



Département Informatique
ADIA-IISE/2023-2024

Module Bases de données et modélisation
TP 3

Travaux Pratiques 3
Langage de manipulation des données : Requêtes avancées

Réaliser par :
TARZOUKT Abderrahim

Encadré par :
M.OUKDACH Yassine

1. Ajouter le champ `date_naissance` à la table `etudiant` et y insérer des valeurs en utilisant le navigateur ou des opérations `update`.

```
SQL> Alter table etudiant  
2 add date_naissance date ;  
Table altered.
```

2. Afficher le nom, le prénom et la date de naissance (le nom du jour, celui du mois et l'année) des étudiants.

```
SQL> SELECT nometudiant,
2          prenometudiant,
3          EXTRACT(day FROM date_naissance) AS jour,
4          EXTRACT(month FROM date_naissance) AS mois,
5          EXTRACT(year FROM date_naissance) AS année
6 FROM Etudiant;
```

NOMETUDIANT	PRENOMETUDIANT	JOUR	MOIS	ANNÉE
Dupont	Jean			
Martin	Sophie			
Lefevre	Pierre			
Leclerc	Marie			
Dubois	Alice			

3. Calculer de deux façons différentes l'âge moyen des étudiants.

```
SQL> select avg(sysdate-date_naissance)/365 as age_moyen
2 from Etudiant ;
```

```
AGE_MOYEN
-----
```

4. Afficher le nombre d'étudiants de l'établissement et le moyen de leurs âges.

```
SQL> SELECT COUNT(*) AS nombre_etudiants,
2          AVG(MONTHS_BETWEEN(SYSDATE, date_naissance) / 12) AS age_moyen
3 FROM Etudiant;
```

NOMBRE_ETUDIANTS	AGE_MOYEN
5	

5. Afficher le nombre d'étudiants de l'établissement, le minimum, le maximum et la moyenne de leurs âges.

```
SQL> SELECT COUNT(*) AS nombre_etudiants,
2          MIN(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM date_naissance)) AS age_minimum,
3          MAX(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM date_naissance)) AS age_maximum,
4          AVG(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM date_naissance)) AS age_moyen
5 FROM Etudiant;
```

NOMBRE_ETUDIANTS	AGE_MINIMUM	AGE_MAXIMUM	AGE_MOYEN
5			

6. Lister par année le nombre d'étudiants, la moyenne, le maximum et le minimum des notes.

```
SQL> SELECT EXTRACT(YEAR FROM e.date_naissance) AS annee,
2          COUNT(r.codeetudiant) AS nombre_etudiants,
3          AVG(r.note) AS moyenne_notes,
4          MAX(r.note) AS note_maximale,
5          MIN(r.note) AS note_minimale
6 FROM Etudiant e
7 JOIN Resultat r ON e.codeetudiant = r.codeetudiant
8 GROUP BY EXTRACT(YEAR FROM e.date_naissance);
```

ANNEE	NOMBRE_ETUDIANTS	MOYENNE_NOTES	NOTE_MAXIMALE	NOTE_MINIMALE
4	69.8725	98.12	37.49	

7. Lister par année le nombre d'étudiants de l'année 1 et 2, ainsi que la moyenne, le maximum et le minimum de leurs notes.

```
SQL> Select count (distinct codeetudiant),avg (note),min(note),max(note),extract(year from datenaissance)
2 from Resultat natural join Etudiant
3 where extract(year from datenaissance) in(1,2)
4 group by extract(year from datenaissance) ;
```

no rows selected

8. Chercher par année le nombre d'étudiants des villes Agadir et Casa, ainsi que la moyenne, le maximum et le minimum de leurs notes :

```
SQL> select count(distinct codeetudiant) , avg (note) , min(note) , max(note) , extract(year from datenaissance)
2 from resultat natural join etudiant
3 where ville in('agadir' , 'casa')
4 group by extract (year from datenaissance) ;
```

