

# Conception des bases de données SQL

INSY2S  
Septembre 2018



# Objectifs du cours

- **Le SQL : Structured Query Language**
- **L'interrogation des données**
- **Les opérateurs, les fonctions et les jointures**
- **La mise à jour des données**
- **Le langage de définition de données**

## Le SQL : Structured Query Language

- **Le langage SQL est un langage simple qui permet de travailler sur une base de données**
  - LID
    - ⊙ Langage d'interrogation des données : SELECT
  - LDD
    - ⊙ Langage de définition des données : ALTER, CREATE, DROP
  - LMD
    - ⊙ Langage de manipulation des données : UPDATE, INSERT, DELETE
  - LCD
    - ⊙ Langage de contrôle des données : GRANT, REVOKE

## L'interrogation des données La syntaxe du SELECT

- Permet d'interroger de manière simple des données
- **SELECT** [DISTINCT] { \* | [nom\_table.]nom\_col1[,nom.col2....] | [,expression] }  
FROM { [nom\_créateur1.]nom\_table1[,nom\_créateur2.]nom\_table2...}  
[WHERE prédicat]  
[GROUP BY nom\_coli[,nom\_colj.....] [HAVING prédicat] ]  
[ORDER BY nom\_colp [DESC] [,nom\_colq [ASC] ...] ;
- On peut commenter le code SQL en l'encadrant par /\* Commentaire \*/
- **Syntaxe** :- entre crochets des arguments facultatifs- entre accolades des arguments obligatoire - les choix possibles sont séparés par des barres verticales
- **SELECT** : choix des colonnes à afficher
- **FROM** : choix des tables de travail
- **WHERE** : condition sur les données à sélectionner
- **GROUP BY/ HAVING/ ORDER BY** : vu plus tard

# L'interrogation des données La syntaxe du SELECT

- `SELECT *` : permet de choisir toutes les colonnes d'une table
- `SELECT col1` : permet de choisir une colonne d'une table
- `SELECT distinct` : permet de choisir des valeurs distinctes d'une table
- Exemple : Les prénoms différents des employés
- `DESC table` : affiche la description d'une table

## EXERCICES : Simples sélections

- 1 : Sélectionner toutes les colonnes de la table SERV.
- 2 : Sélectionner d'une autre manière ces colonnes.
- 3 : Sélectionner les colonnes SERVICE et NOSERV de la table SERV.
- 4 : Sélectionner toutes les colonnes de la table EMP.
- 5 : Sélectionner les emplois de la table EMP.
- 6 : Sélectionner les différents emplois de la table EMP.

## **SELECT Les valeurs nulles**

- Une valeur nulle est différente de la valeur zéro. Elle correspond à une valeur non renseignée.
- Il faut comprendre cette valeur comme l'ensemble vide et imaginer les calculs en découlant :
  - $\text{NULL} + 3$  donne  $\text{NULL}$ .
  - $\text{NULL} = 0$  est faux.
  - $\text{NULL}$  n'est pas inférieur à 3 et  $\text{NULL}$  n'est pas supérieur à 3.
- Pour détecter une clause **NULL**, il faut utiliser le prédicat **IS NULL**.  
L'inverse étant **IS NOT NULL**.
- Pour transformer la clause **Null**, on utilise la fonction **IFNULL (Null VaLue)**
  - $\text{IFNULL}(\text{champ}, 0)$  donne la valeur 0 si le champ est nul sinon donne sa valeur.
  - Important pour protéger les colonnes pouvant être nulles.

# EXERCICES :

## SELECT AVEC CLAUSE WHERE : Prédicats simples

- 7 : Sélectionner les employés du service N°3.
- 8 : Sélectionner les noms, prénoms, numéro d'employé, numéro de service de tous les techniciens.
- 9 : Sélectionner les noms, numéros de service de tous les services dont le numéro est supérieur à 2.
- 10 : Sélectionner les noms, numéros de service de tous les services dont le numéro est inférieur ou égal à 2.
- 11 : Sélectionner les employés dont la commission est inférieure au salaire. Vérifiez le résultat jusqu'à obtenir la bonne réponse.



