SQL
Langage d'Interrogation des Données
LID

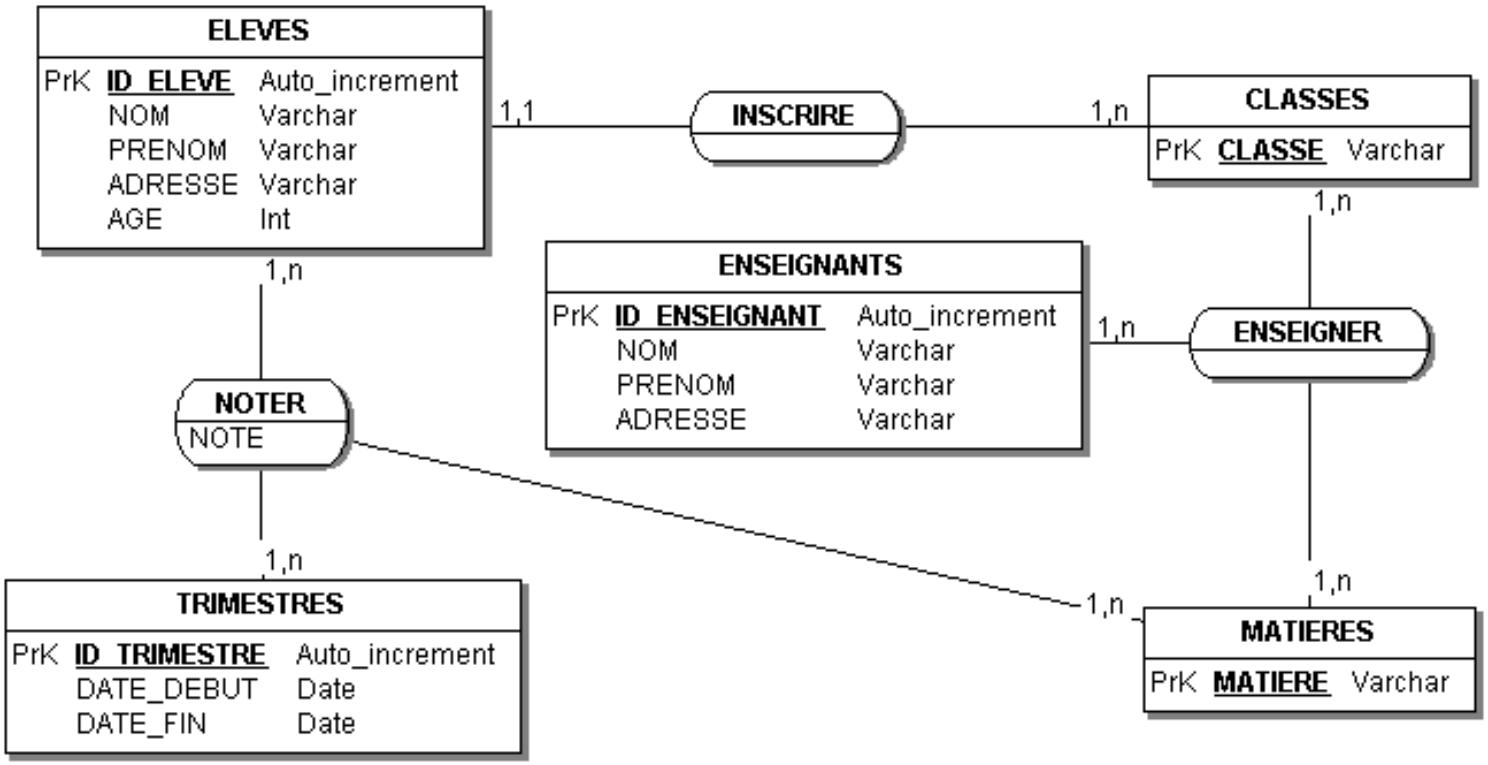
Christian Bonhomme

CNAM 2015-2016

Langage d'Interrogation des Données : LID

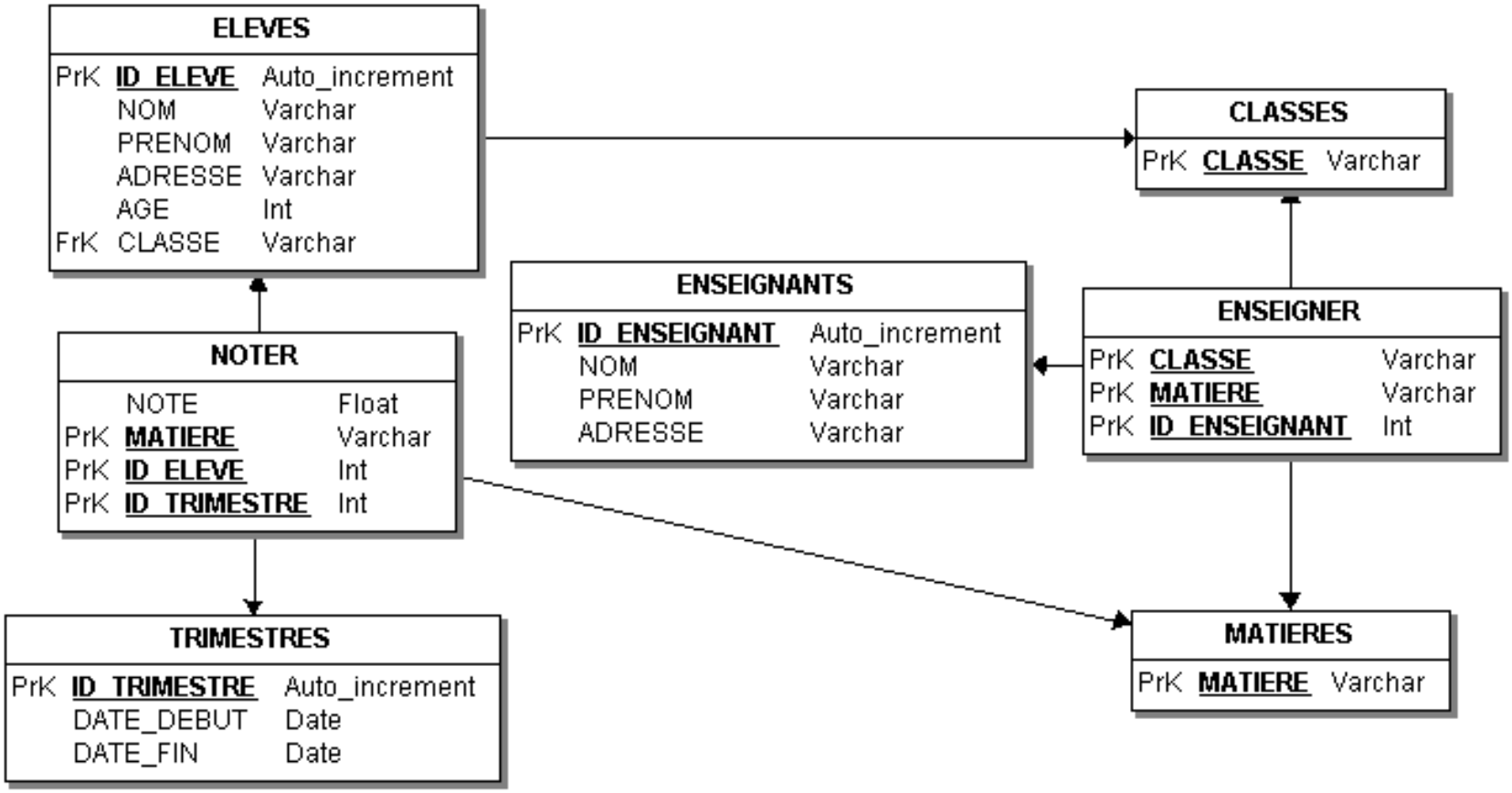
La Base de Données **ECOLES** dont voici le MCD et le MLD, nous servira d'exemple tout le long du cours sur le LID.

MCD



Langage d'Interrogation des Données : LID

MLD



LID : Algèbre relationnelle

L’algèbre relationnelle est un outil permettant de réaliser des opérations appelées requêtes sur des tables. Cet outil se compose de :

- La **projection**.
- La **restriction**.
- L'**union**.
- L'**intersection**.
- La **différence**.
- La **jointure**.
- La **division**.

LID : Projection

L'opération de projection permet de sélectionner uniquement certains attributs d'une table.

```
select  [distinct] ATTRIBUT_1, …, ATTRIBUT_N
from TABLE_NAME;
```

La clause **distinct** permet d'éliminer les doublons : si dans le résultat plusieurs lignes sont identiques, une seule sera conservée.

