|  |  |
| --- | --- |
| **BTS SIO** | **BTS SIO (SLAM)** |
|  | **Bloc 1** |
|  |  |
| **Persistance en BDD** | |

**SQL**

|  |  |  |  |
| --- | --- | --- | --- |
| **date** | **révision** | | |
|  | Timothée Robert | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **BTS SIO** | |  | **BTS SIO (SLAM)** |
|  | | |  |

|  |  |
| --- | --- |
| Introduction au SQL |  |

Le SQL, Structured Query Langage, comporte plusieurs parties :

* Un **langage de définition de la structure de données, appelé LDD**, Langage de définition de données, DDL en anglais, permettant de créer des ***base de données*** et des ***tables***

Nous avons déjà vu comment créer une Base de données et des tables lors de précédents TP.

* Un **langage de manipulation des données, appelé LMD**, DML en anglais, qui permet de manipuler les données, avec 4 manipulations de base
  + SELECT : sélection de données dans une ou plusieurs tables
  + INSERT : ajout de données dans une table
  + DELETE : suppression de données dans une table
  + UPDATE : Mise à jour, c’est-à-dire modification, de données dans une table
* Un **langage de contrôle de données**, LCD, DCL en anglais (Data Control Langage), permettant de gérer
  + les accès à une base de données, en fonction des droits utilisateurs (instructions GRANT, DENY, REVOKE)
  + et la concurrence (instructions COMMIT, ROLLBACK, LOCK)
* Un langage de contrôle des transactions

Ensuite, pour chaque Base de données, il existe des extensions, par exemple le PL-SQL pour Oracle et le Transact SQL pour Microsft SQL Server.

Nous allons aborder aujourd’hui la sélection dans une table

# Sélection dans une table

## Exemples de sélection

1.1

SELECT \* FROM JEUX;

1.2

SELECT JEUX.CODE\_JEU, JEUX.LIB\_JEU FROM JEUX ;

1.3

SELECT DISTINCT CODE\_DEVELOPPEUR FROM JEUX ;

## Explications :

L’instruction **SELECT** permet de lister les champs (colonnes) d’une ou plusieurs tables que l’on souhaite récupérer.

L’instruction **FROM** permet de lister la ou les tables à partir desquelles on récupère ces champs.

SELECT \* FROM JEUX;

En français : Sélectionner toutes les lignes et toutes les colonnes depuis la table JEUX  
Cette commande interroge la table Jeux, et retourne **tous les champs (colonnes)** de cette table, ainsi que **tous les enregistrements** (les lignes).

SELECT JEUX.CODE\_JEU, JEUX.LIB\_JEU FROM JEUX ;

En français : Sélectionner toutes les lignes et les colonnes CODE\_JEU et LIB\_JEU depuis la table JEUX

Remarque : je recommande de préfixer systématiquement par le nom de la table pour ensuite s’habituer et faciliter les JOINTURES (combinaisons de champs de plusieurs tables).

SELECT DISTINCT CODE\_DEVELOPPEUR FROM JEUX ;

En français : sélectionner les valeurs distinctes de CODE\_DEVELOPEUR à partir de la table JEUX.

L’instruction DISTINCT permet de récupérer les valeurs DISTINCTES d’un champ ou d’une combinaison de champs

Exercices :

* Donner le résultat des requêtes 1.2, et 1.3
* Que donnent les instructions suivantes ?

1.4 SELECT DISTINCT CODE\_PAYS FROM DEVELOPPEUR ;

1.5 SELECT ID\_TRANSACTION FROM VENTES;

En fait l'ordre SQL SELECT est ***composé de 6 clauses*** dont 4 sont optionnelles.  
Clauses de l'ordre SELECT :

|  |  |
| --- | --- |
| **SELECT** | Spécification des colonnes du résultat |
| **FROM** | Spécification des tables sur lesquelles porte l'ordre |
| **WHERE** | Filtre portant sur les données (conditions à remplir pour que les lignes soient présentes dans le résultat) |
| **GROUP BY** | Définition d'un groupe (sous ensemble) |
| **HAVING** | Filtre portant sur les résultats (conditions de regroupement des lignes) |
| **ORDER BY** | Tri des données du résultat |

# Filtres : restrictions

## Instruction Where

L’instruction WHERE permet d’effectuer des filtres sur le résultat d’une requête.

La clause WHERE n'est jamais utilisée seule : elle complète les instructions comme [SELECT](http://jacky.aubry.free.fr/aj_suite/access/sql/acSelect.htm), [DELETE](http://jacky.aubry.free.fr/aj_suite/access/sql/acDelete.htm), [UPDATE](http://jacky.aubry.free.fr/aj_suite/access/sql/acUpdate.htm)..., et sert à filtrer les données.

Dans le SELECT, on choisit les ***champs*** (colonnes). Avec le WHERE on filtre les ***lignes***.

