**BASE DE DONNÉES LINGUISTIQUES**

**24/11 journée des talents.**

**la semaine prochaine le cours commence à 8 heure, avec le corpus arborés**

**Présentation : 3ou 10/12 les requếtes, les raisons , les tables**

**examen 17/12 examen sur papier**

( pas de la programmation, information niveau \_\_)

Enseignante : Iris Taravella

Extraction d’information ( données orales )

24/09/2021 PREMIÈRE SÉANCE (PYTHON à s’autoformer)

La 01/10 , 22/10 pas de cours

Évaluation: 2 parties

Épreuve sur papier, le dernière séance, le 17/12

Un projet par 3 ou 4, créer un base de donnée, et à présenter à l’oral.

Exercices à réaliser, et corriger en ligne.

Éditeur à choisir pour travailler dessus.

Comprendre ce que c’est bdd L et en créer.

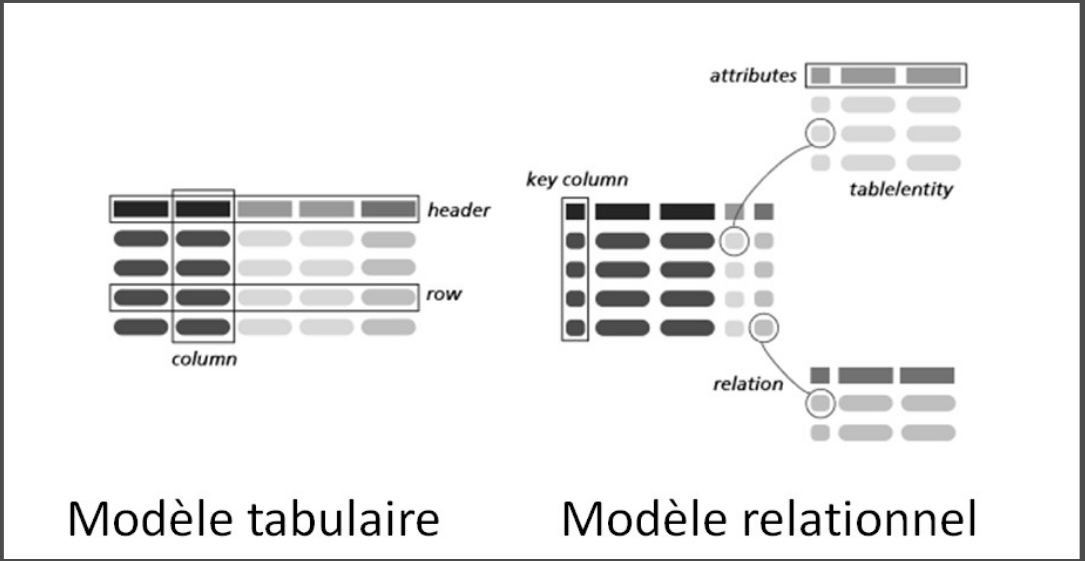
[**NEO4J-**](https://fr.wikipedia.org/wiki/Neo4j) **un peu plus compliqué, pour les gens plus informatiques**

Introduction

Nécessité de stocker des quantités de données de plus en plus importantes ( et aussi organiser

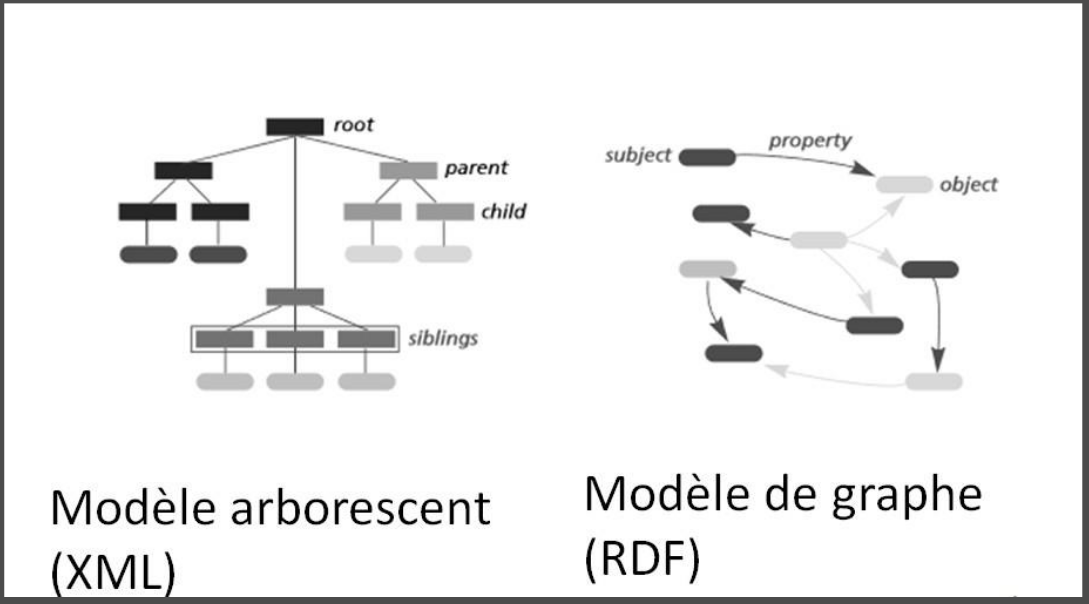
informatique

Un modèle première étape de l



Modèle Tabulaire ( tableau avec des colonnes et des lignes )

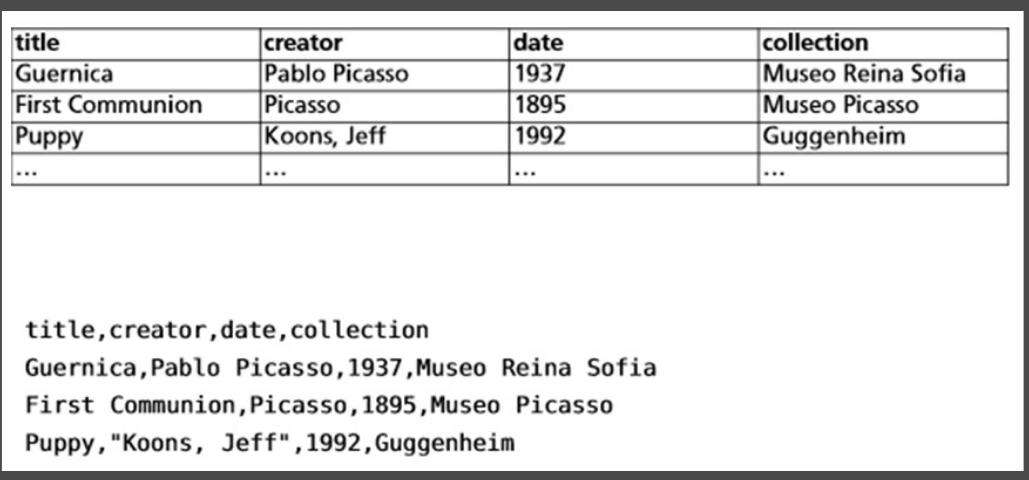
M relationnel ( plusieurs tables, mais elles ont des liens entre elles. ) --le modèle de ce cours



Modèle fondé sur un langage de balises ( modèles arborescence XML)

