TP 4 : Jointures

Rappel du contexte d'étude "magasin" L'entreprise que nous allons étudier est un commerce de proximité de type "supérette" organisé en rayons contenant des articles de diverses marques. Lors du passage en caisse, on conserve la date, le détail des articles achetés ainsi que le montant total qui a été réglé pour cet achat. On conserve également les informations concernant les fournisseurs et les livraisons. Pour chaque livraison, on mémorise le fournisseur ainsi que le ou les lots qui ont été livrés. Chaque lot concerne un seul article. Un lot est identifié par son numéro de lot et par le code-barres de l'article qu'il concerne. Un article possède un identifiant unique, le codebarres. C'est son appartenance à un lot qui permet de retrouver les informations qui le concernent (dates de fabrication, de livraison et de péremption). On peut d'ailleurs voir sur les photos suivantes quatre articles ayant le même codebarres, mais des numéros de lot différents et donc des dates de péremption différentes.

Les jointures

La jointure est le mécanisme le plus puissant du langage SQL.
Elle permet de combiner les données de plusieurs tables en les liant logiquement suivant les relations qui existent entre elles (clés primaires et / ou étrangères).

Afin de réaliser les exercices utilisez le diagramme fourni dans le fichier **Modèle Logique de données.pdf** celui-ci va vous permettre de visualiser les liens entre les tables

La jointure "classique"

La jointure "classique" est une jointure basée sur l'équivalence entre 2 champs. Elle opère en associant l'enregistrement d'une première table à l'enregistrement d'une seconde table, uniquement si la valeur d'un champ de la première table est égale à la valeur du champ correspondant de la seconde table.

Reprenons l'exemple précédent : on souhaite obtenir la liste des articles et des rayons. Ce qui veut dire que si on veut obtenir la liste des articles avec le rayon où ils sont placés, il faut que l'on associe un article à un rayon.

D'après notre schéma relationnel, on dispose de la clé étrangère "idRayon" de la table article qui fait référence à la clé primaire "idRayon" de la table "rayon", ce qui signifie rappelons-le, qu'un article est placé dans un et un seul rayon (conséquence de la cardinalité 1,1 de notre schéma conceptuel des données). Tout ce qui nous reste à faire, c'est donc de préciser au moteur d'interprétation SQL qu'au lieu de faire un produit cartésien, on ne veut associer que les enregistrements de la table article et de la table rayon où la valeur de la clé étrangère "idrayon" de "article" correspond à la valeur de la clé primaire de la table "rayon".

Vous allez donc rajouter "WHERE a.idRayon = r.idRayon" à la requête précédente pour donner la requête suivante :

```
SELECT a.libelle AS 'Article', r.libelle AS 'Rayon'
FROM article a, rayon r
WHERE a.idrayon = r.idrayon
```

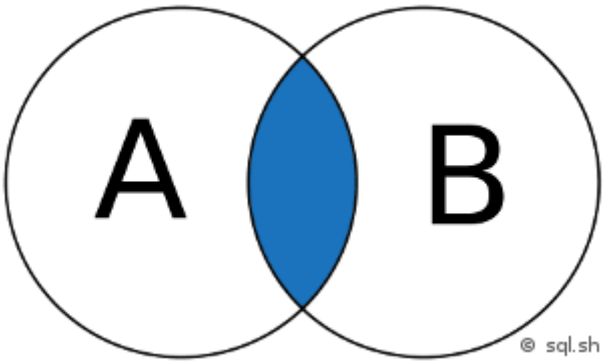
Testez la requête

La jointure avec "JOIN"

Depuis la norme SQL-92, la jointure "classique" est devenue obsolète par la mise à disposition de l'opérateur relationnel "JOIN". La jointure "classique" comporte de trop nombreux inconvénients avec, en particulier, une consommation extrême de ressources systèmes à chaque "produit cartésien". Et d'un point de vue strictement logique, le fait de rassembler toutes les données pour ensuite n'en prendre en compte qu'une infime partie, laissait un peu à désirer…

La jointure “INNER JOIN”

La jointure INNER JOIN permet de sélectionner les données communes entre les deux tables (les données qui figurent dans les deux tables) .




```sql
SELECT a.libelle AS 'Article', r.libelle AS 'Rayon'
FROM article a
INNER JOIN rayon r ON  a.idrayon = r.idrayon
WHERE 1 = 1
```

Testez la requête

La jointure "GAUCHE" avec "LEFT JOIN"

La jointure "gauche" s’effectue avec "LEFT" et conserve les lignes sans lignes équivalentes dans la table de gauche, mais supprime les lignes sans lignes équivalentes dans la table de droite.

On souhaite, par exemple, afficher la liste des fournisseurs ainsi que la liste de leurs livraisons. On va donc conserver toutes les lignes de la table de gauche.

```
SELECT raisonSociale, idLot, idArticle
FROM fournisseur f
LEFT JOIN lot l ON f.idfournisseur = l.idfournisseur
```

Testez la requête

Contrairement à une jointure "équivalente", cette jointure "gauche" nous permet d'avoir également la liste des fournisseurs qui n'ont pas effectué de livraisons (valeurs nulles dans idLot et idArticle) :

Résultats | Messages

	raisonsociale	idLOT	idArticle
1	Legum'Rapid	NULL	NULL
2	Livre boite	NULL	NULL
3	Le laitier	L131199944	3237745052708
4	Le laitier	L131199944	3343719338918
5	Le laitier	L131199944	3593960835823
6	Le laitier	L174895349	3354349412151
7	Le laitier	L277444363	3390734546604

Table fournisseur

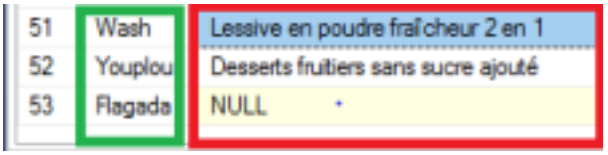
Table Lot

La jointure "DROITE" avec "RIGHT JOIN".

La jointure "droite" s'effectue avec "RIGHT" et conserve les lignes sans lignes équivalentes dans la table de droite, mais supprime les lignes sans lignes équivalentes dans la table de gauche.

On souhaite à présent afficher la liste des marques ainsi que les articles correspondants en magasin.

```
SELECT nom, libelle
FROM article
RIGHT JOIN marque ON article.idMarque = marque.idMarque
```



51	Wash	Lessive en poudre fraîcheur 2 en 1
52	Youplou	Desserts fruitiers sans sucre ajouté
53	Flagada	NULL

Table marque

Table article

Testez la requête

Exercice 4

Réaliser les requêtes permettant d'obtenir les informations ci-dessous.

-la liste des articles (références et libellés) qui ont été achetés

-la liste des articles (références et libellés) qui ont été achetés avec la date à laquelle ils ont été achetés

- liste des articles (références, libellés et prix) avec le nom de la marque de l'article (renommé en "Marque") et le libellé du rayon d'appartenance (renommé en "Rayon)

- la liste des articles (Références, libellés et prix), qui ont été achetés, avec le nom de la marque de l'article et le libellé du rayon d'appartenance ainsi que la date à laquelle ils ont été achetés

-reprenez la requête précédente et utilisez des alias de champs pour chaque champ utilisé : idArticle "Référence" , nom "Marque" , libelle "Produit" ,libelle "Rayon" ,prix "P.U", dateAchat "Date"

- liste des fournisseurs qui n'ont pas encore effectué de livraison -la liste des rayons qui n'ont aucun article attribué