

TP 2 : Les conditions simples

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.

Nous allons maintenant voir comment restreindre une requête à un ensemble bien défini en effectuant une sélection dite "conditionnelle" des enregistrements. Pour cela, nous avons besoin de l'instruction "WHERE". Cette instruction va nous permettre de définir une ou plusieurs conditions que les enregistrements devront vérifier pour pouvoir faire partie des enregistrements concernés par notre requête.



Les opérateurs et les mots-clés utilisés dans les conditions sont également appelés "prédicats"

Les opérateurs logiques "AND", "OR" et "NOT"

On souhaite à présent obtenir la liste de tous les articles à "moins d'1 euro". Pour cela nous allons utiliser la condition WHERE.

```
SELECT idarticle, libelle, prix
FROM article
WHERE prix < 1
```

Il existe 6 opérateurs de comparaison

Opérateur	Description	Accepté sur papier
<	inférieur à	
>	supérieur à	
<=	inférieur ou égal à	≤
>=	supérieur ou égal à	≥
=	égal à	
<> ou !=	différent de	≠

Tester les différents opérateurs avec la requête précédente.

Les opérateurs logiques "AND", "OR" et "NOT"

Ces opérateurs vont nous permettre de combiner plusieurs conditions en construisant des expressions logiques. Par exemple, si on souhaite obtenir la liste de tous les articles dont le prix se situe entre 5 et 10 euros.

Il nous suffit de modifier la condition dans le WHERE :

```
SELECT idarticle, libelle, prix
FROM article
WHERE prix >= 5
      AND prix <= 10
```

Tester les différents opérateurs logiques.

Le prédicat d'intervalle « BETWEEN » (ENTRE)

Le prédicat "BETWEEN" nous permet de définir des intervalles d'une manière plus efficace et plus intuitive qu'avec les prédicats de comparaison. Par exemple, si on souhaite obtenir la liste de tous les articles dont le prix se situe entre 5 et 10 euros.

```
SELECT idarticle, libelle, prix
FROM article
WHERE prix BETWEEN 5 AND 10
```

Autre exemple : on veut obtenir la liste des livraisons effectuées entre le 15 et le 31 mars :

```
SELECT idLot
FROM lot
WHERE dateLivraison BETWEEN '2010/03/15'
AND '2010/03/31'
```

Le prédicat de liste "IN"

Le prédicat "IN" permet de vérifier si des valeurs spécifiques se trouvent dans une liste de valeurs plus simplement qu'avec les prédicats de comparaison.

Par exemple, on veut obtenir le nom des fournisseurs de "Toulon", "Ollioules", "La Seyne sur Mer" et "Sanary sur Mer":

– Avec les prédicats de comparaison on aura :

```
SELECT raisonSociale
FROM fournisseur
WHERE adresseVille = 'Toulon'
OR adresseVille = 'Ollioules'
OR adresseVille = 'La Seyne sur Mer'
OR adresseVille = 'Sanary sur Mer'
```

– Avec le prédicat de liste "IN":

```
SELECT raisonSociale
FROM fournisseur
WHERE adresseVille IN ('Toulon', 'Ollioules',
                        'La Seyne sur Mer', 'Sanary sur Mer')
```



Le prédicat "IN" permet de sélectionner les enregistrements qui contiennent l'une des valeurs de la liste. En associant l'opérateur logique "NOT", on peut faire l'inverse en sélectionnant les enregistrements qui ne contiennent aucune des valeurs présentes dans la liste.

Le prédicat de comparaison partielle "LIKE"

Celui-ci permet d'effectuer une recherche dans des chaînes de caractères. Il est utilisé lorsque l'on ne connaît qu'une partie de la chaîne de caractères, ou que l'on connaît sa forme générale.

Le prédicat "LIKE" permet d'obtenir les enregistrements où la valeur de la colonne sélectionnée est similaire à la chaîne de caractères comparée. Pour utiliser correctement ce prédicat, il faut utiliser l'un des deux caractères de substitution du langage SQL :

Le caractère pourcentage "%" remplace n'importe quelle chaîne de caractères (0 ou plusieurs) :

- on souhaite obtenir la liste des fournisseurs présents dans le Var (département 83) :

```
SELECT raisonSociale  
FROM fournisseur  
WHERE adresseCP LIKE '83%'
```

Le blanc souligné "_" (également appelé "tiret du bas" ou "underscore") remplace n'importe quel caractère (1 seul par "_") :

- même requête que précédemment. On souhaite obtenir la liste des fournisseurs présents dans le Var (département 83) :

```
SELECT raisonSociale  
FROM fournisseur  
WHERE adresseCP LIKE '83____'
```



L'inconvénient du blanc souligné est qu'il faut connaître avec exactitude le nombre de caractères sur lesquels on veut un "joker".

Voir plus :

<https://sql.sh/cours/where/like>

Le prédicat de nullité "NULL"

Le prédicat "NULL" permet d'obtenir les enregistrements où la valeur de la colonne sélectionnée est nulle, c'est-à-dire sans aucune valeur.

Par exemple, on veut obtenir la liste des fournisseurs (raisons sociales et numéros de téléphone) pour qui on ne dispose pas de leurs emails :

Ce qui nous donne en SQL :

```
SELECT raisonSociale, numeroTelephone  
FROM fournisseur  
WHERE email IS NULL
```



On ne peut pas écrire "= NULL" parce que NULL n'est pas "égal à" NULL (la valeur NULL représente une valeur inconnue et il est impossible de dire si deux valeurs inconnues sont égales).

En utilisant l'opérateur "NOT", on peut faire l'inverse, c'est-à-dire ne récupérer que les enregistrements où la valeur de la colonne n'est pas nulle.

Par exemple, on veut obtenir la liste des fournisseurs (raisons sociales et numéros de téléphone) qui nous ont fourni leurs emails :

```
SELECT raisonSociale, numeroTelephone  
FROM fournisseur  
WHERE email IS NOT NULL
```

Exercice :

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

- la liste des fournisseurs dont la distance est de moins de 100 km ;
- la liste des fournisseurs qui se situent à plus de 10 km mais à moins de 100 km ;
- la liste des fournisseurs ayant une adresse email chez "orange.fr";
- la liste des fournisseurs qui ont dont la raison sociale contient « ar » ;
- la liste des lots fabriqués avant le 23 mars 2010 et qui ont été livrés entre le 24 et le 26 mars ;
- la liste des articles du rayon n° 12 dont le packaging ou l'unité de mesure n'a pas été renseigné ;
- la liste des articles dont le packaging est "Par 3", "Par 6" ou "Par 12".