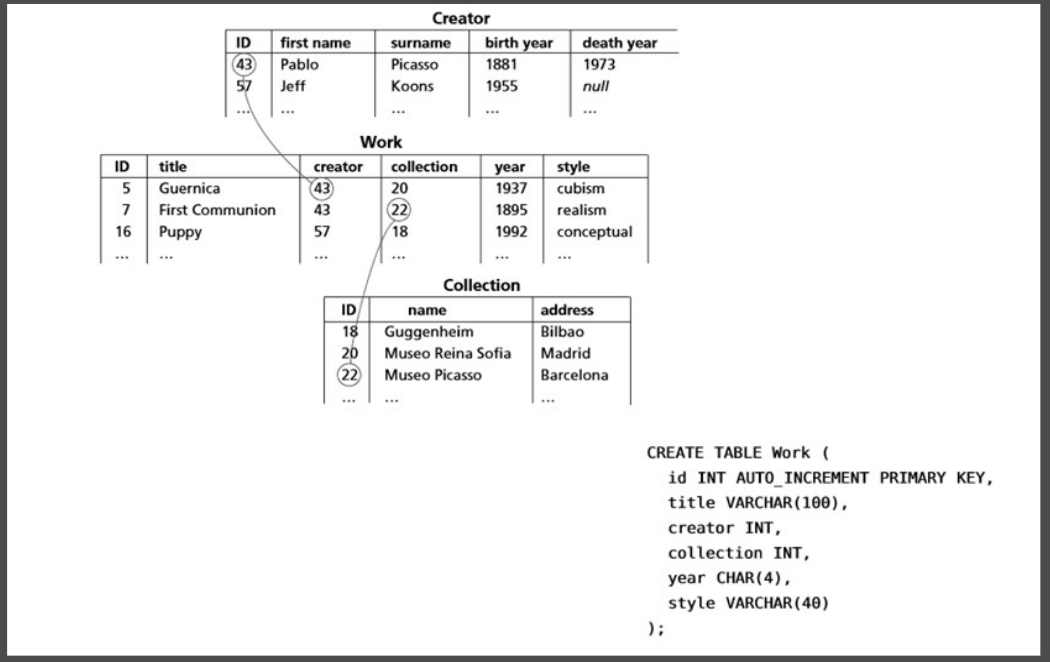
Modèle de graphe (RDF) les flèches sont étiqueté, ils ont un sens, le lien est sémantique ( modéliser les connaissance, le plus utilisé à aujourd’hui)

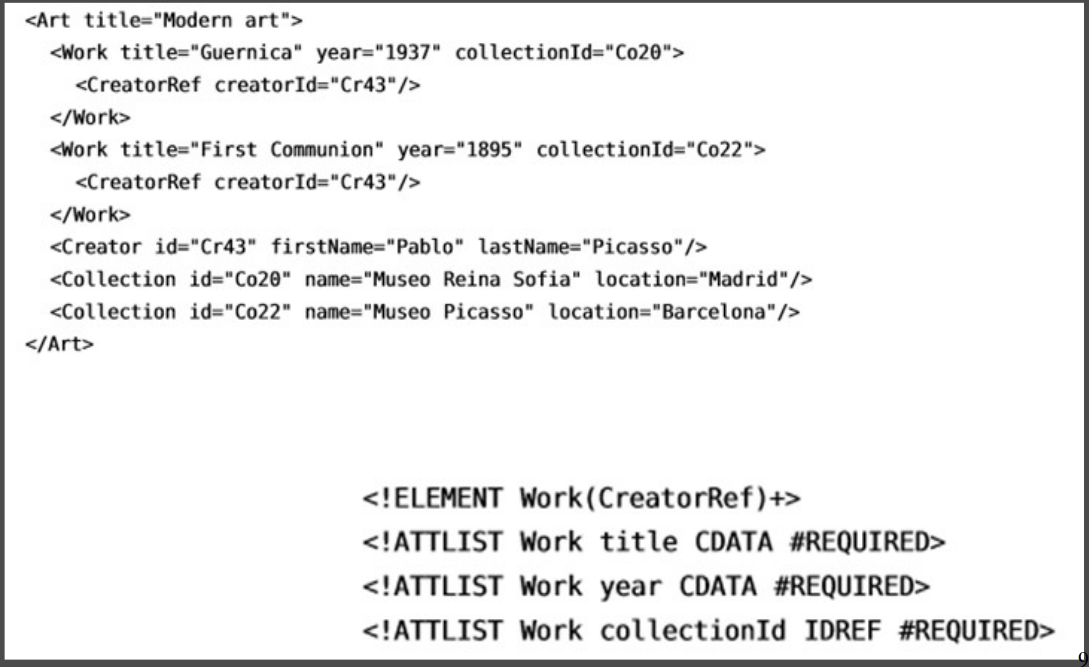
Représenter les mêmes éléments sous les différents modèls



qui a créé quel lot

forme xml





pourquoi on préfère ce modèle là, il y a document Tdd grammaire pour le document

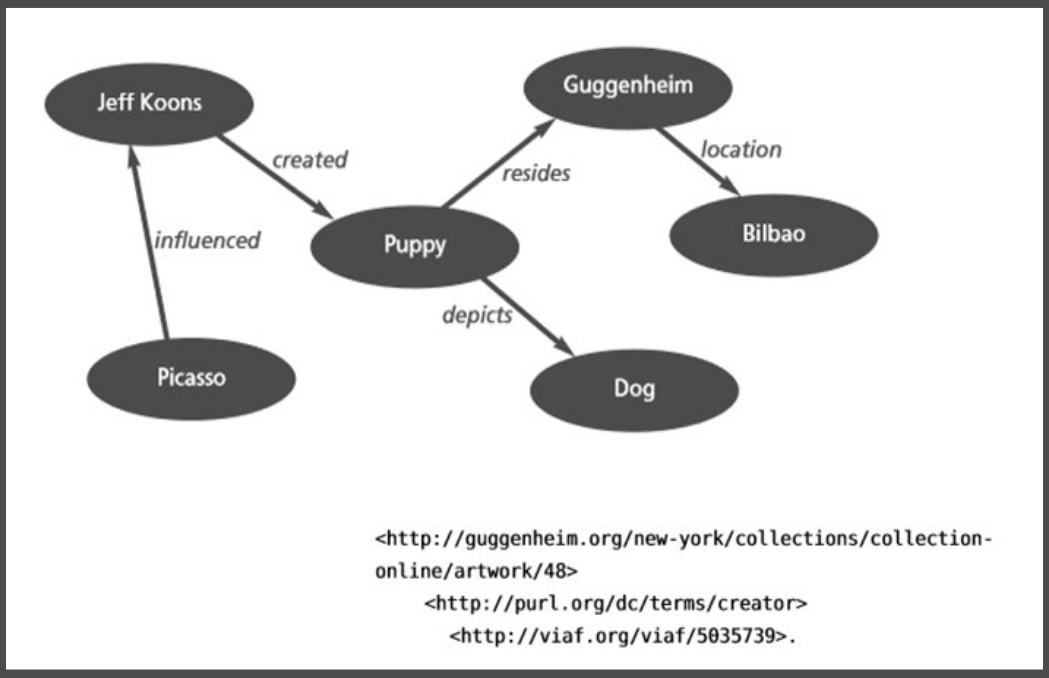
garantir l’homogénéité, textes structurés de la même manière.

