**Dossier BDD.4 : Langage de requête, interrogation d'une base de données**

|  |
| --- |
| Objets d’un traitement informatique |

Contexte (suite dossier précédent):

Les magasins Leclerc veulent améliorer leur outil CRM afin de mieux cibler leurs campagnes publicitaires.

**CLIENT** (num,nom,prenom,adresse, CP, ville, tel, numCategorie)

Clé primaire : num

Clé étrangère : numCategorie en référence à num de la relation Categorie

**CATEGORIE** (num, nom)

Clé primaire : num

**PRODUIT** (num,nom,prix)\*

Clé primaire : num

**ACHAT** (numClient,numProduit, date, nombre)

Clé primaire: numClient, numProduit,

Clés étrangères : numClient en référence à num de la relation client

numProduit en référence à num de la relation Produit

\* : Un Produit est identifié par son code barre, composé de 10 chiffres.

|  |  |
| --- | --- |
| **Travail à faire** | |
| **1** | Un client peut-il acheter plusieurs produits différents ? Citez un exemple de jeu de données |
| **2** | Un produit peut-il être acheté par plusieurs clients ? Citez un exemple de jeu de données |
| **3** | Faire le MCD de ce modèle |
| **4** | Un client peut-il acheter plusieurs fois le même produit à des dates différentes ? Citez un exemple de jeu de données |
| **5** | Proposez une solution au schéma relationnel pour que l'ensemble de ces points soient respectés |
| **6** | Faire la requête SQL de mise à jour de la base de données |

# Langage d'interrogation des données simple

Formalisme :

**SELECT `***Champ1*` , `*Champ2*` , **…**

**FROM** `nomTable` ;

**Pour sélectionner tous les champs mettre le caractère \* dans le Select**

|  |  |
| --- | --- |
| **Travail à faire** | |
| **7** | Afficher le nom des catégories |
| **8** | Afficher toute la liste des clients |
| **9** | Afficher le nom des villes des clients |

## Distinct

Rajouter "Distinct" après le SELECT permet de ne pas afficher les doublons.

SELECT **DISTINCT `***Champ1*` , `*Champ2*` , **…**

FROM `nomTable` ;

|  |  |
| --- | --- |
| **Travail à faire** | |
| **10** | Optimiser l'affichage du nom des villes des clients. |
| **11** | Afficher le nom des villes et les numéros de catégorie de ces villes. |
| **12** | En déduire la ville ayant toutes les catégories de client (1 à 5) |
| **13** | Afficher le nom des produits |
| **14** | Afficher la liste des achats |

## Les tris

En SQL il est possible de trier le résultat du Select :

* par ordre croissant : Ascendant 🡪 ASC
* par ordre décroissant : Descendant 🡪 DESC

SELECT **`***Champ1*` , `*Champ2*` , **…**

FROM `nomTable`

**ORDER BY** `*Champ1*` **DESC**, `*Champ2*` **ASC ;**

|  |  |
| --- | --- |
| **Travail à faire** | |
| **15** | Afficher la liste des prix du moins chère au plus chère. |
| **16** | Afficher la liste des achats triée par produit (ordre croissant), puis de la date du plus récent au plus ancien. |
| **17** | Afficher la liste des clients par ordre alphabétique. |

# SQL - Les critères de restrictions

Des critères de restrictions peuvent être utilisés par des requêtes de mise à jour (UPDATE), de suppression (DELETE) ou d'interrogation/sélection (SELECT).

Formalisme :

**Requête d'interrogation**

SELECT `Champ1` , `Champ2` , …   
FROM `nomTable`

WHERE Restriction 1

AND Restriction 2

OR Restriction 3….

**Requête de suppression**

DELETE FROM ` nomTable `   
WHERE Restriction 1

AND Restriction 2

OR Restriction 3….