Le symbole * permet d'obtenir tous les attributs.

```
select  *
from TABLE_NAME;
```

LID : Projection et classement

Récupère tous les noms et prénoms des élèves :

```
select NOM, PRENOM
from ELEVES;
```

Récupère les différentes dates de début des trimestres :

```
select distinct DATE_DEBUT
from TRIMESTRES;
```

LID : Projection + order by

La clause **order by** est utilisée pour trier les tuples du résultat suivant un ou plusieurs attributs.

```
select  [distinct] ATTRIBUT_1, …, ATTRIBUT_N
from TABLE_NAME
order by ATTRIBUT_X [asc|desc], …, ATTRIBUT_Y [asc[desc];
```

Le mot-clé **order by** trie les enregistrements en ordre croissant par défaut **(asc).** Pour trier les enregistrements dans un ordre décroissant, vous pouvez utiliser le mot-clé **desc**.

LID : Projection+ order by

Récupère tous les noms et prénoms des élèves et les trie suivit l'ordre alphabétique de leur nom :

```
select NOM, PRENOM
from ELEVES
order by NOM;
```

Récupère les différentes dates de début des stages et les trie par date décroissante :

```
select distinct DATE_DEBUT
from TRIMESTRES
order by DATE_DEBUT desc;
```

LID : Restriction

La restriction ou sélection permet d'extraire d'une table les lignes ou tuples qui satisfont une ou plusieurs clauses (conditions).

```
select  [distinct] ATTRIBUT_1, …, ATTRIBUT_N
from TABLE_NAME
where <clause de recherche>
[and | or <clause de recherche>];
```

Une condition de recherche est de la forme : **ATTRIBUT1 opérateur ATTRIBUT2**

LID : Restriction - Opérateurs

Les différents opérateurs d'une condition de recherche sont :

Opérateur	Description
=	Egal
!=	Différent
<>	Différent
>	Supérieur
>=	Supérieur ou égal
<	Inférieur
<=	Inférieur ou égal

Opérateur	Description
between … and …	Entre … et …
in	Dans
like	Comme
is NULL	Indéfini
any	Au moins 1
all	Tout

LID : Restriction

Récupère les noms et prénoms des élèves dont l'âge est supérieur ou égal à 18 ans :

```
select NOM, PRENOM
from ELEVES
where AGE >= 18;
```

Récupère les prénoms des élèves dont le prénom se termine par 'el' et don l'âge est égal à 16 ans.

```
select NOM, PRENOM
from ELEVES
where PRENOM like '%el'
and AGE = 16;
```

LID : Union

L'union de deux tables TABLE1 et TABLE2 de même projection, notée **TABLE1 U TABLE2** ou **UNION(TABLE1,TABLE2)**, récupère les enregistrements appartenant aux deux tables sans doublon.

```
select  [distinct] ATTRIBUT_1, …, ATTRIBUT_N
from TABLE1
where <clause de recherche>
union
select  [distinct] ATTRIBUT_A, …, ATTRIBUT_Z
from TABLE2
where <clause de recherche>;
```

LID : Union

Récupère tous les noms et prénoms des élèves et des enseignants de l'école.

```
select  NOM, PRENOM
from ENSEIGNANTS
union
select  NOM, PRENOM
from ELEVES;
```

LID : Intersection

L'intersection de deux tables TABLE1 et TABLE2 de même projection, notée **TABLE1 ∩ TABLE2** ou **INTERSECT(TABLE1,TABLE2)**, récupère les enregistrements appartenant à la fois à TABLE1 et TABLE2.

```
select  [distinct] ATTRIBUT_1, …, ATTRIBUT_N
from TABLE1
where <clause de recherche>
intersect
select  [distinct] ATTRIBUT_A, …, ATTRIBUT_Z
from TABLE2
where <clause de recherche>;
```

LID : Différence

La différence de deux tables TABLE1 et TABLE2 de même projection, notée **TABLE1 - TABLE2** ou **MINUS(TABLE1,TABLE2)**, récupère les enregistrements appartenant à TABLE1 mais pas à TABLE2.

```
select  [distinct] ATTRIBUT_1, …, ATTRIBUT_N
from TABLE1
where <clause de recherche>
minus
select  [distinct] ATTRIBUT_A, …, ATTRIBUT_Z
from TABLE2
where <clause de recherche>;
```

LID : Jointure

La jointure permet de rapprocher les informations de plusieurs tables dont un ou des identifiants d'une table sont clefs étrangères dans une autre table.

```
select [distinct] ATTRIBUT_1, …, ATTRIBUT_N
from TABLE1, …, TABLEN
where <clause de recherche>
[and <clause de recherche>];
```

LID : Jointure

Quelles sont les matières enseignées par les enseignants ?

```
select  distinct ENSEIGNANTS.NOM, PRENOM, MATIERES.MATIERE
from MATIERES, ENSEIGNER,  ENSEIGNANTS
where MATIERES.MATIERE = ENSEIGNER.MATIERE
and ENSEIGNANTS.ID_ENSEIGNANT = ENSEIGNER.ID_ENSEIGNANT;
```