Faiblesses de ces modèles

La sémantique des éléments n’est pas explicitée dans la représentation du modèle

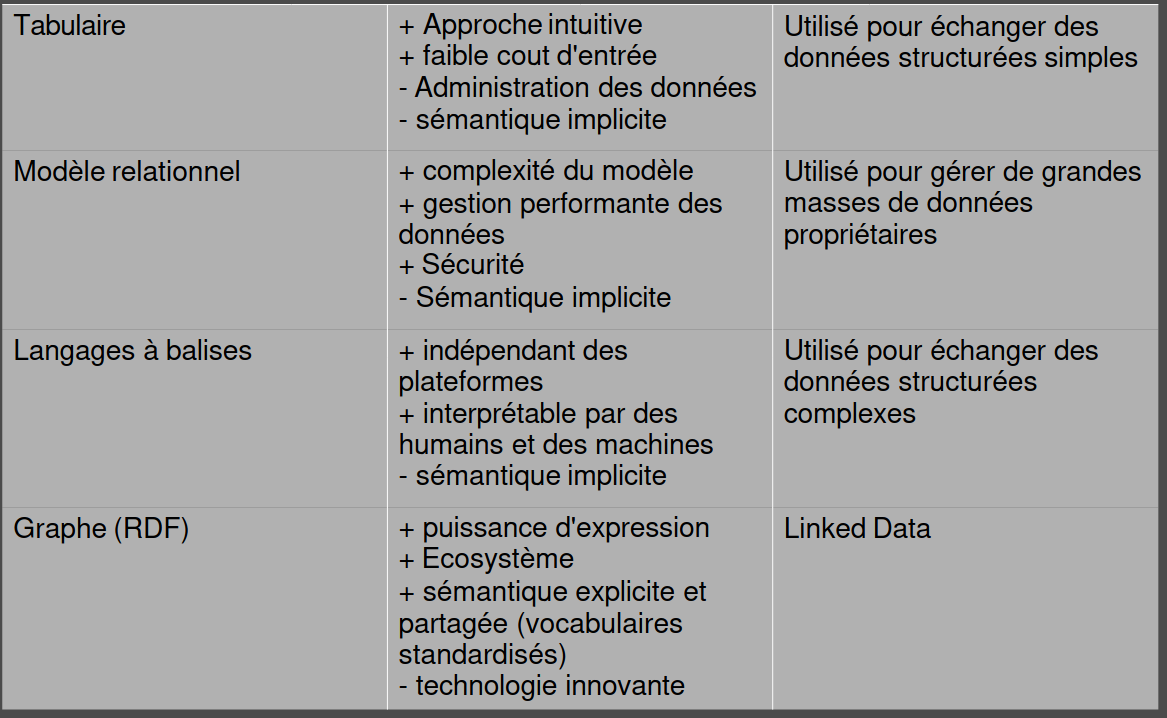
Pour avoir un sens, le RDF est plus puissant. les conceptions sont liées.

Autologie ??



Dans les entreprises, réseaux pas ouvert, favorise - Relationnel

COMPARATIF( LES + ET LES - DANS CHAQUE MODÈLE)



Définitions : Base de données

– Une base de données (BD) est un ensemble de

données sur un domaine d’application particulier

Définition

Collection de données persistantes utilisées par des systèmes informatique

système de Gestion de Base de Données

Acteurs

* Concepteur
* Administrateur des données
* Administrateur de la base
* Utilisateur final

SGBDR

Un système de gestion

Objectif d’un SGBD

PAS D’erreur

confidentialité des données

pas de répétition - non redondance

utilisation des données

* accès aux données par des non informaticiens
* accès simultanés aux même données au même moment - accès concurrents)

ONdistingue

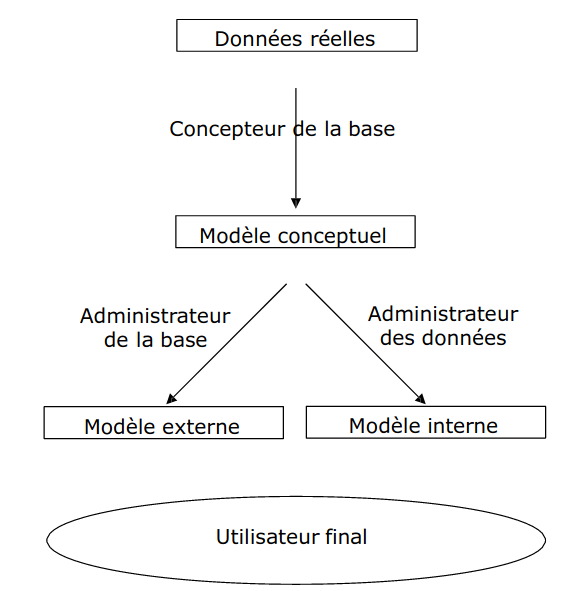
* schéma externe

TRois niveaux d’abstraction

Niveau conceptuel

Niveau interne

Niveau externe



Administrateur de bases de données ( pas un métier que tal vise)

Définit et créer le schéma conceptuel -logique à

définit et crée le schéma interne (physique)

crée les schémas externes : s'assure que les utilisateur aient accès aux donnée s dont ils ont besoins

définit les règles de sécurité

maintient l'évolution de la base

BDD- but du cours, choisir un pour le travail

Regarder les vidéos et choisir un SGBD pour après travailler dessus pour le projet.

* [MySQL](https://www.youtube.com/watch?v=Rm8KK3oM2J0)
* [PostgreSQL](https://www.youtube.com/watch?v=GI3eO14Fy90)
* pour les entreprises, pas choisir celui là ,Oracle
* [SQLite](https://www.youtube.com/watch?v=Pni6WxHFTUg), le plus difficile , pro . à regarder( mais à tenter d’installer )
* Access microsoft/ libreoffice base, pour les gens qui sont moins à l’aise avec l’informatique.

Relation; c’est une table.

Attribut: une colonne

Tuples : les lignes

Valeurs:

Clé d’une relation,

clé primaire- référent unique de la relation; clé étrangère - clé qui est pour une relation primaire mais introduite dans une autre relation) ,

exercice :

Étudiant

| Numéro | Étudiant |
| --- | --- |
| 1 | clara |
| 2 | Paul |

Cour

| Numéro | Cours |
| --- | --- |
| 1 | Français |
| 2 | Anglais |
| 3 | Musique |

Étudiant - cours ( non répétition ??)