**Requête de mise à jour**

UPDATE `nomTable`

SET `ChampsMaj`

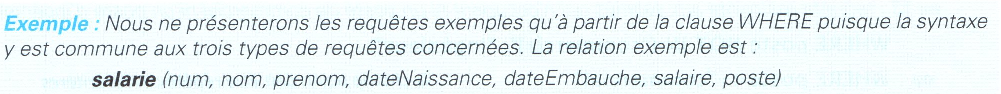
WHERE Restriction 1

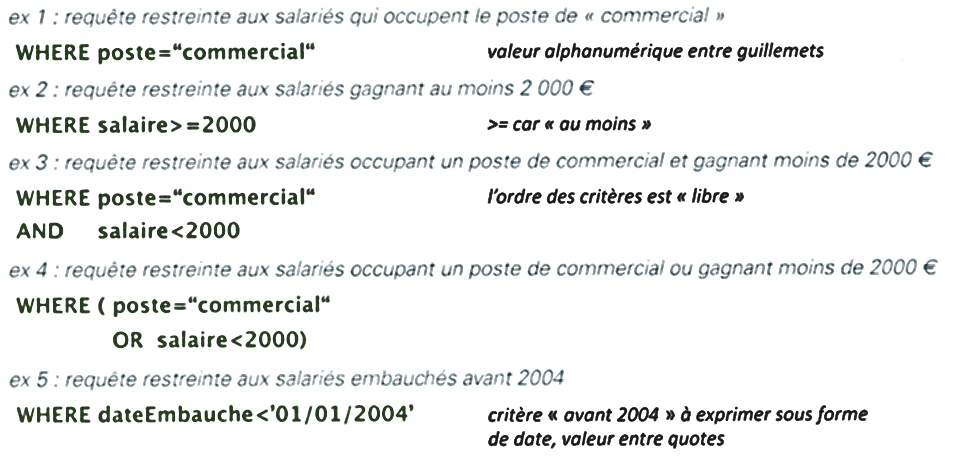
AND Restriction 2

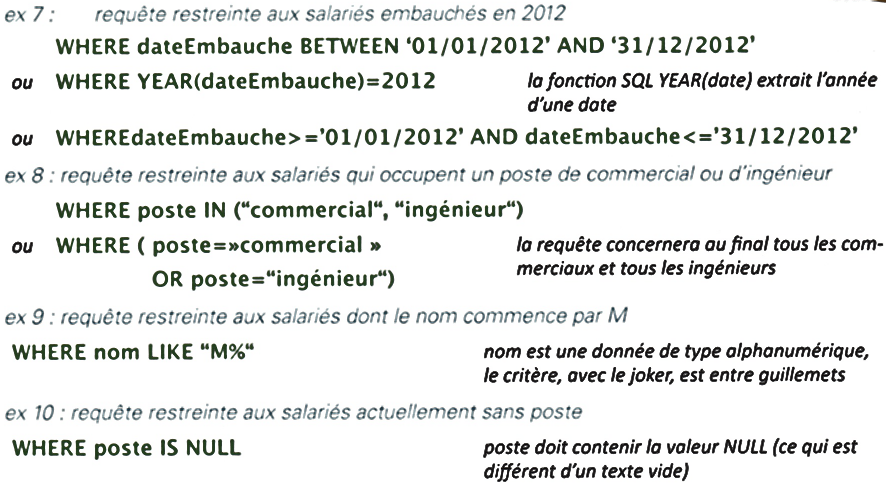
OR Restriction 3….