## Exemples

2.1

SELECT CODE\_JEU FROM JEUX WHERE CODE\_DEVELOPPEUR = 'LS' ;

2.2

SELECT CODE\_JEU FROM JEUX WHERE DATE\_SORTIE > '2014-01-01' ;

2.3

SELECT \* FROM JEUX WHERE LIB\_JEU LIKE 'D%' ;

***Explications*** :

SELECT CODE\_JEU FROM JEUX WHERE CODE\_DEVELOPPEUR = 'LS' ;

En français : sélectionner tous les enregistrements (lignes) de la table JEUX pour lesquels le champ (colonne) CODE\_DEVELOPPEUR vaut LS, et ne conserver que le champ (colonne) CODE\_JEU

Lorsque on compare avec du texte, ce texte figure entre simples quotes (‘’). Ceci est valable pour la plupart des SGBD (à part Access qui n’est pas vraiment un SGBD) : Oracle, SQL Server, Derby, MySQL, Postgres …

SELECT CODE\_JEU FROM JEUX WHERE DATE\_SORTIE > '2014-01-01' ;

**En français** : sélectionner tous les enregistrements (lignes) de la table JEUX pour lesquels la date de sortie (champ DATE\_SORTIE) est supérieure au 1er Janvier 2014.

Lorsqu’on compare avec un champ date, la syntaxe est spécifique, et varie avec chaque SGBD. Pour MYSQL, une **constante date** est de la forme ‘YYYY-MM-DD’

Concrètement, pour saisir une valeur de date dans un filtre (WHERE) ou lorsqu’on insère une donnée, on commence par une quote (‘) puis les 4 chiffres de l’année puis un tiret (-) puis les deux chiffres du mois puis de nouveau un tiret puis les deux chiffres du jour puis une quote (‘) fermante[[1]](#footnote-1).

## Prédicat (Filtre) simple

Un prédicat simple est le résultat de la comparaison de deux expressions au moyen d'un opérateur de comparaison qui peut être :

= égal

!= différent

< inférieur

<= inférieur ou égal

> supérieur

>= supérieur ou égal

Différentes types d’expression : NUMERIQUE, CARACTERE et DATE peuvent être comparés au moyen de ces opérateurs :

* Pour les types date, la relation d'ordre est l'ordre chronologique.
* Pour les types caractère, la relation d'ordre est l'ordre alphabétique.

Il faut ajouter à ces opérateurs arithmétiques classiques les opérateurs suivants :

expr1 BETWEEN expr2 AND expr3

vrai si expr1 est compris entre expr2 et expr3, *bornes incluses*

expr1 IN (expr2, expr3, ...)

vrai si expr1 est égale à l'une des expressions de la liste entre parenthèses

expr LIKE chaine

où chaine est une chaîne de caractères pouvant contenir l'un des caractères jokers :

\_ remplace exactement 1 caractère

% remplace une chaîne de caractères de longueur quelconque, y compris de longueur nulle.

Exercices :

* Donner le résultat des requêtes 2.1, 2.2, 2.3
* Que donnent les instructions suivantes ?

2.4

SELECT DEVELOPPEUR.ID\_EDITEUR, DEVELOPPEUR.LIB\_EDITEUR

FROM DEVELOPPEUR

WHERE CODE\_PAYS IN ('JP', 'BE') ;

2.5

SELECT \* FROM JEUX WHERE PRIX\_SORTIE BETWEEN 50 AND 65 ;

## Prédicats (Filtre) composés

Les opérateurs logiques AND (et) et OR (ou inclusif) peuvent être utilisés pour combiner entre eux plusieurs prédicats. L'opérateur NOT placé devant un prédicat en inverse le sens.

L'opérateur AND est prioritaire par rapport à l'opérateur OR. Des parenthèses peuvent être utilisées pour imposer une priorité dans l'évaluation du prédicat, ou simplement pour rendre plus claire l'expression logique.

***Exemples*** :

2.7

SELECT DEVELOPPEUR.ID\_EDITEUR, DEVELOPPEUR.LIB\_EDITEUR

FROM DEVELOPPEUR

WHERE CODE\_PAYS = 'JP' OR CODE\_PAYS ='BE' ;

2.8

SELECT \* FROM JEUX WHERE (PRIX\_SORTIE BETWEEN 50 AND 65 ) AND DATE\_SORTIE > '2014-01-01' ;

***Explications*** :

SELECT DEVELOPPEUR.ID\_EDITEUR, DEVELOPPEUR.LIB\_EDITEUR

FROM DEVELOPPEUR WHERE CODE\_PAYS = 'JP' OR CODE\_PAYS ='BE' ;

En français :

Sélectionnez le code et le libellé éditeur à partir de la table éditeur, en filtrant les enregistrements pour lesquels le code pays est JP (Japon) ou bien BE (Belgique).

Remarque ; cette requête est identique à la requête :

SELECT DEVELOPPEUR.ID\_EDITEUR, DEVELOPPEUR.LIB\_EDITEUR

FROM DEVELOPPEUR WHERE CODE\_PAYS IN ('JP', 'BE');

SELECT \* FROM JEUX WHERE (PRIX\_SORTIE BETWEEN 50 AND 65 ) AND DATE\_SORTIE > '2014-01-01' ;

En français :

Sélectionnez tous les champs de tous les enregistrements de la table JEUX pour lesquels le prix de sortie (prix de ventes initial ou conseillé) est compris entre 50 et 65 euros et qui sont sortis après le 1 Janvier 2014.

Exercices :

* Donner le résultat des requêtes 2.7, 2.8

1. La quote simple n’est pas à confondre avec la quote double (guillemet) «  [↑](#footnote-ref-1)