| Étudiant | Cours |
| --- | --- |
| 1 | 2 |
| 2 | 3 |
| 2 | 1 |
| 1 | 3 |

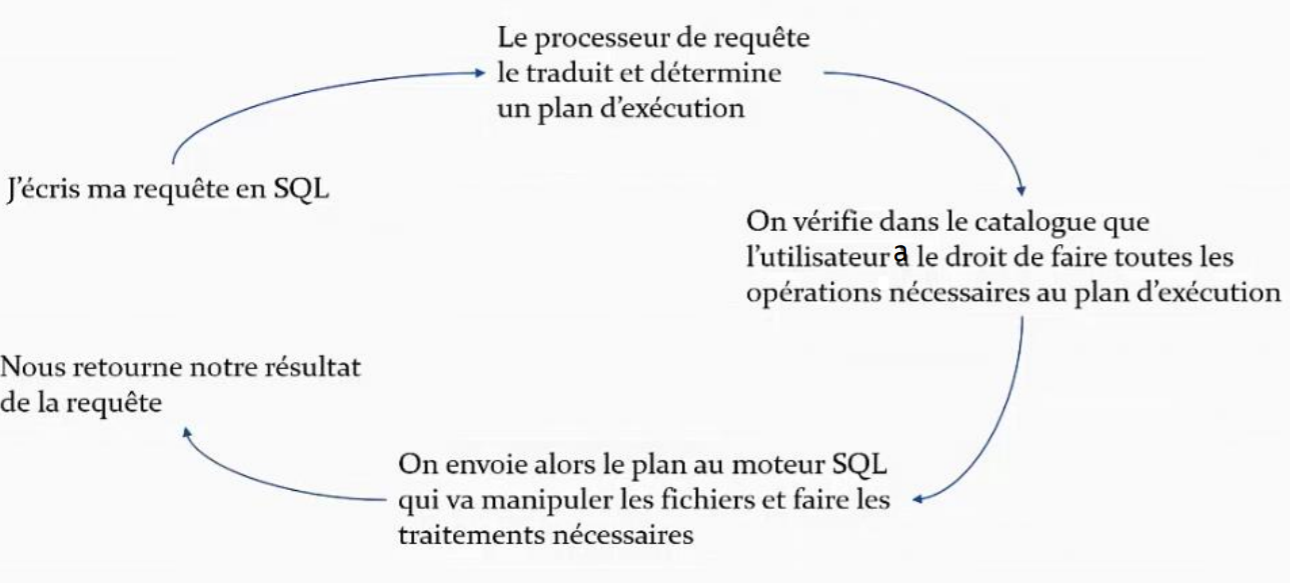
Devoir, choisir le logiciel de travail, préparer le terrain. employés ( la table à travailler dessus) SQL langage à maîtriser

Langage SQL

SGBD

* Moteur SQL
* Catalogue ( pour les utilisateurs droit d’accès)
* Langage de requête

le processsu entre je tape la requête et le résultat sort



Résultat - sous forme d’une table

Mots clés : une table ( aujourd’hui

Sélection de colonnes •

SELECT nom\_col1, nom\_col2

FROM nom\_table ;

SELECT \* (toutes les colonnes sont sélectionnées)

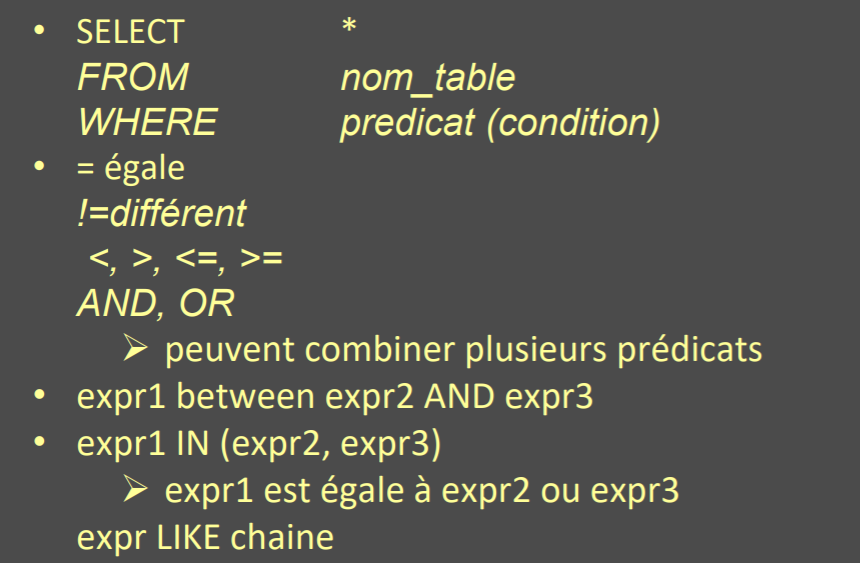
FROM nom\_table ;

SELECT DISTINCT nom\_col

FROM nom\_table ;

#permet d'éliminer les duplications, #pour éviter de sélectionner des lignes en double (redonantes)

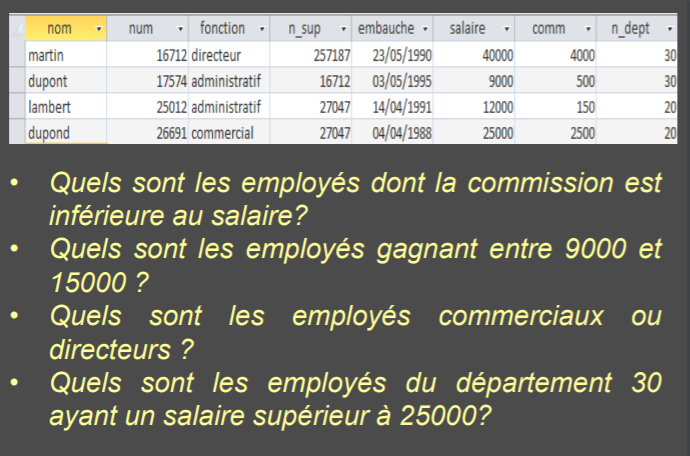
Interroger une base : sélection de lignes



ex: combiner plusieurs prédicat

LIKE , utiliser comme le signe ‘=’ devant une chaîne de caractères

Exercice en cours :



1< SELECT nom(ou \*)

FROM Nom\_de\_table

WHERE salaire > comm ;

2< SELECT nom

FROM Nom\_de\_table

WHERE salaire BETWEEN 9000 et 15000 ;

3< SELECT nom

FROM Nom\_de\_table

WHERE fonction = ‘commerciale’ OR fonction = ‘directeur’;

< SELECT nom

FROM Nom\_de\_table

WHERE fonction IN(‘commercial’,’directeur’) ;

4< SELECT nom

FROM Nom\_de\_table

WHERE n\_dept = 30 AND salaire > 25000 ;

Dans libreoffice, pas de “ mais ‘

IN est utilisé devant un ensemble des éléments, mais pas un seul.

Placer après la clause WHERE

ORDER BY Attribut DESC (par ordre décroissant)

ORDER BY Attribut ASC (par ordre croissant)

●Exemple SELECT nom\_client

FROM Clients

ORDER BY nom\_clientASC;

LIKE

motif caractères spéciaux : \*

intervalles [ ]

SELECT nom\_client

FROM Clients

WHERE nom\_client LIKE ”M\*”;

nom de client qui commence par M

SELECT numéro\_compte

FROM Comptes

WHERE numéro\_compte LIKE '[1-4]\*'; ( un numéro de compte qui commence par un chiffre entre 1 et 4.

●IN (valeur1, valeur2, . . .)

