

# Bases de Données

## Module : Requêtes SQL Select

- Evaluation Individuelle
  - durée 1H30
  - jeudi 21 novembre 2023 après midi
- Professeur : David MOTSCH
- Mail : motsch@3il.fr



# Bases de Données

- Compétences visées :
  - Etre capable de :
    - Rechercher des données en optimisant les requêtes
    - Faire des calculs sur les données
      - Sur toutes les tables
      - Sur des sous-ensembles des tables
    - Imbriquer des recherches
  - Tout en connaissant le vocabulaire des bases de données et en sachant expliquer à un utilisateur néophyte



# Bases de Données

## ➤ Définition :

- Bases de données relationnelles (BD)
- SGBD (MySQL, Oracle, PostgreSQL, SQLite...)
- SQL (Structured Query Language)
- Tables – Entités – Relations
- Colonne – Attribut – Champ
- Ligne – Enregistrement – Tuple
- Langage de définition de données (LDD) (Table, Contrainte...)
- Langage de manipulation de données (LMD) (Select, CRUD)
- Langage de contrôle de données (LCD) (User, Role, Droit...)
- Sélection de données
- Jointure interne, externe et Produit cartésien
- Union – Intersection – Minus (Ensemble)

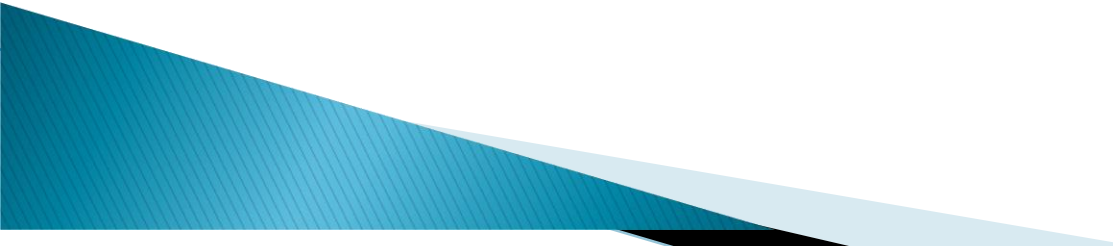
# Bases de Données

## Exemple Bases de données

ELEVES     (**Numele**, Nom, Prenom, DateCreation)  
MATIERES   (**Nummat**, Libelle, Coeff)  
NOTES      (**Numele, Nummat**, Note)

## Création table

```
CREATE TABLE nom_table (  
col1 type [constraint],  
col2 type [constraint],  
...  
[constraint nom_contrainte lacontrainte]);
```



# Bases de Données

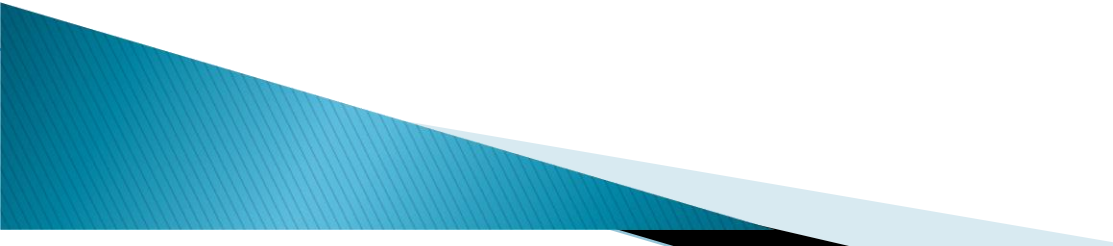
## Liste des types données

VARCHAR2 (taille),  
NUMBER [(taille[,precision])],  
CLOB,  
DATE  
...

