Les sous-requêtes(1)

Une sous requête est une requête avec la commande SELECT imbriquée avec les autres commandes (UPDATE, INSERT, DELETE et CREATE)

Une sous-requête peut être utilisée dans les clauses suivantes :

- La clause WHERE d'une instruction UPDATE,DELETE et SELECT
- La clause FROM de l'instruction SELECT
- La clause VALUES de l'instruction INSERT INTO
- La clause SET de l'instruction UPDATE
- L'instruction CREATE TABLE.

Les sous-requêtes(2)

- Dans la clause WHERE:

Ce type de sous-requête permet de comparer une valeur de la clause WHERE avec le résultat retourné par une sous-requête. Dans ce cas on utilise les opérateurs de comparaison suivants :

=, != ,< ,<= ,> ,>= ,et IN.

Les sous-requêtes(3)

- Exemple :

```
SELECT NUMAD FROM RESULTATS
WHERE  CODE_COURS='KED' AND  NOTE <
                (SELECT NOTE FROM RESULTATS
                WHERE NUMAD=100 AND
                CODE_COURS='KED');
```

Cette requête ramène les numéro d'admission (NUMAD)des étudiants dont la note en KED est plus petite que celle de l'étudiant dont le numéro d'admission est 100 pour le même cours (CODE_COURS=KED)

Les sous-requêtes(4)

- Quelle est-la requête qui ramène les noms des étudiants et leurs numéros et qui fait la même chose que la requête précédente ?

- Réponse:

Les sous-requêtes(5)

```
SELECT E.NUMAD, E.NOM,R.NOTE
FROM RESULTATS R INNER JOIN ETUDIANTS E
ON  E.NUMAD= R.NUMAD
WHERE CODE_COURS='KED'
AND  NOTE <
                (SELECT NOTE FROM RESULTATS
                WHERE NUMAD=100 AND
                CODE_COURS='KED');
```

Les sous-requêtes(6)

L'opérateur IN

IN est utilisée lorsque la sous requête retourne plus qu'une rangée :

Exemple

```
SELECT NUMAD,NOM, PRENOM
FROM ETUDIANTS
WHERE NUMAD NOT IN
                (SELECT NUMAD FROM RESULTATS);
```

Les sous-requêtes(7)

Question :

Comment avoir tous les étudiants qui sont dans le même programme que l'étudiant dont le nom est PATOCHE

Les sous-requêtes(8)

Réponse

```
SELECT * FROM ETUDIANTS
WHERE CODEPRG = (SELECT CODEPRG FROM
ETUDIANTS WHERE NOM='PATOCHE');
```

Les sous-requêtes(9)

Question:

Comment peut-on avoir les étudiants (nom et prénoms) qui ont la même note que l'étudiant numéro 100 et dans le cours KED

Les sous-requêtes(10)

Réponse

```
SELECT nom, prenom  FROM ETUDIANTS
WHERE NUMAD IN
        (SELECT NUMAD FROM RESULTATS
      WHERE CODE_COURS ='KED' AND NOTE =
            (SELECT NOTE FROM RESULTATS
            WHERE CODE_COURS='KED' AND
            NUMAD =100));
```

Semaine 6

- Retour sur la semaine précedente
  - Point de vue de l'élève
  - Point de vue de l'enseignante
- Les sous requêtes, suite et exemple
- Requêtes avec opérateurs d'ensemble. Exemple
- Travailler sur le Tp no1.

Les sous-requêtes(11)

IN, ANY et ALL

IN est utilisée lorsque la sous requête retourne plus qu'une rangée :

Exemple

```
SELECT NUMAD,NOM, PRENOM
FROM ETUDIANTS
WHERE NUMAD NOT IN
                (SELECT NUMAD FROM RESULTATS);
```

Les sous-requêtes(12)

ALL On utilise l'operateur ALL pour que la comparaison se fasse pour toutes les valeurs retournées. Le résultat est vrai si toutes les valeurs répondent à la comparaison

ANY: On utilise l'operateur ALL pour que la comparaison se fasse pour toutes les valeurs retournées. Le résultat est vrai au moins une des valeurs répond à la comparaison

Les sous-requêtes(12)

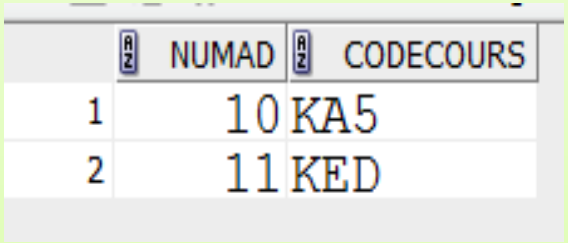
Exemple, voici le contenu de la table Resultats.