SELECT nom\_agence, adresse\_agence

FROM Agences

WHERE nom\_agence IN (‘Jussieu’, ‘Sèvres’);

Trouve les agences qui se situe à Jussier ou Sèvre

●Remarque : possibilité d'exprimer la négation avec NOT

●BETWEEN valeur1 AND valeur2

●

SELECT numéro\_compte

FROM Comptes

WHERE solde BETWEEN -50 AND 400;

Des numéro de compte qui ont un solde entre -50 et 400

●NULL Permet de vérifier qu'un attribut contient une valeur nulle (ou non nulle)

Attention, NULL n’est pas 0 car NULL c’est rien du tout mais 0 c’ets un valeur

SELECT nom\_client, adresse\_client

FROM Clients

WHERE adresse\_client IS NOT NULL

● Remarque: ne pas confondre NULL avec la valeur égale à zéro pour un attribut numérique et la chaine'' '' pour un attribut chaîne de caractères

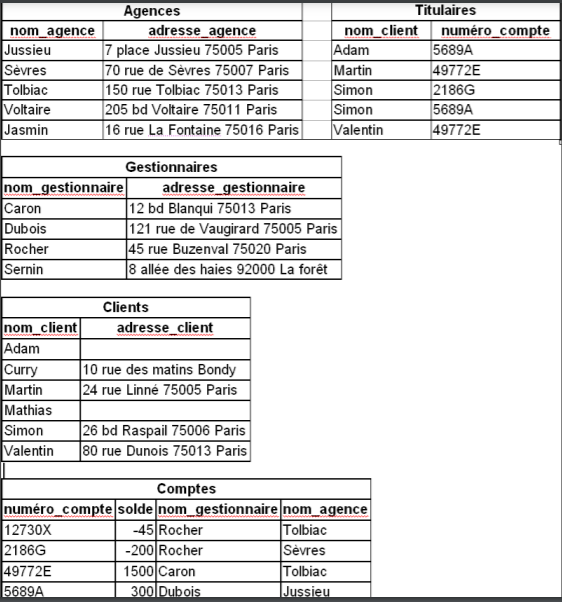
Quels sont les employés dont la commission a la valeur NULL ? ( trouver les employés à qui on n’a pas attribué de commission )

SELECT nom

FROM Nom\_de\_table

WHERE comm IS NULL ;

**TRAVAILLER SUR PLUSIEUR TABLES:**



●Q1: Liste des clients ?

SELECT nom\_client

FROM Clients ;

●Q2: Liste des clients qui ont un compte

SELECT nom\_client

FROM Titulaires

WHERE numero\_compte IS NOT NULL

●Q3: Liste des clients avec leur adresse

SELECT nom\_client, adresse\_client

FROM Clients

WHERE adresse\_clientd IS NOT NULL

●Q4: Quelle est l'adresse du client de nom 'Simon '

SELECT adresse\_client

FROM Clients

WHERE nom\_client = ‘Simon’

●Q5: Liste des comptes créditeurs (solde positif)

SELECT solde

FROM Comptes

WHERE solde > 0 ;

●Q6: Quels sont les numéros de compte de l’agence « Jussieu »

SELECT numero\_compte

FROM Compte

WHERE nom\_agence = ‘Jussieu’;

●Q7: Sélectionner dans la table Comptes les tuples dont le solde est créditeur (>0) et dont le nom d’agence est «Jussieu »

SELECT \*

FROM Comptes

WHERE nom\_agence = ‘Jussieu’ AND solde > 0 ;

**Créer une base de données et montrer les requêtes qu’on peut faire dessus. ( 4/5 tables sont suffisants)**

**DEVOIR**

**finction d’agregation etc**

**ORIGINALITé**

**Le domaine à réfléchir idée original ( médical / musical /théâtral/ FLE )**

côté commercial

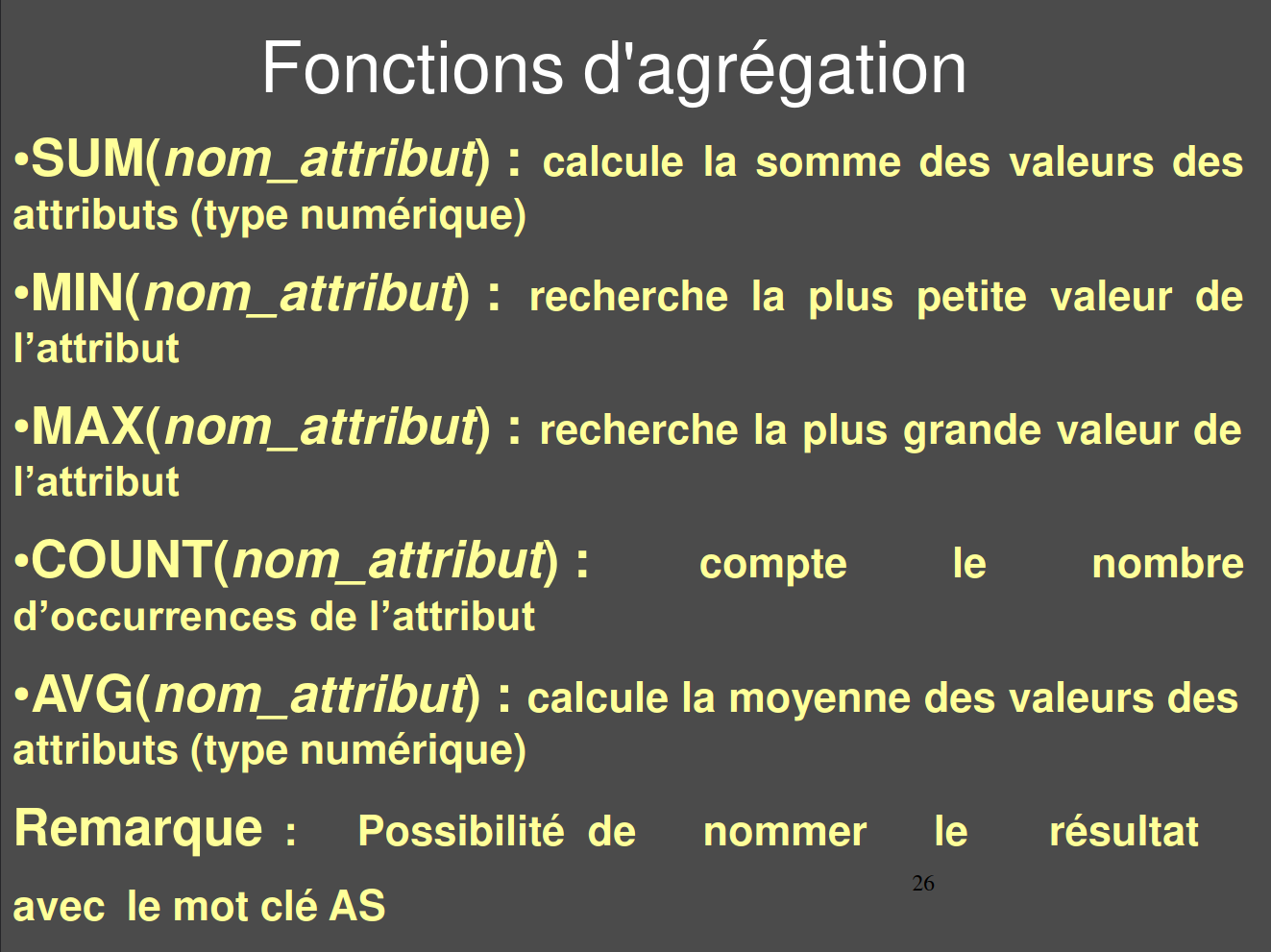
comment c’était crée étape par étapes ( le moindre détail

Les requêtes intéressants original

conculsion : ce qu’on a appris , les réflexions, résultat

ORAL important - envoyez la table

SÉANCE 3 - 15/10



Donner le total des salaires du département <”à

SELECT SUM(salaire)

FROM emp

WHERE n\_dept=30

Quel est le nombre de clients?

