

02 – Interroger une BASE

SI3 – Exploitation des données

Introduction

- Nécessité de prendre des données dans plusieurs tables.
- Les tables sont reliées virtuellement
- Les clés étrangères ne sont que des contraintes, pas des liens.
- Mise en relation d'une table avec une autre par un 'lien' clé primaire<->clé étrangère

Démonstration Jointure

- Comment connaître le nom des rayons de chaque produit?

idProduit	idRayon	Nom	pxUnit	UnitEnStock
P ₁	R ₁	Lapin	7	28
P ₂	R ₁	Poulet	4	33
P ₃	R ₂	Coca Cola	2	578

idRayon	idPersonnel	Nom
R ₁	PERS ₁	Boucherie
R ₂	PERS ₂	Liquide

La multiplication des tables

- Lorsque l'on spécifie plrs tables dans le FROM
- On multiplie les tables pour en avoir 1:

idProd	idRay	Nom	pxU	UnitStock	idRay	idPers	Nom
P1	R1	Lapin	7	28	R1	PERS1	Boucherie
P2	R1	Poulet	4	33	R1	PERS1	Boucherie
P3	R2	Coca Cola	2	578	R1	PERS1	Boucherie
P1	R1	Lapin	7	28	R2	PERS2	Liquide
P2	R1	Poulet	4	33	R2	PERS2	Liquide
P3	R2	Coca Cola	2	578	R2	PERS2	Liquide

Démonstration jointure

- 1 champ en commun dans les 2 tables:
idRayon
- Principe : mettre les informations des rayons
en face des informations des produits
- **Produit.idRayon = Rayon.idRayon**

La multiplication des tables / sélection

- Lorsque l'on spécifie plrs tables dans le FROM
- On multiplie les tables pour en avoir 1:

idProd	idRay	Nom	pxU	UnitStock	idRay	idPers	Nom
P1	R1	Lapin	7	28	R1	PERS1	Boucherie
P2	R1	Poulet	4	33	R1	PERS1	Boucherie
P3	R2	Coca Cola	2	578	R1	PERS1	Boucherie
P1	R1	Lapin	7	28	R2	PERS2	Liquide
P2	R1	Poulet	4	33	R2	PERS2	Liquide
P3	R2	Coca Cola	2	578	R2	PERS2	Liquide

Résultat de la jointure

idProduit	idRayon	Nom	pxUnit	UnitEnS	idRayon	idPersonnel	Nom
P1	R1	Lapin	7	28	R1	PERS1	Boucherie
P2	R1	Poulet	4	33	R1	PERS1	Boucherie
P3	R2	Coca Cola	2	578	R2	PERS2	Liquide

La jointure en SQL dans Where

```
SELECT Produit.nom , Rayon.nom  
FROM Produit , Rayon  
WHERE Produit.idRayon = Rayon.idRayon
```

- ❑ Résultat : 1 table à 2 colonnes:
 - ❑ l'une contenant le nom des produits,
 - ❑ et l'autre contenant le nom des rayons associés
- ❑ Attention : champ nom présent dans plusieurs tables -> préfixe de la table

La jointure avec Inner Join

- Possibilité d'effectuer la jointure directement dans le FROM avec Inner Join:

- Si 2 tables:

```
SELECT Produit.nom , Rayon.nom  
FROM Produit INNER JOIN Rayon ON Produit.idRayon = Rayon.idRayon
```

- Si plus:

```
SELECT Produit.nom , Rayon.nom, Personnel.nom  
FROM Produit INNER JOIN Rayon ON Produit.idRayon = Rayon.idRayon  
      INNER JOIN Responsable ON Responsable.idPersonnel=Rayon.idPersonnel  
      INNER JOIN Personnel ON Personnel.idPersonnel = Responsable.idPersonnel
```

Jointure d'une table à elle-même

- Dans certains cas, la clé étrangère fait référence à la clé primaire de la même table.
- Résultat d'une association réflexive, ex:
 - Gestion d'une hiérarchie
 - Généalogie
- Obligation de renommer la copie de la table

Jointure d'une table à elle-même


- Personnel (idP, nomP, prenomP, #idSup)
- But : Connaître le nom du supérieur hiérarchique de Pierre MARTIN :

```
SELECT Sup.nom  
FROM Personnel, Personnel as Sup  
WHERE Personnel.idSup = Sup.idP  
AND Personnel.nom = 'MARTIN'  
AND Personnel.prenom = 'Pierre' ;
```

Jointures externes

- Permet de faire une jointure avec des éléments même non-présents dans une des tables

idRayon	nomRayon
R1	Habillement
R2	Boucherie



idProduit	nomProduit	idRayon
1	T-shirt	R1
2	Robe	R1
3	Steak	R2
4	Lapin	R2
5	Poulet	R2

LEFT JOIN ou RIGHT JOIN

- Limite les résultats à ceux incluant chaque ligne de la table à gauche du JOIN

```
SELECT Produit.nom , Rayon.nom  
FROM Rayon LEFT JOIN Produit ON  
Rayon.idRayon=Produit.idRayon;
```

nomRay	nomProd
Habillement	T-shirt
Habillement	Robe
Boucherie	Steak
Boucherie	Lapin
Boucherie	Poulet
Liquide	null