	NUMAD	CODECOURS	NOTE
1	10	KED	65
2	10	KA5	85
3	10	KEE	60
4	11	KED	85
5	11	KA5	70
6	11	KEE	75
7	12	KED	60
8	12	KA5	77
9	12	KEE	75

Les sous-requêtes(12)

select numad, codecours from Resultats where note>= ALL(select note from Resultats where codecours ='KA5');

Cette requête retourne le résultat suivant:



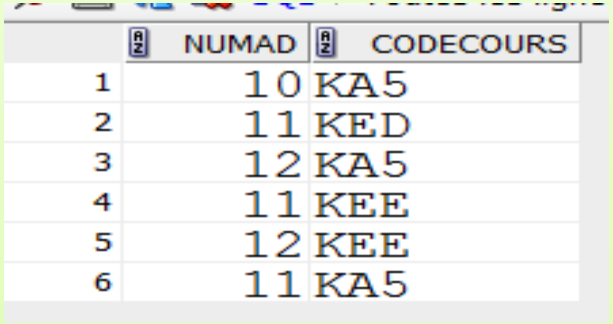
	NUMAD	CODECOURS
1	10	KA5
2	11	KED

Dans ce cas ALL a comparé au Maximum de la note de KA5

Les sous-requêtes(12)

select numad, codecours from Resultats where note>= ANY(select note from Resultats where codecours ='KA5');

Cette requête retourne le résultat suivant



	NUMAD	CODECOURS
1	10	KA5
2	11	KED
3	12	KA5
4	11	KEE
5	12	KEE
6	11	KA5

Dans ce cas ANY a comparé au Minimum de la note de KA5:

Les sous-requêtes(12)

```
select numad, codecours from Resultats where
    note>= ANY(select note from Resultats where
    codecours ='KA5');
```

Cette requête retourne le résultat suivant

Est équivalente à

```
select numad, codecours from Resultats where
    note>= (select min(note) from Resultats where
    codecours ='KA5')
```

Les sous-requêtes(12)

```
select numad, codecours from Resultats where
    note>= ALL(select note from Resultats where
    codecours ='KA5');
```

Cette requête retourne le résultat suivant

Est équivalente à

```
select numad, codecours from Resultats where
    note>= (select max(note) from Resultats
    where codecours ='KA5')
```

Les sous-requêtes(11)

L'opérateur EXISTS

Cet opérateur est utilisé dans une sous-requête pour vérifier l'existence ou non d'enregistrements dans la sous-requête.

Exemple:

```
SELECT * FROM clients WHERE  numclient =10
AND EXISTS (SELECT numclient FROM commande WHERE
    numclient=10);
```

Si le numéro client 10 a commandé des articles(est dans la table commande) alors on ramène les informations qui lui corresponde.

Les sous-requêtes(11)

Attention: différence entre exists et IN

exists exécute la requête externe en premier tandis que IN exécute la requête interne en premier

--IN retourne un résultat dès qu,il ya correspondance entre la clause du where et au moins un des élement de la liste (correspoond à ANY)

---EXISTS retourne des résultat dès que la requête interne contient un enregistrement (une valeur).

Les sous-requêtes: La création de table

Ce type de sous requête permet de créer une table à partir d'une autre table. La nouvelle table contient les valeurs de la sous-requête

```
CREATE TABLE NOTEKED
    (NUMADMISSION,NOTEKED)
AS SELECT NUMAD,NOTE FROM RESULTATS
    WHERE CODE_COURS='KED';
```

Question :Quelle-est la requête qui permet de créer une table NOTESKED qui va avoir comme attributs NOMETUDIANT, PRENOMETUDIANT,TITRECOURS,NOTEKED

Les sous-requêtes: La création de table

Réponse

```
CREATE TABLE NOTESKED (nomEtudiant,
   prenomEtudiant,titreCours,noteKed)
AS
SELECT nom, prenom,titrecours,note
FROM ((Etudiants E INNER JOIN Resultats R  ON
   E.Numad=R.Numad)
INNER JOIN Cours CR ON
   R.Codecours=CR.Codecours)
WHERE R.Codecours='KED' ;
```

Les sous-requêtes: SET de UPDATE (déjà vu)

Dans ce cas le résultat retourné par la sous-requête doit contenir une seule rangée.

```
UPDATE RESULTATS SET NOTE =
  (SELECT AVG(NOTE) FROM RESULTATS WHERE
    CODECOURS ='KEG')
WHERE NOTE <50 AND CODECOURS ='KEG';
```

Les sous-requêtes: INSERT INTO

Ce type de sous-requête permet d'insérer des données dans une table à partir d'une autre table. La sous requête est utilisée à la place de la clause VALUES de la requête principale et peut retourner plusieurs résultats

```
INSERT INTO COURS_DU_SOIR
(SELECT * FROM COURS);
```

La table COURS_DU_SOIR doit avoir des colonnes de même types que la table Cours.