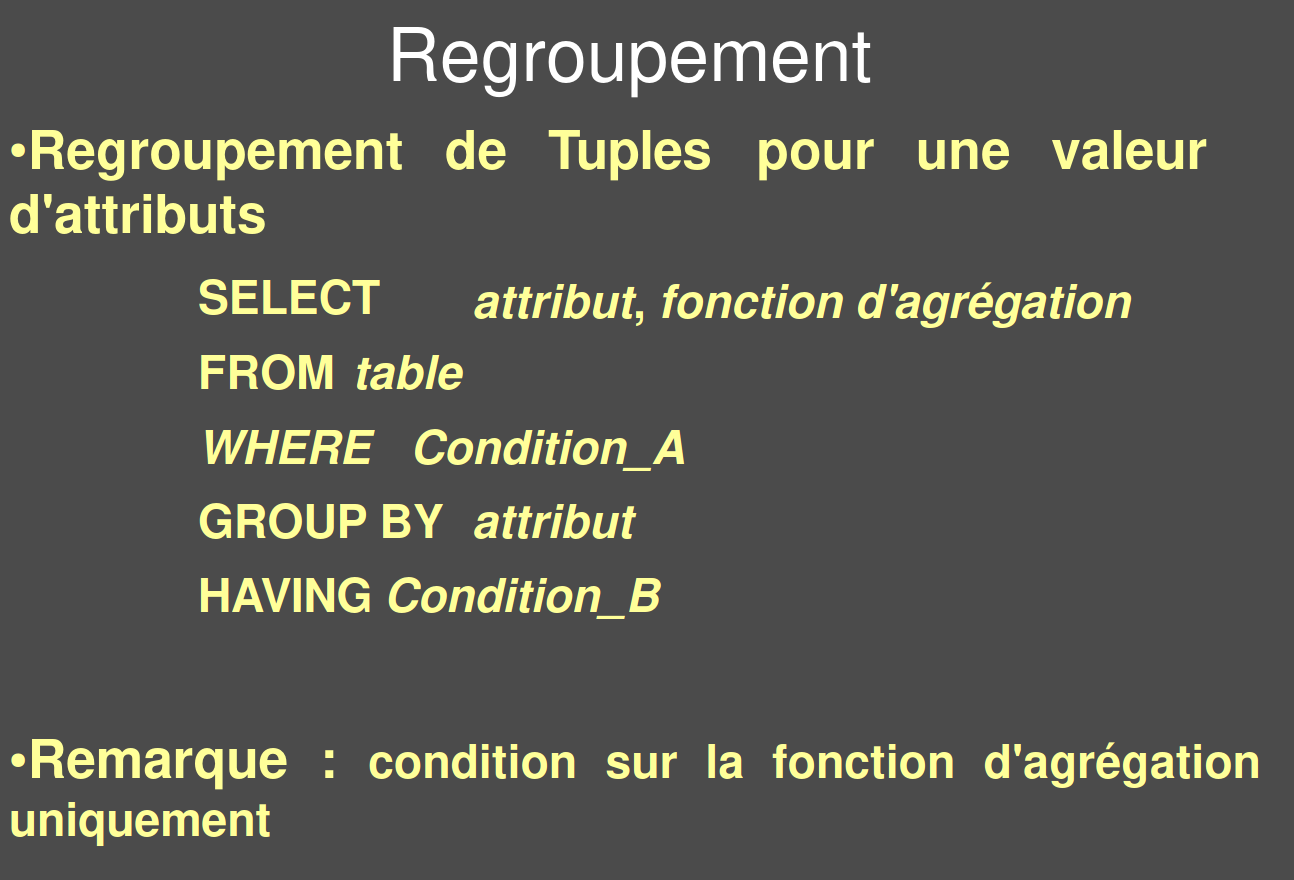
SELECT COUNT(nom\_client)

FROM Clients ;

Quel est le solde moyen de tous les comptes ?

SELECT AVG(solde)

FROM Comptes ;



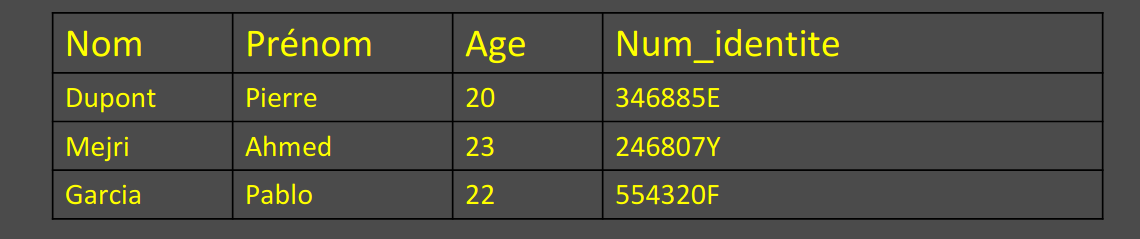
Total des salaires pour chaque département :

SELECT SUM(salaire),n\_dept

FROM EMPLOYES

GROUP BY n\_dept

On utilise HAVING quand on utilise GROUP BY et quand il y a des fonctions d’agrégation.



Liste des personnes de prénom "Pierre”

SELECT \*

FROM Personnes

WHERE Prénom LIKE “Pierre”

Liste des prénoms dont le nombre d'occurrences > 10

SELECT COUNT(Prénom), COUNT(Prénom) AS NOMBRE

FROM Personnes

GROUP BY Prénom

HAVING COUNT(Prénom)>10 ; ( HAVING NOMBRE >10)

On peut aussi pour ne pas retaper , nommer COUNT(Prénom) AS NOMBRE

Liste des prénoms dont le nombre d'occurrences > 3 pour les personnes dont le nom commence par M

SELECT COUNT (\*) AS NOMBRE

FROM Personnes

WHERE NOM LIKE ‘M%’ # Restriction sur le SELECT

GROUP BY Personnes

HAVING COUNT(\*) >3 # Restriction sur le regroupement

Quel est le solde moyen de tous les comptes de toutes les agences, regroupés par gestionnaire?

SELECT AVG(solde),nom\_gestionnaire

FROM Comptes

GROUP BY nom\_gestionnaire ;

