Cours 1 : Requêtes imbriquées

Rappel

- Une condition de sélection utilisée dans une clause **WHERE** ou dans une clause **HAVING** s'exprime sous la forme d'une comparaison entre la valeur d'un attribut (ou expression) et une valeur de référence qui, dans le cas des requêtes simples, est soit un attribut (ou expression), soit une constante

Exemple 1

```
SELECT titre
FROM film
WHERE numfilm <= 2908
```

numfilm ➢ valeur d'attribut
<= ➢ compairaison
2908 ➢ constante

Exemple 2

```
SELECT *
FROM film, individu
WHERE realisateur = numindividu
```

realisateur ➢ valeur d'attribut
= ➢ compairaison
numindividu ➢ valeur d'attribut

Exemple 3

```
SELECT *
FROM location
WHERE TO_CHAR(dateretour, 'YYYY') = '2019'
```

TO_CHAR(...)	➢ expression
=	➢ comparaison
'2019'	➢ constante

SOUS-SELECT

- Le langage SQL étend la notion de valeur de référence au résultat d'une requête. Ainsi, l'expression d'une clause **WHERE** peut prendre la forme :

WHERE *attribut/expression opérateur* (SELECT …)

- On dit alors que la requête, dont le résultat sert de valeur de référence, est une requête imbriquée ou une sous-requête

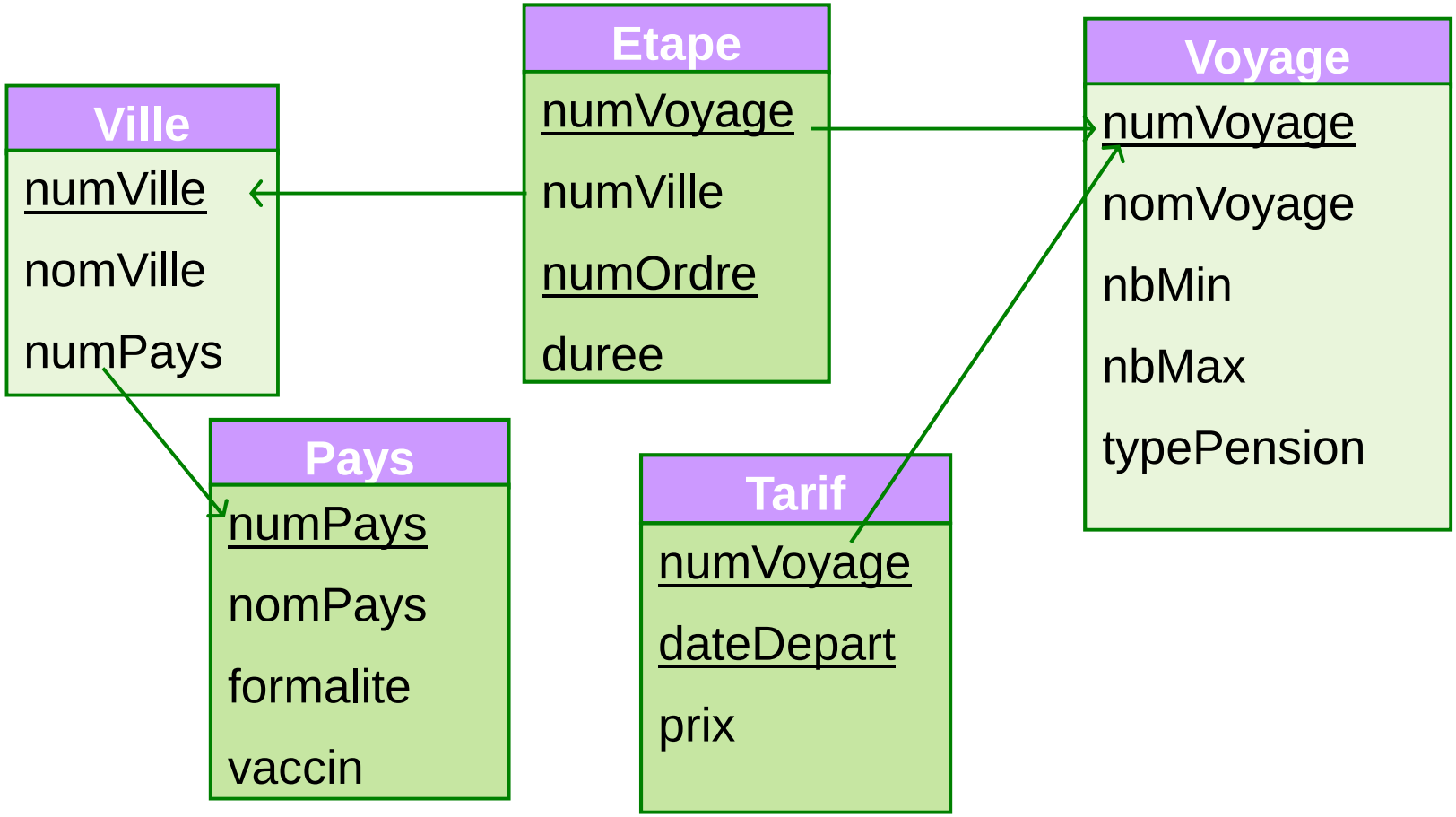
Requêtes imbriquées

- Il est possible d'imbriquer plusieurs requêtes et il n'y a pas de limite dans le nombre de niveaux d'imbrication
- Le résultat de chaque requête imbriquée sert de valeur de référence dans la condition de sélection de la requête de niveau supérieur, appelée requête principale