Les règles qui s'applique à la clause VALUES s'applique ici.

Les sous-requêtes: Dans la clause FROM

Exemple:

```
select * from
(select * from resultats order by note desc)
   where rownum <=3;
```

Cette requête renvoit les trois meilleurs resultats.

Requêtes SELECT avec les opérateurs d'ensembles

- Trois opérateurs: INTERSECT, UNION, MINUS.
- Principe:
  - Le nombre de colonnes renvoyées par SELECT 1 doit être le même que celui renvoyé par SELECT 2
  - Le type de données SELECT 1 doit être le même que celui de SELECT 2
  - La clause optionnelle ORDER BY doit se faire selon un numéro de colonne et non le nom de colonne
  - SELECT1 et SELECT2 ne peuvent contenir le clause ORDER BY
- exemples

Requêtes SELECT avec les opérateurs d'ensembles

Table Fournisseurs

	NUMFOURNISSEURS	NOMFOURNISSEUR	ADRESSE
1	13	ALAIN PATOCHE	444 RUE DE LA LUNE MONTREAL, QC
2	10	LE MAGNIFIQUE	23 BOULEVARD ST LAURENT BLAIVILLE,QC
3	11	QUEBECINC	444 RUE DE LA LUNE QUEBEC, QC
4	12	LAVALINC	123 RUE SAINT-GEORGES LAVAL, QC

Table Clients

	NUMCLIENT	NOMCLIENT	ADRESSE
1	11	LE ROI DES SINGES	123 RUE SAINT-GEORGES MONTREAL, QC
2	12	LE RIGOLOT	34 RUE PILON LAVAL, QUEBEC
3	13	ALAIN PATOCHE	444 RUE DE LA LUNE MONTREAL, QC
4	10	LE MAGNIFIQUE	23 BOULEVARD ST LAURENT BLAIVILLE,QC
5	20	MR BEAN	10 AVENUE DE L'HUMOUR LONDRES
6	21	SIMPSON	AVENUE DE LA BÉTISE, SPRINGFIELD

Requêtes SELECT avec les opérateurs d'ensembles

- Opérateur INTERSECT: (ET mathématiques)

```
SELECT NOMCLIENT, ADRESSE
FROM CLIENTS
INTERSECT
SELECT NOMFOURNISSEUR, ADRESSE
FROM FOURNISSEURS
ORDER BY 1;
```

A pour résultat les fournisseurs 10 et 13: les fournisseurs qui sont aussi des clients.

Requêtes SELECT avec les opérateurs d'ensembles

- Opérateur UNION:(OU Exclusif mathématiques)

```
SELECT NOMCLIENT, ADRESSE
FROM CLIENTS
UNION
SELECT NOMFOURNISSEUR, ADRESSE
FROM FOURNISSEURS
ORDER BY 1;
```

A pour résultat TOUS les fournisseurs et tous les clients. Si des doublons sont trouvés, ils sont ramenés une seule fois

Pour ramener toutes les lignes, il faut utiliser l'opérateurs ALL

Requêtes SELECT avec les opérateurs d'ensembles

	NOMCLIENT	ADRESSE
1	ALAIN PATOCHE	444 RUE DE LA LUNE MONTREAL, QC
2	LAVALINC	123 RUE SAINT-GEORGES LAVAL, QC
3	LE MAGNIFIQUE	23 BOULEVARD ST LAURENT BLAIVILLE,QC
4	LE RIGOLOT	34 RUE PILON LAVAL, QUEBEC
5	LE ROI DES SINGES	123 RUE SAINT-GEORGES MONTREAL, QC
6	MR BEAN	10 AVENUE DE L'HUMOUR LONDRES
7	QUEBECINC	444 RUE DE LA LUNE QUEBEC, QC
8	SIMPSON	AVENUE DE LA BÉTISE, SPRINGFIELD

Requêtes SELECT avec les opérateurs d'ensembles

- Opérateur MINUS:

```
SELECT NOMCLIENT, ADRESSE
FROM CLIENTS
MINUS
SELECT NOMFOURNISSEUR, ADRESSE
FROM FOURNISSEURS
ORDER BY 1;
```

A pour résultat TOUS Les CLIENTS (résultat du premier SELECT) MOINS tous les fournisseurs (résultat du deuxième SELECT)

Requêtes SELECT avec les opérateurs d'ensembles

	NOMCLIENT	ADRESSE
1	LE RIGOLOT	34 RUE PILON LAVAL, QUEBEC
2	LE ROI DES SINGES	123 RUE SAINT-GEORGES MONTREAL, QC
3	MR BEAN	10 AVENUE DE L'HUMOUR LONDRES
4	SIMPSON	AVENUE DE LA BÉTISE, SPRINGFIELD