# EXERCICES :

SELECT : Travailler avec les nuls

- 12 : Sélectionner les employés qui ne touchent jamais de commission.
- 13 : Sélectionner les employés qui touchent éventuellement une commission et dans l'ordre croissant des commissions.
- 14 : Sélectionner les employés qui ont un chef.
- 15 : Sélectionner les employés qui n'ont pas de chef.

# SELECT

## Les opérateur logiques AND, OR et NOT

- Rappels logiques
  - VRAI AND VRAI donne VRAI
  - VRAI AND FAUX donne FAUX
  - FAUX AND FAUX donne FAUX
  - VRAI OR VRAI donne VRAI
  - VRAI OR FAUX donne VRAI
  - FAUX OR FAUX donne FAUX
- AND prioritaire sur le OR : il faut donc parenthéser le OR
- NOT inverse le résultat logique de l'opération

# EXERCICES

## **SELECT AVEC CLAUSE WHERE : and, or, not**

- **16 : Sélectionner les noms, emploi, salaire, numéro de service de tous les employés du service 5 qui gagnent plus de 20000 €.**
- **17 : Sélectionner les vendeurs du service 6 qui ont un revenu mensuel supérieur ou égal à 9500 €.**
- **18 : Sélectionner dans les employés tous les présidents et directeurs. Attention, le français et la logique sont parfois contradictoires.**
- **19 : Sélectionner les directeurs qui ne sont pas dans le service 3.**
- **20 : Sélectionner les directeurs et « les techniciens du service 1 ».**
- **21 : Sélectionner les « directeurs et les techniciens » du service 1.**
- **22 : Sélectionner les employés du service 1 qui sont directeurs ou techniciens.**
- **23 : Sélectionner les employés qui ne sont ni directeur, ni technicien et travaillant dans le service 1.**

## Les différents types de données :

trois types principaux de données

- **Numérique :**
- Pour les **Nombres entiers** on utilise souvent :

| Type      | Nombre d'octets | Minimum              | Maximum             |
|-----------|-----------------|----------------------|---------------------|
| TINYINT   | 1               | -128                 | 127                 |
| SMALLINT  | 2               | -32768               | 32767               |
| MEDIUMINT | 3               | -8388608             | 8388607             |
| INT       | 4               | -2147483648          | 2147483647          |
| BIGINT    | 8               | -9223372036854775808 | 9223372036854775807 |

- Pour les **Nombres décimaux** on utilise **FLOAT** , **NUMBER** ou **DECIMAL(nombre,précision)**
- **NUMBER(5,2)** signifie 5 chiffres dont 2 après la virgule (par ex : 432,36)
- La notation décimale : -12.34
- La notation scientifique avec la lettre E : -12.3E-4 vaut  $-12,3 \times 10^{-4}$
- **Caractère : VARCHAR (longueur)**
  - **VARCHAR (10)** signifie une chaîne de 10 caractères maximum (ex : 'Chien')
  - Attention différence entre les minuscules et majuscules
- **Date : DATE**
  - Stockage interne sous forme d'un nombre en jours à partir d'une date de référence
  - Les minutes et secondes sont stockés en fraction de journée
  - Format d'affichage par défaut suivant le paramétrage de la base

# Les opérateurs : La comparaison de données

- Les symboles relationnels

|                   |               |
|-------------------|---------------|
| égal              | =             |
| inférieur         | <             |
| supérieur         | >             |
| inférieur ou égal | <=            |
| supérieur ou égal | >=            |
| différent         | <> (!= ou ^=) |

- L'ordre

| <i>TYPE DE DONNEE</i> | <i>ORDRE CORRESPONDANT</i> |
|-----------------------|----------------------------|
| NUMERIQUE             | ORDRE NATUREL DES REELS    |
| DATE                  | ORDRE CHRONOLOGIQUE        |
| CARACTERE             | ORDRE LEXICOGRAPHIQUE      |

# Les opérateurs : LIKE, IN, BETWEEN

- **LIKE**

- Syntaxe : **expr LIKE chaîne modèle**
- chaîne est une chaîne de caractères pouvant contenir l'un des caractères jokers suivants:
  - ⊙ '\_' underscore remplace 1 caractère exactement.
  - ⊙ '%' remplace une chaîne de caractères de longueur quelconque, y compris de longueur nulle.

