
5 - access("ANNEEINSCRIPTION"=2015)
 filter("ANNEEINSCRIPTION"=2015)

```

EXPLAIN PLAN FOR permet de calculer le plan d'exécution de la requête qui suit. La deuxième requête permet d'afficher le plan d'exécution précédemment calculé. Ici la requête pour laquelle nous calculons le plan d'exécution consiste à afficher les étudiants inscrits en 2015. Le /\*+ use\_merge(e, i) \*/ dans cette dernière impose un SORT MERGE JOIN : chaque table est triée sur les attributs impliqués dans la condition de jointure. Une fois le tri réalisé, l'opération de jointure par fusion peut alors être effectuée.

Les index sur numE sont triés, la table etudiant est parcouru en fonction du tri effectué sur sa clé primaire. La jointure a lieu (merge join) puis on recupere le resultat.

## 2.3 Différentes écritures d'une même requête (3 points)

Vous écrirez de deux manières différentes, la requête : donnez les étudiants qui s'apprécient mutuellement (couples d'amis). Vous en donnerez les expressions algébriques et vous formulerez votre avis sur la requête qui vous semble la plus efficace.

```

SELECT DISTINCT e1.prenom, e1.nom, e2.prenom, e2.nom
FROM Apprecie a, Etudiant e1, Etudiant e2
WHERE a.numEtudiant = e1.numINE
 AND a.numAmi = e2.numINE
INTERSECT
SELECT DISTINCT e1.prenom, e1.nom, e2.prenom, e2.nom
FROM Apprecie a, Etudiant e1, Etudiant e2

```

```
WHERE a.numEtudiant = e2.numINE
AND a.numAmi = e1.numINE;
```

```
SELECT DISTINCT e1.prenom, e1.nom, e2.prenom, e2.nom
FROM Apprecie a1, Apprecie a2, Etudiant e1, Etudiant e2
WHERE a1.numEtudiant = e1.numINE
AND a1.numAmi = a2.numEtudiant
AND a2.numAmi = e2.numINE
AND a1.numEtudiant = a2.numAmi;
```

## 3 Architecture Oracle

### 3.1 Éléments mis en jeu (4 points)

Une figure incomplète de l'organisation physique d'une base de données Oracle vous est donnée (figure 1 en fin de sujet à rendre avec la 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 (ou autres processus) sont également données, 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\$sgainfo, v\$sgastat, v\$sql, v\$datafile, v\$logfile

## 4 Question surcouche procédurale PL/SQL (6 points)

Vous donnerez l'écriture d'une procédure nommée nbreLikers qui renvoie pour un étudiant donné dont le numéro INE est passé en paramètre d'entrée, le nombre d'étudiants qui l'apprécie. Vous proposerez une version non sécurisée et une version sécurisée de cette procédure.

## 5 Annexe

```
CREATE TABLE Etudiant(
 numINE varchar(10),
 nom varchar(20),
```

```

 prenom varchar(15),
 dateNaissance date,
 genre varchar(5),
 CONSTRAINT Etudiant_PK PRIMARY KEY (numINE)
);

CREATE TABLE Formation(
 codeF varchar(9),
 nomF varchar(20),
 niveau varchar(3),
 nomResponsable varchar(20),
 CONSTRAINT Formation_PK PRIMARY KEY (codeF)
);

CREATE TABLE InscritDans(
 numINE varchar(10),
 codeF varchar(9),
 anneeInscription number(4),
 CONSTRAINT InscritDans_PK PRIMARY KEY (numINE, codeF,
anneeInscription),
 CONSTRAINT InscritDans_FK_Etudiant FOREIGN KEY (numINE) REFERENCES
Etudiant (numINE),
 CONSTRAINT InscritDans_FK_Formation FOREIGN KEY (codeF) REFERENCES
Formation (codeF)
);

CREATE TABLE Apprecie(
 numEtudiant varchar(10),
 numAmi varchar(10),
 CONSTRAINT Apprecie_PK PRIMARY KEY (numEtudiant, numAmi),
 CONSTRAINT Apprecie_FK_Etudiant_numEtu FOREIGN KEY (numEtudiant)
REFERENCES Etudiant (numINE),
 CONSTRAINT Apprecie_FK_Etudiant_numAmi FOREIGN KEY (numAmi)
REFERENCES Etudiant (numINE)
);

INSERT INTO Formation VALUES ('HM1IN-604', 'IPS', 'M1', 'K. T. ');
INSERT INTO Formation VALUES ('HM1SN-601', 'BCD', 'M1', 'A. F-L');
INSERT INTO Formation VALUES ('HM1PH-602', 'Physique-Numerique', 'M1',
'D. C. ');
INSERT INTO Formation VALUES ('HM1BE-611', 'EPI', 'M1', 'C. M. ');
INSERT INTO Formation VALUES ('HM1GE-600', 'Geomatique', 'M1', 'C. G. ');
INSERT INTO Formation VALUES ('HM1BE-599', 'STIC-Eco', 'M1', 'C. M. ');

INSERT INTO Etudiant VALUES ('20101234', 'Martin', 'Paul', '20-aug-1990',
'homme');
INSERT INTO Etudiant VALUES ('20102345', 'Martin', 'Marie',
'19-apr-1990', 'femme');
INSERT INTO Etudiant VALUES ('20112345', 'Martinez', 'Mathilde',
'19-jun-1991', 'femme');
INSERT INTO Etudiant VALUES ('20111234', 'Martinetti', 'Paul',
'2-aug-1989', 'homme');
INSERT INTO Etudiant VALUES ('20113456', 'Chadi', 'Amina', '8-mar-1990',
'femme');
INSERT INTO Etudiant VALUES ('20123456', 'Faye', 'Moussa', '1-mar-1991',
'homme');

```

```
INSERT INTO InscritDans VALUES ('20101234', 'HM1IN-604', '2015');
INSERT INTO InscritDans VALUES ('20112345', 'HM1IN-604', '2015');
INSERT INTO InscritDans VALUES ('20102345', 'HM1SN-601', '2015');
INSERT INTO InscritDans VALUES ('20112345', 'HM1SN-601', '2015');
INSERT INTO InscritDans VALUES ('20113456', 'HM1BE-599', '2015');
INSERT INTO InscritDans VALUES ('20123456', 'HM1BE-599', '2015');

INSERT INTO Apprecie VALUES ('20101234', '20112345');
INSERT INTO Apprecie VALUES ('20112345', '20101234');
INSERT INTO Apprecie VALUES ('20101234', '20102345');
INSERT INTO Apprecie VALUES ('20112345', '20102345');
INSERT INTO Apprecie VALUES ('20123456', '20102345');
INSERT INTO Apprecie VALUES ('20123456', '20101234');
INSERT INTO Apprecie VALUES ('20102345', '20101234');
```

Figure 1 – Architecture physique Oracle (structures et processus)