Types de requêtes imbriquées

- La sous-requête peut renvoyer soit une ligne, soit plusieurs
- Si la sous-requête renvoie plusieurs lignes, nous avons deux cas, selon que la sous-requête est indépendante de la requête principale ou synchronisée avec elle
- Si la sous-requête est synchronisée avec la requête principale, on parle de requête corrélée

Exemple de base : VOYAGES



Exemple de requête imbriquée renvoyant une seule ligne

- Donner les numéros des voyages ayant le même type de pension que le voyage numéro 4

```
SELECT numVoyage FROM voyage
WHERE typePension = (
                    SELECT typePension
                    FROM voyage
                    WHERE numVoyage = 4
                    )
```

Requête imbriquée renvoyant une seule ligne

- Si la requête ne renvoie rien ou si elle renvoie plusieurs lignes, SQL génère une erreur

- L'attribut et la valeur de référence renvoyée par la sous-requête doivent avoir des types compatibles

Exemple de requête imbriquée renvoyant une seule ligne (2)

- Donner les numéros et les dates de départ des voyages ayant le prix le plus bas sur l'ensemble de tous les prix

```
SELECT numVoyage, dateDepart
FROM tarif
WHERE prix = (
            SELECT MIN(prix)
            FROM Tarif
            )
```

Exemple de requête imbriquée renvoyant une seule ligne (2bis)

- Donner les numéros et les dates de départ des voyages ayant le prix le plus bas sur l'ensemble de tous les prix

```
SELECT numVoyage, dateDepart
FROM tarif
WHERE prix <= ALL(
            SELECT prix
            FROM Tarif
            )
```

Requêtes imbriquées renvoyant plusieurs lignes

- La condition de sélection emploie alors :
  - L'opérateur **IN** (équivalent à **= ANY**)
  - L'opérateur **NOT IN** (équivalent à **!= ALL**)
  - Un opérateur simple (**=**, **!=**, **<>**, **<**, **>**, **<=**, **>=**) suivi de **ALL** ou **ANY**
  - L'opérateur **EXISTS**

Rappel

- **ANY** : la condition est vraie si elle est vérifiée pour au moins une des valeurs renvoyées par la sous-requête
- **ALL** : la condition est vraie si elle est vérifiée pour chacune des valeurs renvoyées par la sous-requête
- **IN** : la condition est vraie si elle est vérifiée pour une des valeurs renvoyées par la sous-requête

Exemple de requête imbriquée renvoyant plusieurs lignes

- Donner les numéros des voyages passant pour une ville où passe aussi le voyage numéro 4

```
SELECT DISTINCT numVoyage
FROM Etape WHERE numVille IN (
            SELECT numVille FROM Etape
            WHERE numVoyage = 4
            )
AND numVoyage != 4
```

Jointure interne ou requête imbriquée ?

```
SELECT DISTINCT nomVoyage
FROM voyage V, tarif T
WHERE V.numVoyage = T.numVoyage
AND dateDepart > '31-08-19'
```

Jointure interne ou requête imbriquée ? (bis)

```
SELECT nomVoyage
FROM voyage
WHERE numVoyage IN (
            SELECT numVoyage
            FROM Tarif
            WHERE dateDepart > '31-08-19‘
            )
```

Jointure interne ou requête imbriquée ? (ter)

- Dans le premier cas on pourrait obtenir la date de départ comme résultat du **SELECT** alors que c'est impossible dans le deuxième cas

GROUP BY dans un sous-select rendant plusieurs lignes

- Dates de départ des voyages dont la durée totale est de moins de 10 jours

```
SELECT DISTINCT dateDepart
FROM tarif WHERE numVoyage IN (
        SELECT numVoyage FROM etape
        GROUP BY numVoyage
        HAVING SUM(duree) < 10
        )
```

GROUP BY dans un sous-select rendant plusieurs lignes (2)

- Nombre de voyages dont la durée totale est de moins de 10 jours

```
SELECT COUNT(*)
FROM voyage WHERE numVoyage IN (
        SELECT numVoyage FROM etape
        GROUP BY numVoyage
        HAVING SUM(duree) < 10
        )
```

Exemple de requête imbriquée avec NOT IN

- Numéros des voyages dont il n'y a pas eu de départs après le mois de septembre 2018

```
SELECT numVoyage FROM voyage
WHERE numVoyage NOT IN (
        SELECT numVoyage FROM tarif
        WHERE dateDepart > '30-09-18'
        )
```

Exemple de requête imbriquée avec NOT IN (suite)

- Remarque : NOT IN *n'est pas équivalent* à IN plus la négation de la condition

```
SELECT numVoyage FROM voyage
WHERE numVoyage IN (
            SELECT numVoyage FROM tarif
            WHERE dateDepart <= '30-09-18'
            )
```