- **IN**

- **expr1 IN (expr2, expr3 . . .)**
- Vrai si expr1 est égale à une des expressions de la liste entre parenthèses.
- NOT IN pour inverser cette condition

- **BETWEEN**

- **expr1 BETWEEN expr2 AND expr3**
- Vrai si expr1 est compris entre expr2 et expr3, bornes incluses.

## EXERCICES : Prédicats construits avec IN, BETWEEN, LIKE

- 24 : Sélectionner les employés qui sont techniciens, comptables ou vendeurs.
- 25 : Sélectionner les employés qui ne sont ni technicien, ni comptable, ni vendeur.
- 26 : Sélectionner les directeurs des services 2, 4 et 5.
- 27 : Sélectionner les subalternes qui ne sont pas dans les services 1, 3, 5.
- 28 : Sélectionner les employés qui gagnent entre 20000 et 40000 euros, bornes comprises.
- 29 : Sélectionner les employés qui gagnent moins de 20000 et plus de 40000 euros.
- 30 : Sélectionner les employés qui ne sont pas directeur et qui ont été embauchés en 88.
- 31 : Sélectionner les directeurs des services 2 ,3 , 4, 5 sans le IN.
- 32 :Sélectionner les employés dont le nom commence par la lettre M.
- 33 : Sélectionner les employés dont le nom se termine par T.
- 34 : Sélectionner les employés ayant au moins deux E dans leur nom.
- 35 : Sélectionner les employés ayant exactement un E dans leur nom.
- 36 : Sélectionner les employés ayant au moins un N et un O dans leur nom.
- 37 : Sélectionner les employés dont le nom s'écrit avec 6 caractères et qui se termine par N.
- 38 : Sélectionner les employés dont la troisième lettre du nom est un R.
- 39 : Sélectionner les employés dont le nom ne s'écrit pas avec 5 caractères.

# SELECT ORDER BY

- Les critères de tri sont spécifiés dans une clause ORDER BY figurant après la clause FROM et la clause WHERE
- La syntaxe est la suivante :
  - ORDER BY nom\_col1 | num\_col1 [DESC], [nom\_col2 | num\_col2 [DESC], ....
- Le tri se fait selon la colonne 1, les lignes ayant la même valeur dans la colonne 1 sont triées selon la colonne 2, etc...
- Pour chaque colonne le tri peut être ascendant (ASC option par défaut) ou descendant (option DESC)
- Pour simplifier, il n'est pas indispensable de nommer explicitement la colonne mais en donnant sa position dans le Select.



# EXERCICES : Maître de l'ordre

- **40** : Trier les employés (nom, prénom, n° de service, salaire) du service 3 par ordre de salaire croissant.
- **41** : Trier les employés (nom, prénom, n° de service , salaire) du service 3 par ordre de salaire décroissant.
- **42** : Idem en indiquant le numéro de colonne à la place du nom colonne.
- **43** : Trier les employés (nom, prénom, n° de service, salaire, emploi) par emploi, et pour chaque emploi par ordre décroissant de salaire.
- **44** : Idem en indiquant les numéros de colonnes.
- **45** : Trier les employés (nom, prénom, n° de service, commission) du service 3 par ordre croissant de commission.
- **46** : Trier les employés (nom, prénom, n° de service, commission) du service 3 par ordre décroissant de commission, en considérant que celui dont la commission est nulle ne touche pas de commission.

# OPérateurs Ensemblistes: LA JOINTURE

- C'est une opération qui permet d'obtenir dans une même ligne des informations provenant de plusieurs tables.
- **SELECT ... FROM table1, table2, table3, ..., tablen**
- Dans le cas de deux tables ayant une colonne en commun, on sélectionne les couples ayant une valeur commune dans cette colonne, on dit alors que l'on réalise une JOINTURE des deux tables, bien entendu cette sélection est exprimée à l'aide d'une clause WHERE.
- Lorsqu'il y a ambiguïté sur la table à laquelle appartient une colonne, il faut préfixer cette colonne avec le nom de la table (ou son alias).
- Exemple: afficher la liste des employés avec la ville dans laquelle ils travaillent.
- Pour une jointure d'une table avec elle-même, il faut utiliser des alias différents pour cette table.