Avec alias et concaténation :

```
select  distinct concat(E.NOM, ' ', PRENOM) as Enseignants, M.MATIERE as Matières
from MATIERES M, ENSEIGNER,  ENSEIGNANTS E
where M.MATIERE = ENSEIGNER.MATIERE
and E.ID_ENSEIGNANT = ENSEIGNER.ID_ENSEIGNANT;
```

LID : Opérateurs

Opérateurs arithmétiques : +, -, *, /

```
select  ATTRIBUT_1 opérateur  nombre
from TABLE
where <clause de recherche>
[and <clause de recherche>];
```

Quelle est la note d'anglais du 3ème trimestre de l'étudiant GUITTON Francis, si l'on lui ajoute 2 points?

```
select  NOTE  + 2
from NOTER N, ELEVES E
where MATIERE = 'anglais'
and N.ID_ELEVE  = E.ID_ELEVE
and NOM = 'GUITTON'
and PRENOM = 'Francis'
and ID_TRIMESTRE = 3;
```

LID : fonctions

Fonctions arithmétiques

Fonctions les plus utilisées :

Fonction	Description
absc(n)	Valeur absolue de m
ceil(n)	Le plus entier inférieur qui dépasse n
floor(n)	Le plus grand entier inférieur à n
mod(n, m)	Reste de la division de m par n
power(n, m)	n est élevé à la puissance m
round(n, [m])	n est arrondi à m décimales (m=0 si omis)
trunc(n, [m])	n est tronqué à m décimales (m=0 si omis)
sqrt(n)	Racine carrée de n
least(n1, n2, ...)	La plus petite valeur parmi n1, n2…
greatest(n1, n2, ...)	La plus grande valeur parmi n1, n2…

LID : fonctions

Fonctions sur les chaînes de caractères

Fonctions les plus utilisées :

Fonction	Description
length(c)	Longueur de la chaîne c
upper(c)	Conversion en majuscule de la chaîne c
lower(n)	Conversion en minuscule de la chaîne c
substr(c, n , [m])	Extrait la sous-chaîne de c à partir du n-ième caractère et ayant pour longueur m
ltrim(c)	Supprime les espaces à gauche de la chaîne c
rtrim(c)	Supprime les espaces à droite de la chaîne c
concat(c, l)	Concatène la chaîne c et la chaîne l

LID : fonctions

Fonctions sur les dates

Fonctions les plus utilisées :

Fonction	Description
adddate(d, j)	Ajoute à une date d un nombre jour j
addtime(d, t)	Ajoute à une date d un temps t
date(d)	Extrait la date de la date d
datediff(d1, d2)	Soustrait la date d2 de la date d1
hour(d)	Extrait l'heure de la date d
month(d)	Extrait le mois de la date d
now()	Renvoie la date et l'heure actuelle
week(d)	Extrait la semaine de la date d
year(d)	Extrait l'année de la date d

LID : fonctions

Fonctions d'agrégation

Fonctions les plus utilisées :

Fonction	Description
count()	Compte le nombre de tuples
avg()	Calcule la moyenne d'un attribut
sum()	Calcule la somme d'un attribut
min()	Recherche la plus petite valeur d'un attribut
max()	Recherche la plus grande valeur d'un attribut
std()	Renvoie l'écart type d'un attribut
variance()	Renvoie la variance standard d'un attribut

LID : Agrégation

Quelle est la moyenne des notes de l'école ?

```
select  avg(NOTE)
from NOTER;
```

Quelle est la moyenne des notes de l'élève GUITTON Francis?

```
select  avg(NOTE)
from ELEVES E, NOTER N
where N.ID_ELEVE = E.ID_ELEVE
 and NOM = 'GUITTON'
 and PRENOM = 'Francis';
```

LID : Agrégation

Quelle est la moyenne des notes d'anglais de l'élève GUITTON Francis ?

```
select  avg(NOTE)
from ELEVES E, NOTER N, MATIERES M
where N. MATIERE = M. MATIERE
 and M. MATIERE = 'anglais'
 and N.ID_ELEVE = E.ID_ELEVE
 and E.NOM = 'GUITTON'
 and PRENOM = 'Francis';
```

LID : Agrégation

Quelle est la moyenne des notes d'anglais des élèves pour la classe 1 S2 ?

```
select  avg(NOTE)
from ELEVES E, NOTER N, MATIERES M,CLASSES C
where N.MATIERE = M.MATIERE
 and M.MATIERE = 'anglais'
 and N.ID_ELEVE = E.ID_ELEVE
 and C.CLASSE = E.CLASSE
 and C.CLASSE = '1 S2';
```

LID : Division

Il n'existe pas de commande SQL permettant de réaliser directement une division.
Cependant il est toujours possible de trouver une autre solution, notamment par l'intermédiaire des opérations de calcul et d'agrégation.