Exemple LEFT JOIN

- Trouver les noms de rayon qui n'ont aucun produit

```
SELECT Rayon.nom  
FROM Rayon LEFT JOIN Produit ON  
    Rayon.idRayon=Produit.idRayon  
WHERE idProd IS NULL ;
```

Rayon.nom
Liquide

Jointures avec comparaison

- Pour les jointures, l'égalité est la plus courante :
 - P.idRayon = R.idRayon
- Possibilité d'utiliser d'autres comparateurs
- Quels sont les produits qui coûtent plus que le « lapin »?

```
SELECT P1.nom  
FROM Produit as P1 , Produit as P2  
WHERE P1.pxUnit > P2.pxUnit  
AND P2.nom = "lapin" ;
```

Les sous-requêtes

- Le but : remplacer une valeur de la requête (dans le WHERE) par une autre requête
- Requête la plus interne exécutée en premier.
- La sous-requête renvoie:
 - Aucune occurrence
 - Une occurrence
 - Plusieurs occurrences.
- On utilise alors des opérateurs différents

Les sous requêtes : renvoie d'1 ligne

- Quels sont les produits qui coûtent plus qu'un lapin?
- Il faut d'abord connaître le prix d'un lapin
- Ensuite on cherchera les produits qui coûtent plus que ce prix.
- La requête qui cherche le prix d'un lapin renvoie 1 seule occurrence:
 - Si plusieurs lignes, c'est qu'il y a plusieurs lapins dans la table.
 - La question initiale est alors stupide

Les sous requêtes : Exemple

- SELECT pxUnit
- FROM Produit
- WHERE nom = 'Lapin' ;
- SELECT nom
- FROM Produit
- WHERE pxUnit > 8.5;

pxUnit
8.5

Les sous requêtes : Exemple

- SELECT nom
- FROM Produit
- WHERE pxUnit > (SELECT pxUnit
FROM Produit
WHERE nom = 'Lapin'
);

Ss requête: renvoie de plrs lignes

- Impossible d'utiliser les comparateurs standards du type = , < , > ... seuls.
- Soit on souhaite comparer à toutes les valeurs retournées par la sous requête. (ALL)
- Soit on souhaite comparer à au moins une des valeurs retournées par la sous requête. (ANY)
- Soit utiliser l'opérateur IN (valable uniquement pour une égalité)

Les sous requêtes : Exemple

- Liste des produits plus chers que tous les produits du rayon 'Boucherie':
- 2 étapes :
 - Construction de l'ensemble des prix des produits du rayon 'Boucherie'
 - Comparaison des prix de chaque produit avec tous les prix renvoyés ci-dessus.

Les sous requêtes: Exemple

- Prix des produits du rayon Boucherie:

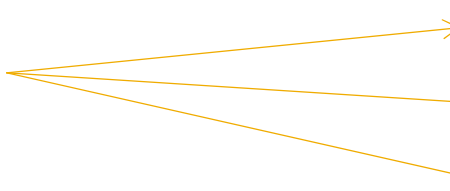
```
SELECT pxUnit  
FROM Produit as P , Rayon as R  
WHERE P.idRayon = R.idRayon  
      AND R.nom = 'Boucherie' ;
```

pxUnit
8,55
3,45
9,50

Les sous requêtes: Exemple

- Noms des produits plus chers que tous les prix précédents :

- SELECT nom
- FROM Produit
- WHERE pxUnit >



pxUnit
8,55
3.45
9.50

- La comparaison est faite 3 fois pour chaque ligne de la table Produit
- Seules les lignes où les 3 comparaisons sont vraies sont gardées

Les sous requêtes: Ex Complet

- Liste des produits plus chers que tous les produits du rayon 'Boucherie':

```
SELECT nom
FROM Produit
WHERE pxUnit > ALL (SELECT pxUnit
                     FROM Produit as P , Rayon as R
                     WHERE P.idRayon = R.idRayon
                          AND R.nom = 'Boucherie'
                     );
```


Exercice

- Liste des passages en caisse (idPassage) qui contiennent au moins 1 produit en commun avec le passage 234.

```
SELECT idPassage
FROM Contenir
WHERE idProd = ANY (
    SELECT idProd
    FROM Contenir
    WHERE idPassageCaisse = 234
);
```

Sous-sous requêtes

- Liste des passages en caisse (idPassage) qui contiennent au moins 1 produit en commun avec les passages du 23 juin 2011.