COUNT(DISTINCTCODEETUDIANT)	AUG(NOTE)	MIN(NOTE)	MAX(NOTE)
1	37.49	37.49	37.49
1988			

9.Lister par année le nombre d'étudiants de l'année 1 et 2, venant de Marrakech ou Taroudant, ainsi que la moyenne, le maximum et le minimum de leurs notes :

```
SQL> select count(distinct codeetudiant),avg (note),min(note),max(note),extract(year from datenaissance)
2 from resultat natural join Etudiant
3 where ville in ('MARRAKECH','TAROUDANT')
4 and extract(year from datenaissance) in(1,2)
5 group by extract(year from datenaissance) ;
```

no rows selected

10. Lister par année le nombre d'étudiants, la moyenne, le maximum et le minimum de leurs notes, tels que la moyenne soit comprise entre 12 et 16 :

```
SQL> select count(distinct codeetudiant), avg (note), min(note), max(note), EXTRACT(YEAR FROM datenaissance)
2   from Resultat natural join Etudiant
3   group by EXTRACT(YEAR FROM datenaissance)
4   having avg(note) between 12 and 16;

no rows selected
```

11. Lister par année le nombre d'étudiants, la moyenne, le maximum et le minimum de leurs notes, ordonnés par moyenne des notes :

```
SQL> Select count(distinct codeetudiant), avg (note), min(note), max(note), extract(year from datenaissance )
2   from Resultat natural join Etudiant
3   group by extract(year from datenaissance )
4   order by avg(note) ;
```

COUNT(DISTINCTCODEETUDIANT)	AUG(NOTE)	MIN(NOTE)	MAX(NOTE)
1	37.49	37.49	37.49
1988			
1	49.61	49.61	49.61
1992			
1	94.27	94.27	94.27
1998			

COUNT(DISTINCTCODEETUDIANT)	AUG(NOTE)	MIN(NOTE)	MAX(NOTE)
1	98.12	98.12	98.12
1995			

12. Lister par année le nombre d'étudiants, la moyenne, le maximum et le minimum de leurs notes, ordonnés par nombre d'étudiants et la moyenne des notes descendantes :

```
SQL> Select count(distinct codeetudiant),avg (note),min(note),max(note),extract(year from datenaissance)
2  From Resultat natural join Etudiant
3  group by extract(year from datenaissance )
4  order by 1, 2 desc;
```

COUNT(DISTINCTCODEETUDIANT)	AUG(NOTE)	MIN(NOTE)	MAX(NOTE)
1	98.12	98.12	98.12
1995			
1	94.27	94.27	94.27
1990			
1	49.61	49.61	49.61
1992			

COUNT(DISTINCTCODEETUDIANT)	AUG(NOTE)	MIN(NOTE)	MAX(NOTE)
1	37.49	37.49	37.49
1988			

13. Donner la liste des enseignants dont le nom contient 'i' à la fin :

```
SQL> SELECT INITCAP(nomenseignant)
2  FROM Enseignant
3  WHERE nomenseignant LIKE '%i';
```

no rows selected

14. Donner les noms des enseignants qui assurent plus de 2 cours :

```
SQL> select nomenseignant,count(codecours)
2  from Enseignant natural join Charge
3  group by codeenseignant, nomenseignant
4  having count(codecours)>=2;
```

NOMENSEIGANT	COUNT(CODECOURS)
Dupont	2

15. Afficher les noms et les spécialités des enseignant dont les spécialités sont connues :

```
SQL> select nomenseignant, specialite
  2  from Enseignant
  3  where specialite is not null;
```

NOMENSEIGANT	SPECIALITE
Martin	Mathématiques
Dupont	Physique
Lefevre	Informatique
Lefevre	Informatique
Lefevre	Informatique

16. Lister les noms des enseignants ayant le même spécialité :

```
SQL> Select e1.nomenseignant, e2.specialite
  2  from Enseignant e1, Enseignant e2
  3  where e1.specialite = e2.specialite
  4  and e1.codeenseignant < e2.codeenseignant;
```