LID : Agrégation – group by

Toutes les fonctions d'agrégation prennent tout leur sens lorsqu'elles sont utilisée avec la commande **group by** qui permet de filtrer les données sur un ou plusieurs attributs.

```
select  ATTRIBUT_1, agregation (ATTRIBUT_2)
from TABLE
group by ATTRIBUT_1;
```

Où **agregation** est une fonction d'agrégation (**avg()**, **min()**, …).

LID : Agrégation – group by

Quelle est la moyenne des notes par matière :

```
select  M.MATIERE, avg(NOTE)
from NOTER N, MATIERES M
where N.MATIERE = M.MATIERE
group by M.MATIERE;
```

LID : Agrégation – group by

Quelle est la moyenne des notes par classe :

```
select  C.CLASSE, avg(NOTE)
from NOTER N, CLASSES C, ELEVES E
where N.ID_ELEVE = E.ID_ELEVE
and E.CLASSE = C.CLASSE
group by C.CLASSE;
```

LID : Agrégation – group by

Quelle est la moyenne des notes, de chaque élève pour la classe 1 S2 :

```
select  NOM, PRENOM, avg(NOTE)
from NOTER N, CLASSES C, ELEVES E
where N.ID_ELEVE = E.ID_ELEVE
and E.CLASSE = C.CLASSE
and C.CLASSE = '1 S2'
group by NOM, PRENOM;
```

LID : Agrégation – group by - having

Liée à la clause **GROUP BY**, **HAVING** permet de mettre une condition sur une fonction d'agégation ou sur un agrégat.

```
select  ATTRIBUT_1, agregation (ATTRIBUT_2)
from TABLE
group by ATTRIBUT_1
having agregation (ATTRIBUT_2)  condition valeur;
```

LID : Agrégation – group by - having

Quels sont les élèves qui n'ont pas la moyenne générale ?

```
select  E.NOM, PRENOM, avg(NOTE) as Moyenne
from ELEVES E, NOTER N
where N.ID_ELEVE = E.ID_ELEVE
group by E.NOM, PRENOM
having Moyenne < 10;
```

LID : Agrégation – group by - having

Quels sont les élèves de la classe 1 S2 qui ont une moyenne en anglais supérieure ou égale à 12 ?

```
select  E.NOM, PRENOM, avg(NOTE)
from ELEVES E, NOTER N, MATIERES M, CLASSES C
where N.ID_ELEVE = E.ID_ELEVE
and N.MATIERE = M.MATIERE
and M.MATIERE = 'anglais'
and C.CLASSE = E.CLASSE
and C.CLASSE= '1 S2'
group by E.NOM, PRENOM
having avg(NOTE) >= 12 ;
```

LID : Requêtes imbriquées

Un requête imbriquée est composée de deux ou plusieurs **select**.

Le premier est appelé requête principale, le ou les suivants, sous-requêtes.

L'exécution se fait en deux temps :

- les sous-requêtes extraient les valeurs intermédiaires ;
- la requête principale s'exécute sur les valeurs intermédiaires.

Le lien entre deux **select** est réalisé par :

- **in** : si la sous-requête fournit plusieurs valeurs ;
- **=** : si la sous-requête ne fournit qu'une seule valeur.

LID : Requêtes imbriquées

Quels élèves habitent chez un enseignant de cette école ?

```
select  NOM, PRENOM
from ENSEIGNANTS
where ADRESSE in (SELECT ADRESSE
                  from ELEVES);
```

LID : Requêtes imbriquées

Quels sont les enseignant qui n'enseignent pas dans la classe 1 S2 ?

```
select  NOM, PRENOM
from ENSEIGNANTS
where (NOM, PRENOM) not in
        (select NOM, PRENOM
         from ENSEIGNANTS  E1, ENSEIGNER E2,  CLASSES C
         where E2.ID_ENSEIGNANT = E1. ID_ENSEIGNANT
         and E2.CLASSE = C.CLASSE
         and C.CLASSE = '1 S2');
```

LID : bibliographie

Documentation officielle MySQL 5.7

http://dev.mysql.com/doc/refman/5.7/en/sql-syntax-data-definition.html

w3schools.com

http://www.w3schools.com/sql/default.asp