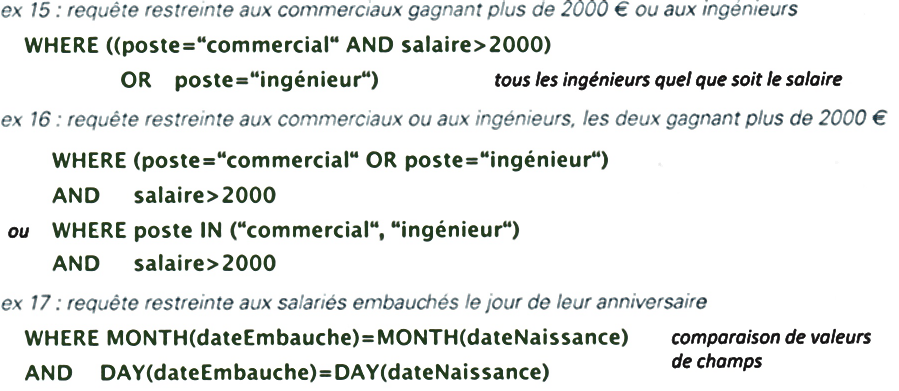
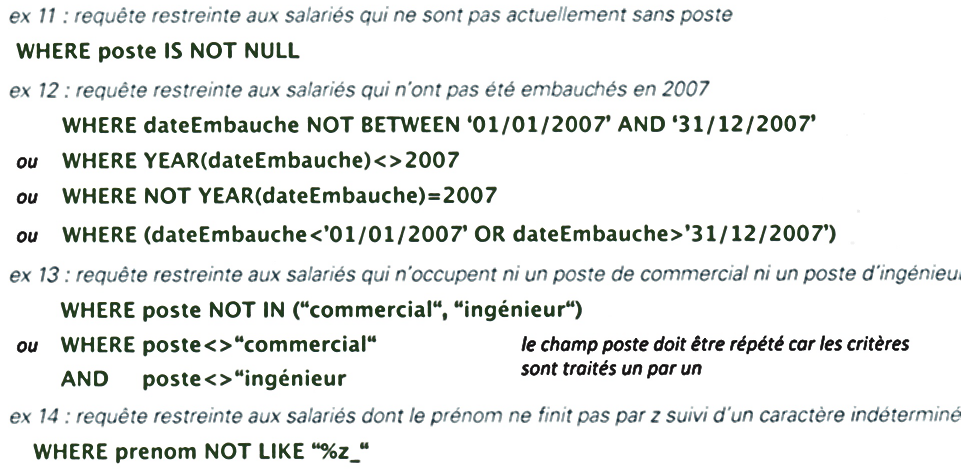
La mise en relation de ces expressions avec ces valeurs se fait au moyen :

* d'opérateurs de comparaison : = , > , < , >= , <= , <> (différent de)
* d'opérateurs SQL
  + expr **BETWEEN** val1 **AND** val2 🡪 expr doit être entre la valeur 1 et la valeur 2)
  + expr **IN** (val1, val2…) 🡪expr doit avoir une des valeurs dans la liste
  + expr **LIKE** chaineCar 🡪 expr doit être égale à la chaine de caractère (utilisation du % possible)
* l'opérateur de négation NOT









|  |  |
| --- | --- |
| **Travail à faire** | |
| **18** | Afficher les produits qui coûtent plus de 500€. |
| **19** | Afficher les informations du client numéro 2 |
| **20** | Afficher la liste des achats de 2014. (le format de la date est : "AAAA-MM-JJ") |

# Langage d'interrogation des données avancée

## Les jointures

Dès lors que la requête se fait sur plusieurs tables des jointures sont nécessaires.

Une jointure se fait toujours entre une **clé étrangère** et une **clé primaire** !

|  |  |
| --- | --- |
| **Travail à faire** | |
| **21** | Quels sont les champs qui peuvent être liés entre la relation catégorie et la relation client ?  Pour répondre à cette question, veuillez relier les champs. |

|  |
| --- |
| CLIENT |
| **num**  nom  prenom  adresse  CP  ville  tel  #numCategorie |

|  |
| --- |
| CATEGORIE |
| **num**  nom |

FORMALISME

SELECT `Table1`.Champ1` , `Table2`.Champ1`, …

FROM `Table1`,`Table2`

**WHERE** `Table1`.`CléPrimaire` **=** `Table2`.`CléEtrangère` ;

|  |  |
| --- | --- |
| **Travail à faire** | |
| **22** | Afficher la liste des clients avec leur nom de catégorie. |
| **23** | Afficher le téléphone et le nom de catégorie des clients |
| **24** | Afficher le téléphone des clients de la catégorie "cadre" |
| **25** | Expliquer à quoi peut servir la requête n°25 pour Leclerc ? |
| **26** | Afficher le nom des produits achetés par M. Fifi |
| **27** | Quels sont les produits achetés par les ouvriers ? |
| **28** | Quelles sont les catégories de client qui achètent la PS4 ? |