NOMENSEIGANT	SPECIALITE
Lefevre	Informatique
Lefevre	Informatique
Lefevre	Informatique

17. Lister les noms des enseignants qui font le même cours :

```
SQL> Select e1.nomenseignant, e2.nomenseignant
  2  from Enseignant e1, Enseignant e2
  3  where (e1. codeenseignant, e2.codeenseignant) in (
  4  select c1.codeenseignant, c2.codeenseignant
  5  from Charge c1, Charge c2
  6  where c1.codecours=c2.codecours) ;
```

NOMENSEIGANT	NOMENSEIGANT
Lefevre	Lefevre
Martin	Martin
Dupont	Dupont

18. Créer une table Etudiants semblable à Etudiant et dont les tuples sont différents de ceux d'Etudiant, puis faire l'union :

```

SQL> CREATE TABLE Etudiants (
2     num_etu NUMBER PRIMARY KEY,
3     nom VARCHAR2(20),
4     prénom VARCHAR2(20),
5     année NUMBER(1),
6     genre CHAR CONSTRAINT plage_genre CHECK (genre IN ('F', 'f', 'M', 'm')),
7     ville VARCHAR2(10)
8 );

```

Table created.

```

SQL> ALTER TABLE etudiants
2  MODIFY codeetudiant VARCHAR2(10);

```

Table altered.

```

SQL> INSERT INTO Etudiants (CODEETUDIANT, NOMETUDIANT, PRENOMETUDIANT, DATENAISSANCE, VILLE, DATE_NA
ISSANCE)
2  VALUES (1, 'amine', 'wafi', TO_DATE('2000-01-15', 'YYYY-MM-DD'), 'Paris', TO_DATE('2000-01-15',
'YYYY-MM-DD'));

```

1 row created.

```

SQL>
SQL> INSERT INTO Etudiants (CODEETUDIANT, NOMETUDIANT, PRENOMETUDIANT, DATENAISSANCE, VILLE, DATE_NA
ISSANCE)
2  VALUES (2, 'Smith', 'Alice', TO_DATE('1998-05-20', 'YYYY-MM-DD'), 'Marseille', TO_DATE('1998-05
-20', 'YYYY-MM-DD'));

```

1 row created.

```

SQL> INSERT INTO Etudiants (CODEETUDIANT, NOMETUDIANT, PRENOMETUDIANT, DATENAISSANCE, VILLE, DATE_NA
ISSANCE)
2  VALUES (3, 'yassin', 'loua', TO_DATE('2001-01-15', 'YYYY-MM-DD'), 'niger', TO_DATE('2001-01-15'
, 'YYYY-MM-DD'));

```

1 row created.

```

SQL>
SQL> INSERT INTO Etudiants (CODEETUDIANT, NOMETUDIANT, PRENOMETUDIANT, DATENAISSANCE, VILLE, DATE_NA
ISSANCE)
2  VALUES (4, 'Halima', 'Nirmine', TO_DATE('1999-05-20', 'YYYY-MM-DD'), 'leon', TO_DATE('1999-05-2
0', 'YYYY-MM-DD'));

```

1 row created.

```
SQL> SELECT * FROM Etudiant
2 UNION
3 SELECT * FROM Etudiants;
```

CODEETUDIA	NOMETUDIANT	PRENOMETUDIANT	DATENAISS
VILLE	DATE_NAIS		
1 Paris	amine 15-JAN-00	wafi	15-JAN-00
2 Marseille	Smith 20-MAY-98	Alice	20-MAY-98
3 niger	yassin 15-JAN-01	loua	15-JAN-01

CODEETUDIA	NOMETUDIANT	PRENOMETUDIANT	DATENAISS
VILLE	DATE_NAIS		
4 leon	Halima 20-MAY-99	Nirmine	20-MAY-99
E001 Agadir	Dupont	Jean	15-JAN-90
E002 Marrakech	Martin	Sophie	22-MAY-92