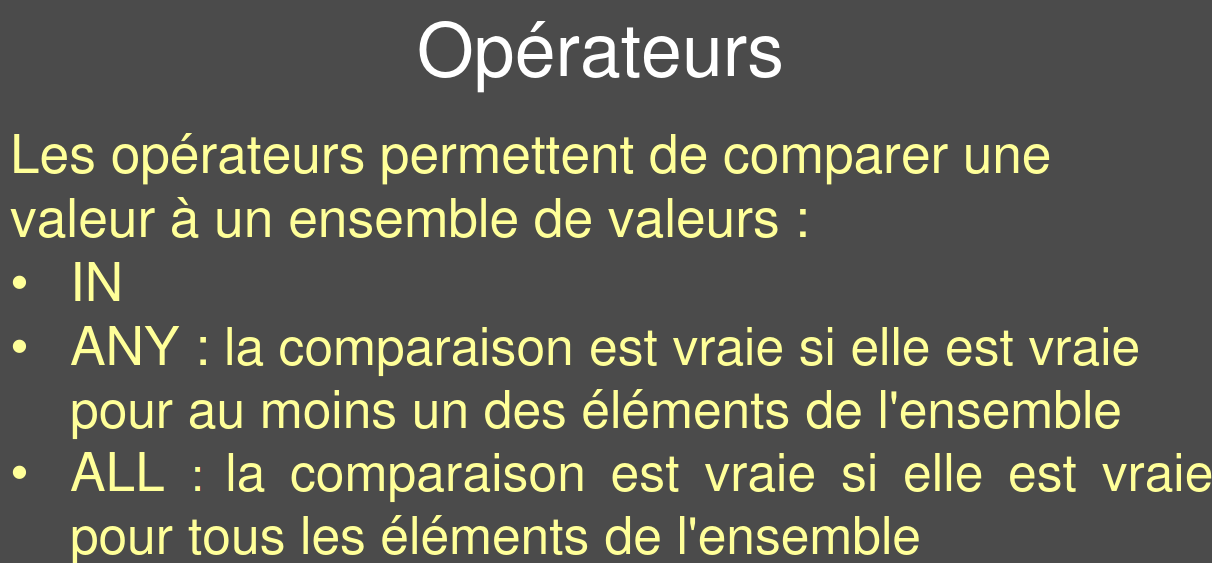
Quels sont les soldes moyens supérieurs à 300 regroupés par gestionnaire?

SELECT AVG(solde),nom\_gestionnaire

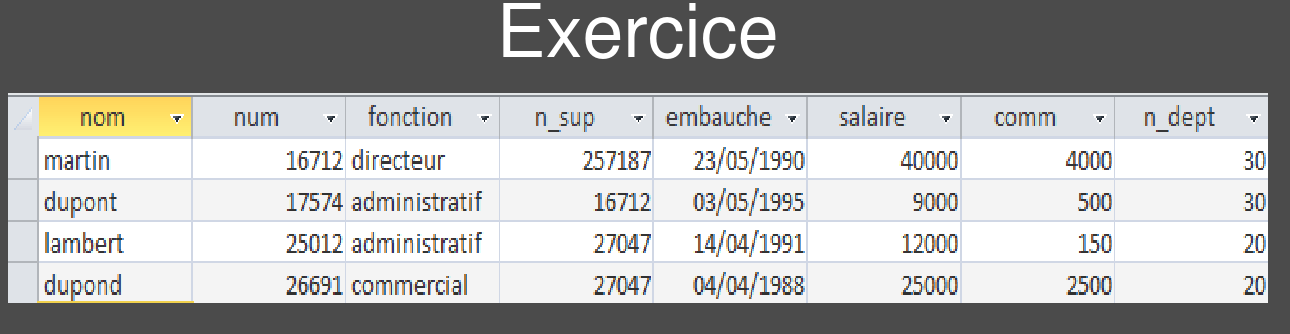
FROM Comptes

GROUP BY nom\_gestionnaire

HAVING AVG(solde)>300 ;



IN (valeur1, valeur2, . . .)



Quels sont les employés gagnant plus que tous les employés du département 30 ?

SELECT nom,salaire

FROM EMPLOYES

WHERE Salaire >

ALL (SELECT salaire

FROM EMPLOYES

WHERE n\_dept LIKE 30);

Liste des employés du départements 20 ayant la même fonction qu’un employé du département de DUPONT

SELECT nom, fonction

FROM EMPLOYES

WHERE n\_dept=20 AND fonction IN

(SELECT fonction

FROM EMPLOYES

WHERE n\_dept = (

SLECT n\_dept

FROM EMPLOYES

WHERE nom=”dupont” ));

Donner pour chaque commercial son revenu (salaire+commission)

SELECT salaire+commission, nom

FROM EMPLOYES

WHERE fonction = ‘commercial’ ;

Donner la liste des employés dont la commission est inférieur à 5% du salaire

SELECT nom, salaire, comm

FROM EMPLOYES

WHERE comm<=salaire \*.05 ;

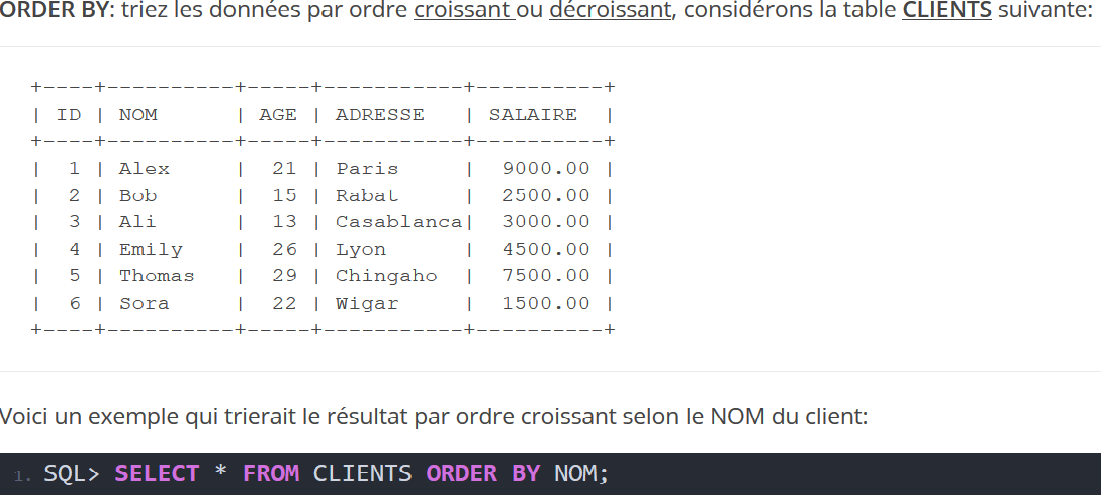
Donner la liste des administratifs classée par commission sur salaire décroissant.

SELECT nom,comm/salaire

FROM EMPLOYES

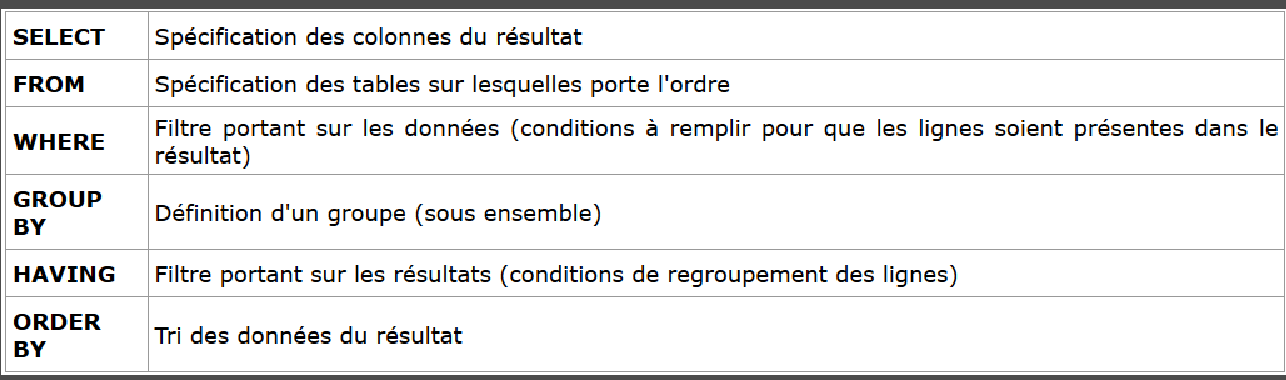
WHERE fonction = “administratif”

ORDER BY comm/salaire DESC;



Prend le dernier (ORDER BY) pour les données identique, il prend le dernier selon ordre.