## Les fonctions d'agrégat

Il s'agit de calculs répétés de manière indépendante pour chaque enregistrement.

|  |  |
| --- | --- |
| Fonction d'agrégat SQL | Description |
| SUM (champ) | Somme des valeurs |
| AVG (champ) | Moyenne des valeurs |
| MIN (champ) | Minimum des valeurs |
| MAX (champ) | Maximum des valeurs |
| COUNT (\*) | Compte le nombre d'enregistrement |
| +,-,\* | Un champ peut aussi être calculé avec un autre champ |

FORMALISME

SELECT **SUM**(`Champ1`)

FROM `Table1` ;

|  |  |
| --- | --- |
| **Travail à faire** | |
| **29** | Afficher le nombre de produit acheté |
| **30** | Afficher le nombre de client |
| **31** | Afficher le prix moyen , le prix minimum et le prix maximum des produits |
| **32** | Afficher l'ensemble des achats en ajoutant une colonne qui calcul le montant total de chaque achat. |
| **33** | Afficher le chiffre d'affaire de Leclerc |

## Le regroupement des résultats

Quand on utilise un agrégat, nous souhaitons parfois grouper les résultats sur un champ. Pour se faire il faudra utiliser la clause GROUP BY.

Formalisme:

SELECT SUM(`Champ2`), Champ1`, …

FROM `Table1`

**GROUP BY** `Champ1` ;

|  |  |
| --- | --- |
| **Travail à faire** | |
| **34** | Afficher le nombre de produit acheté par client |
| **35** | Exécuter l'instruction sans le group By. Que se passe t-il ? |
| **36** | Afficher le chiffre d'affaire (du plus grand au plus petit)de Leclerc par client (afficher le nom et l'email du client) |
| **37** | Afficher le chiffre d'affaire par catégorie et en tirer une conclusion |

## Les restrictions d'affichage des résultats

Il est possible de faire une restriction sur un regroupement.

Formalisme:

SELECT SUM(`Champ2`), Champ1`, …

FROM `Table1`

GROUP BY `Champ1`

**HAVING SUM**(`Champ2`) > valeur

|  |  |
| --- | --- |
| **Travail à faire** | |
| **38** | Afficher le nombre de produit acheté par client (afficher les clients qui ont acheté au moins 10 produits) |
| **39** | Afficher les clients (nom et email) ayant achetés plus de 1000€ de marchandises. |

## Imbrication de requête

Il est possible d'imbriquer plusieurs requêtes dans une restriction

SELECT Champ1`, …

FROM `Table1`

WHERE`Champ1` = **(REQUETE 2)**

|  |  |
| --- | --- |
| **Travail à faire** | |
| **40** | Afficher les informations du produit qui coûte le plus chère |
| **41** | Afficher la liste des clients ayant acheté le produit le plus chère |

## Les grandes étapes d'une requêtes SELECT SQL

**Etape 4 : restriction**

*(si besoin)*

WHERE/AND/OR

**Etape 1 : sélection des champs**

SELECT

**Etape 3 : jointure**

*(si plusieurs tables)*

WHERE/AND

**Etape 5 : Trie**

*(si besoin)*

ORDER BY

**Etape 2 : sélection des tables**

FROM

**Etape 6 : Regroupement**

*(si besoin)*

GROUP BY

**Etape 7 : Restriction/Regroupement**

*(si besoin)*

HAVING…