- Cela renvoie plutôt les numéros des voyages qui ont au eu moins un départ avant 1er octobre 2018

Extension SQL2

- Les lignes résultats peuvent être des listes d'attributs

WHERE (*attribut1*, *attribut2*, …) *opérateur* (
SELECT *attribut1*, *attribut2*, ...
…
)

Exemple

```
SELECT DISTINCT numVoyage
FROM etape WHERE (numVille, duree) = (
                    SELECT numVille, duree
                    FROM etape
                    WHERE numOrdre = 1
                    AND numVoyage = 10
                    )
```

Exemple de requête imbriquée et corrélée

- Pour chaque voyage, donner les dates de départ pour lesquelles le prix est le moins cher

```
SELECT numVoyage, dateDepart
FROM tarif T1 WHERE prix = (
    SELECT MIN(prix)
    FROM Tarif T2
    WHERE T2.numVoyage = T1.numVoyage
    )
```

Exemple de requête imbriquée et corrélée (bis)

- Pour chaque voyage, donner les dates de départ pour lesquelles le prix est le moins cher

```
SELECT numVoyage, dateDepart
FROM tarif T1 WHERE prix <= ALL (
    SELECT prix
    FROM Tarif T2
    WHERE T2.numVoyage = T1.numVoyage
    )
```

Solution sans corrélation (SQL2)

```
SELECT numVoyage, dateDepart
FROM tarif WHERE (numVoyage, prix) IN (
        SELECT numVoyage, MIN(prix)
        FROM Tarif
        GROUP BY numVoyage
        )
```

Exemple de requête imbriquée et corrélée (2)

- Pour chaque voyage, donner les numéros de ses étapes qui durent plus longtemps que sa première étape

```
SELECT numVoyage, numOrdre
FROM etape E1 WHERE duree > (
        SELECT duree FROM Etape E2
        WHERE E2.numVoyage = E1.numVoyage
        AND numOrdre = 1
        )
```

Exemple de requête imbriquée et corrélée (3)

- Pour chaque voyage, donner les numéros de ses étapes qui durent plus longtemps que sa dernière étape

Exemple de requête imbriquée et corrélée (3bis)

```
SELECT numVoyage, numOrdre
FROM etape E1 WHERE duree > (
                SELECT duree FROM Etape E2
                WHERE E1.numVoyage = E2.numVoyage
                AND numOrdre = ???
                )
```

Exemple de requête imbriquée et corrélée (3ter)

```
SELECT numVoyage, numOrdre
FROM etape E1 WHERE duree > (
                SELECT duree FROM Etape E2
                WHERE E1.numVoyage = E2.numVoyage
                AND numOrdre = (
                SELECT MAX(numOrdre) FROM etape E3
                WHERE E3.numVoyage = E1.numVoyage
                )
                )
```

Exemple de requête imbriquée et corrélée (4)

- Donner les numéros des voyages ayant plus de deux dates de départ

```
SELECT DISTINCT numVoyage
FROM tarif T1 WHERE 2 > (
    SELECT COUNT(*) FROM tarif T2
    WHERE T2.numVoyage = T1.numVoyage
    )
```

Exemple de requête imbriquée et corrélée (4bis)

- Donner les numéros des voyages ayant plus de deux dates de départ (sans correlation)

```
SELECT numVoyage
FROM tarif
GROUP BY numVoyage
HAVING COUNT(*) > 2
```

Opérateur EXISTS

... WHERE EXISTS (*sous-select*)

- Cet opérateur permet de construire un prédicat évalué à vrai si la sous-requête renvoie au moins une ligne

Exemple

- Y a-t-il des voyages qui font au moins 6 étapes ?

```
SELECT 'OUI' FROM DUAL
WHERE EXISTS (
                SELECT * FROM etape
                WHERE numOrdre = 6
                )
```

Remarque

- DUAL est une table « dummy » qui sert, de manière générale, à faire de test

```
SELECT SYSDATE FROM DUAL
SELECT 25+18 FROM DUAL
...
```

Exemple 2

- Numéros des voyages qui font moins de 6 étapes

```
SELECT numVoyage FROM voyage V
WHERE NOT EXISTS (
        SELECT * FROM etape E
        WHERE numOrdre = 6
        AND E.numVoyage = V.numVoyage
        )
```

Question 1

- Pour chaque ville de Grèce, donnée par son numéro et son nom, trouver le nombre de voyages qu'y passent

Question 2

- Pour la période du second semestre 2019, donner les noms et numéros des villes où passent des voyages dont le prix moyen sur cette période est inférieur à 900 €

Question 3

- Pour la période du second semestre 2019, donner les numéros et noms des villes où passent des voyages dont le prix moyen sur cette période est inférieur au prix moyen de ce voyage pendant tout 2019

Question 4

- Donner les numéros et les noms des voyages en demi-pension qui ne font pas étape en Grèce

Question 5

- Pour chaque voyage donné par son numéro et son nom, donner le numéro et le nom de la ville où ce voyage fait sa dernière étape