Sous-sous requêtes

```
SELECT idPassageCaisse
FROM Contenir
WHERE idProd = ANY (
    SELECT idProd
    FROM Contenir
    WHERE idPassageCaisse IN
        (SELECT idPassageCaisse
         FROM PassageCaisse
         WHERE datePassage = '2011/06/23'
        )
);
```

Sous-requête renvoie plrs colonnes

- Jusque là, une requête renvoie 1 colonne que l'on compare à 1 colonne
- Possibilité de faire une comparaison avec des n-uplets.
- Par exemple comparer sur les champs nom ET prénom
- Attention à l'ordre des champs

Plrs colonnes : Exemple

- Donner les produits qui sont dans le même rayon et qui ont la même quantité en stock que l'agrafeuse.

```
SELECT idProd
FROM PRODUIT
WHERE (unitEnStock,idRayon) = (
    SELECT unitEnStock,idRayon
    FROM PRODUIT
    WHERE nom = « agrafeuse »
);
```

Autre exemple : anniversaire

- Liste des produits (idProduit) contenus dans les passages en caisse du jour de l'anniversaire de Pierre MARTIN.

Corrigé anniversaire

```
SELECT idProduit
FROM Contenir as C, PassageCaisse as PC
WHERE C.idPassageCaisse = PC.idPassageCaisse
AND (month(datePassage),dayofmonth(datePassage)) = (
    SELECT month(ddn),dayofmonth(ddn)
    FROM Personnel
    WHERE nom = 'MARTIN'
    AND prenom= 'Pierre'
);
```

Sous requête pour joindre

- Donner le rayon où trouver un lapin.

```
SELECT nom
```

```
FROM Rayon
```

```
WHERE idRayon = ( SELECT idRayon  
                  FROM Produit  
                  WHERE nom = 'Lapin'  
                );
```


Sous requête synchronisée

- Déroulement normale d'interrogation:
 - Exécution de la ss-requête la plus interne
 - Elle renvoie le résultat à la requête immédiatement supérieure.
 - Exécution de la requête avec le résultat renvoyé précédemment
- Possibilité de customiser l'exécution afin d'avoir une sous requête synchronisée avec la requête supérieure.

Sous requête synchronisée

- Évaluation de la sous-requête pour chaque ligne de la requête supérieur
- Donner les produits qui ont les même prix?

```
SELECT idProduit  
FROM PRODUIT as p  
WHERE pxUnit = (SELECT produit.pxUnit  
                FROM produit  
                WHERE produit.pxUnit = p.pxUnit  
                AND produit.idProduit != p.idProduit  
                );
```

Permet la synchronisation de l'exécution



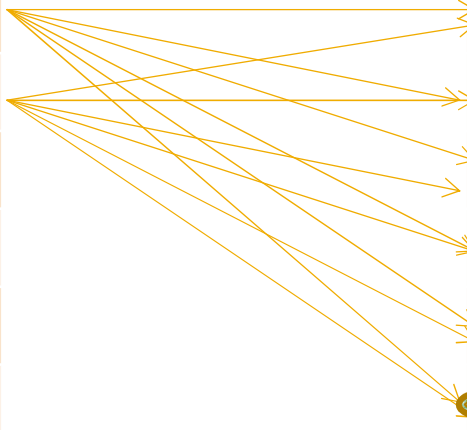
Synchronisation req : Explications

P

idProd	Nom	pxU	Stock
P1	Lapin	7	28
P2	Poulet	4	33
P3	Coca Cola	2	578
P4	Barbecue	12	24
P5	Olivier	13	3
P6	Chaise	7	5

Produit

idProd	Nom	pxU	Stock
P1	Lapin	7	28
P2	Poulet	4	33
P3	Coca Cola	2	578
P4	Barbecue	12	24
P5	Olivier	13	3
P6	Chaise	7	5




Sous requête synchronisée :EXISTS

- EXISTS construit un prédicat vrai si la sous-interrogation qui suit ramène au moins une ligne
- Teste l'existence d'occurrence

Exemple EXISTS:

- Exemple : Nom des rayons contenant au moins 1 produit à 1 euro:

```
SELECT nom  
FROM Rayon as r  
WHERE exists (SELECT *  
              FROM Produit  
              WHERE produit.idRayon=r.idRayon  
              AND pxUnit=1) ;
```



A diagram consisting of a yellow curved line connecting two circular nodes. The first node is positioned over the alias 'r' in the 'FROM Rayon as r' line of the SQL query. The second node is positioned over the 'r.idRayon' attribute in the 'WHERE produit.idRayon=r.idRayon' line of the subquery. This visualizes the correlation between the outer query's table and the subquery's table.

Opérateurs ensemblistes

- Opérations sur les ensembles:
 - Intersection
 - Union
 - Différence
- Seul l'opérateur UNION existe sous MYSQL
- INTERSECT se gère avec l'opérateur IN
- DIFFERENCE se gère avec l'opérateur NOT IN

UNION

- Exemple : Noms et prix des produits des rayons 1 et 2.

```
(SELECT nom , pxUnit  
FROM Produit  
WHERE idRayon = 1 )  
UNION  
(SELECT nom , pxUnit  
FROM Produit  
WHERE idRayon = 2 ) ;
```

UNION avec un OR

- Exemple sans UNION : Noms et prix des produits des rayons 1 et 2.

```
SELECT nom , pxUnit  
FROM Produit  
WHERE idRayon = 1  
OR idRayon = 2 ;
```


FIN

- Questions