GROUP BY, il fait la totalité des données des noms identiques



Obligatoires: SELECT, FROM

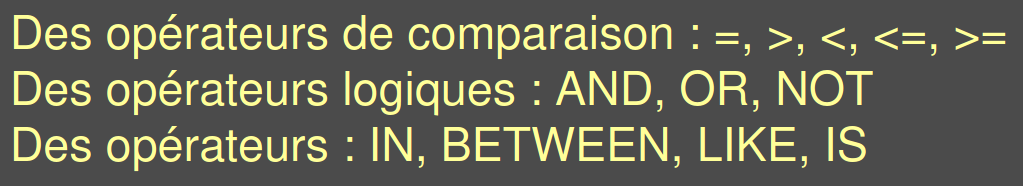
La semaine prochaine pas de cours de BDD le 19 NOVEMBRE BDD il y en a pas.

10 /12 examen en cours. Le 3/12; 26/11 - Présentation du projet

4 semaines pour finir le projet.

DISTINDCT - pas de doublons

SELECT\*FROM table WHERE condition ;



On peut diviser LIKE par un pourcendage ( si je fait ‘a%’- ceux qui commence par un ‘a’)

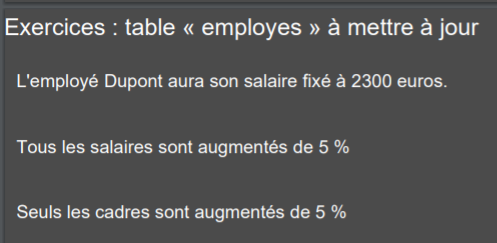
LIKE ‘%a%’ qui contient un a

LIKE ‘pa%on’ qui commence par pa et termine par on

\*= %

Plusieur valeur c’est bien d’utiliser in

WHERE date BETWEEN



1.

UPDATE EMPLOYES

SET salaire = ‘2500’

WHERE nom = ‘dupont’

2.

UPDATE EMPLOYES

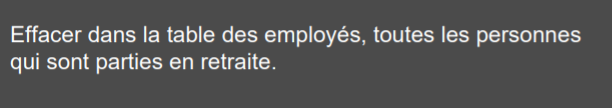
SET salaire = salaire\*1.05

3.

UPDATE EMPLOYES

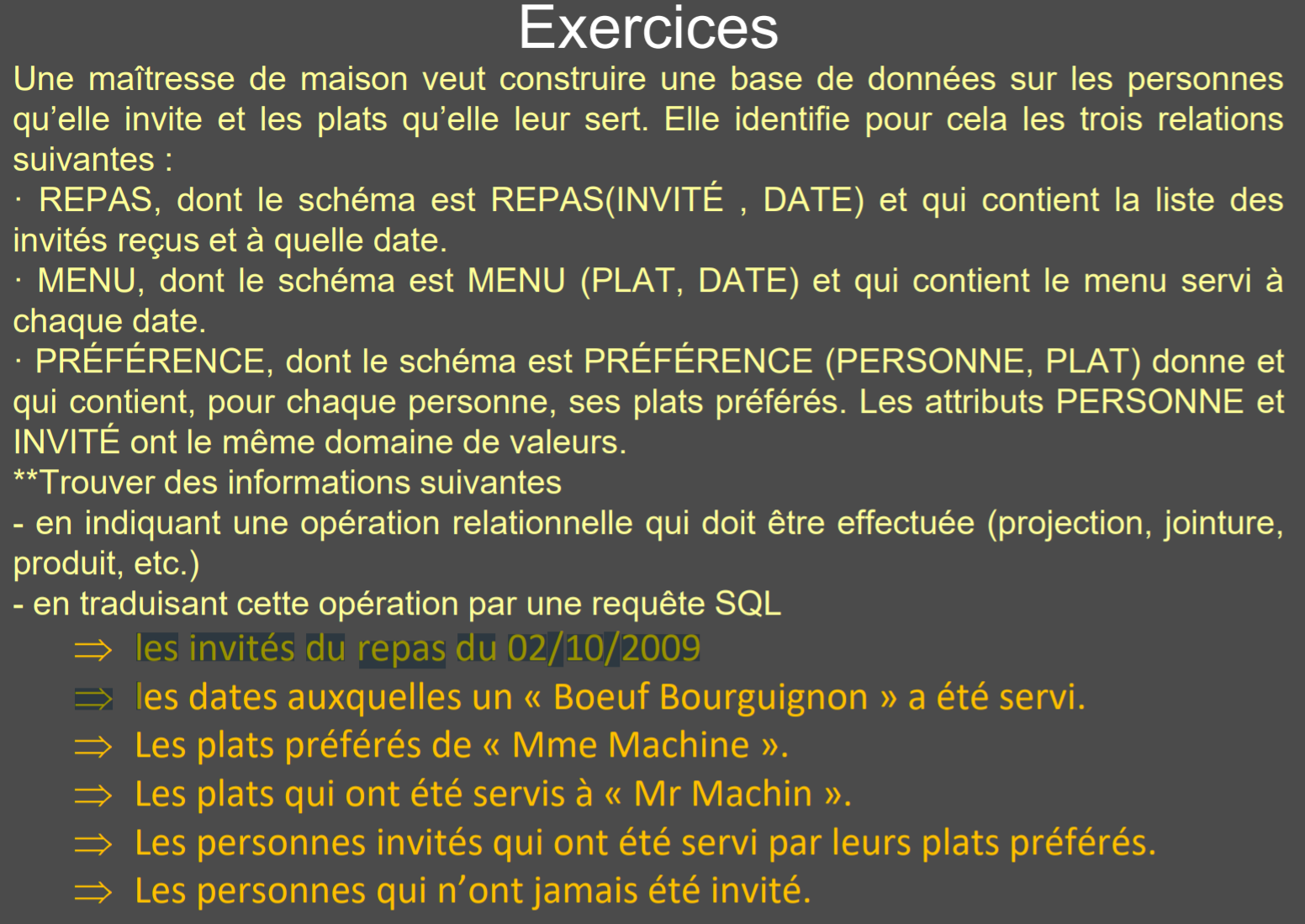
SET salaire = salaire\*1.05

WHERE fonction = ‘directeur’



DELETE FROM EMPLOYES

WHERE ???



1.

SELECT INVITE

FROM REPAS

WHERE DATA LIKE “02/10/2019”

4

SELECT PLAT

FROM MENU

WHERE DATE LIKE (SELECT DATE

FROM REPAS

WHERE INVITE LIKE ‘MR MACHIN’

5

SELECT \*

FROM PREFERENCE , MENU, REPAS

WHERE

REPAS.DATE = MENU.DATE

AND

REPAS.INVITÉ = PREFERENCE.PERSONNE

MENU.PLAT + PRÉFÉRENCE.PLAT

6

SELECT PERSONNE

FROM PRÉFÉRENCE

EXCEPT SELECT INVITE FROM REPAS ;