CODEETUDIA	NOMETUDIANT	PRENOMETUDIANT	DATENAISS
VILLE	DATE_NAIS		
E003 casa	Lefevre	Pierre	03-NOV-88
E004 fes	Leclerc	Marie	18-JUL-95
E005 Errachidia	Dubois	Alice	28-SEP-93

19. Faire l'union d'Etudiant et Etudiants pour l'année 1 :


```
SQL> Select * from Etudiant Where extract(year from datenaissance)=2001
2 Union
3 Select * from Etudiants Where extract(year from datenaissance) = 2001;
```

CODEETUDIA	NOMETUDIAN	PRENOMETUDIAN	DATENAISS
VILLE	DATE_NAIS		
3	yassin	loua	15-JAN-01
niger	15-JAN-01		

20. Faire l'union de : les étudiants de première année de Etudiant » et « les étudiants de troisième année de Etudiants :

```
SQL> Select * from Etudiant Where extract(year from datenaissance)=2001
2 Union
3 Select * from Etudiants Where extract(year from datenaissance) =1998;
```

CODEETUDIA	NOMETUDIAN	PRENOMETUDIAN	DATENAISS
VILLE	DATE_NAIS		
2	Smith	Alice	20-MAY-98
Marseille	20-MAY-98		

21. Donner le nom, la moyenne, le minimum, le maximum des notes de chaque étudiant :

```
SQL> Select nometudiant, min(note), max(note), avg(note)
2 from Etudiant natural join Resultat
3 group by codeetudiant ,nometudiant ;
```

NOMETUDIAN	MIN(NOTE)	MAX(NOTE)	AVG(NOTE)
Lefevre	37.49	37.49	37.49
Leclerc	98.12	98.12	98.12
Dupont	94.27	94.27	94.27
Martin	49.61	49.61	49.61

22. Donner le nom, le moyenne, le minimum, le maximum des notes de chaque étudiant de la première année:

```
SQL> Select nometudiant,Min(note), Max(note),Avg(note)
2 From Etudiant natural join Resultat
3 Where extract(year from datenaissance)=2001
4 Group by codeetudiant, nometudiant;
```

no rows selected

23. Donner le nom, la moyenne, le minimum, le maximum des notes de chaque étudiant dont la moyenne est supérieure à 11:

```
SQL> Select nometudiant, min(note), max(note), avg(note)
  2  from Etudiant natural join Resultat
  3  group by codeetudiant, nometudiant
  4  having avg(note)>11 ;
```

NOMETUDIANT	MIN(NOTE)	MAX(NOTE)	AVG(NOTE)
Lefevre	37.49	37.49	37.49
Leclerc	98.12	98.12	98.12
Dupont	94.27	94.27	94.27
Martin	49.61	49.61	49.61

24. Donner le nom, la moyenne, le minimum, le maximum des notes de chaque étudiant de l'année 2 et dont la moyenne est supérieure à 12:

```
SQL> Select nometudiant, min(note), max(note), avg(note)
  2  from Etudiant natural join Resultat
  3  where extract(year from datenaissance)=2
  4  group by codeetudiant ,nometudiant
  5  having avg(note)>12 ;
```

no rows selected

25. Lister le numéro, le nom, la moyenne de chaque étudiant :

```
SQL> Select codeetudiant, nometudiant, avg(note)
  2  From Etudiant natural join Resultat
  3  Group by codeetudiant, nometudiant ;
```

CODEETUDIA	NOMETUDIANT	AVG(NOTE)
E003	Lefevre	37.49
E004	Leclerc	98.12
E001	Dupont	94.27
E002	Martin	49.61

26. Afficher les enseignants qui assurent tous les cours sauf le cours de Réseaux :

27. Vérifier que les enseignants ayant une charge figurent bien dans la table Enseignant :

```
SQL> select codeenseignant from charge c
      2  where not exists (select codeenseignant from enseignant
      3  where codeenseignant=c.codeenseignant );

no rows selected
```

28. Construire une vue qui contient les numéros, les noms et les moyennes des étudiants de l'année1 :

29. Afficher le contenu de la vue créée à la question précédente :