# Bases de Données

## Création table ELEVE

```
CREATE TABLE ELEVES (  
  Numele NUMBER(3) not null,  
  Nom VARCHAR2(30) not null,  
  Prenom VARCHAR2(30) not null,  
  DateCreation DATE default trunc(sysdate),  
  CONSTRAINT pk_eleve PRIMARY KEY (Numele)  
);
```



# Bases de Données

## Les jointures internes

```
SELECT nom des colonnes  
FROM table1, table2, ...  
  
WHERE conditions sur les colonnes  
GROUP BY nom des colonnes  
HAVING conditions sur les calculs  
ORDER BY nom des colonnes ;
```

```
SELECT nom des colonnes  
FROM table1 INNER JOIN table2  
ON condition de jointure  
WHERE conditions sur les colonnes  
GROUP BY nom des colonnes  
HAVING conditions sur les calculs  
ORDER BY nom des colonnes ;
```



## Les jointures externes

- Pour exprimer une jointure externe on se base sur la syntaxe INNER JOIN en utilisant à la place LEFT OUTER JOIN, LEFT JOIN ou RIGHT OUTER JOIN.
- ```
Select nom, prenom, notes from ELEVES e left outer join NOTES n on e.numele = n.numele;
```

=
- ```
Select nom, prenom, notes from ELEVES, NOTES where e.numele = n.numele(+);
```

# Bases de Données

## Le produit cartésien

- Pour exprimer un produit cartésien on se base sur la syntaxe `INNER JOIN` en utilisant à la place `CROSS JOIN`.

- `Select nom, prenom, notes from ELEVES e cross join NOTES n;`

=

- `Select nom, prenom, notes from ELEVES, NOTES;`

# Bases de Données

## Calcul

`COUNT(*) = COUNT(Colonne) Count (Distinct Colonne)`

`MIN(Colonne) MAX(Colonne)`

`SUM(Colonne) AVG(Colonne)`

Exemple :

`Select count(*) from eleves;`

`Select max(note) from notes;`



# Bases de Données

## Fonctions sur les ensembles

UNION – INTERSECT – MINUS

IN – EXISTS – LIKE

Exemple :

```
Select nom, prenom from eleve where numele = 1
```

```
UNION Select nom, prenom from eleve where numele = 2;
```

=

```
Select nom, prenom from eleve where numele in (1, 2);
```



# Bases de Données

## Autres fonctions

NVL – DECODE – CASE WHEN – LISTAGG – TO\_CHAR ...

Exemple :

```
Select Nom, Prenom, TO_CHAR(nvl(DateCreation, sysdate),  
    'DD/MM/YYYY')  
from ELEVES;
```

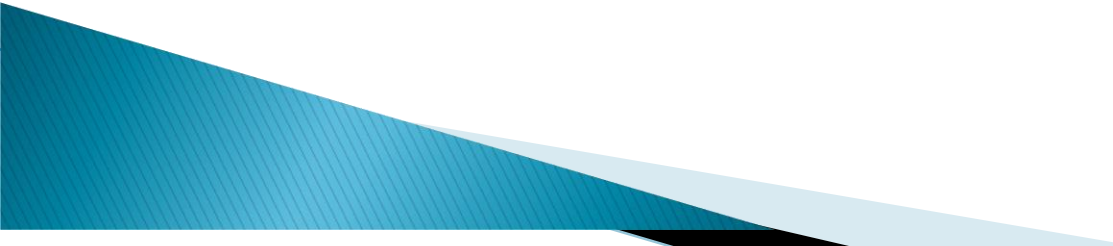
```
Select LISTAGG(Nom, ', ' within group (order by Nom  
    desc) from ELEVES;
```

# Bases de Données

## ➤ Exemples

1. Afficher la matière qui a le coefficient le plus élevé.
2. Afficher pour la matière dont le nummat est 3, le nom et prénom des élèves ainsi que leur note. **On affichera tous les élèves..**
3. Afficher pour chaque élève, leur moyenne générale en tenant compte des coefficients.
4. Afficher le nom et prénom des élèves qui n'ont pas de note dans la matière N°3.

# Bases de Données

5. Afficher le nom et prénom des élèves qui ont une note dans la matière N°2 et dans la matière N°3.
  6. Afficher le nom et prénom de l'élève qui a la meilleur moyenne en tenant compte des coefficients.
  7. Afficher pour chaque matière, le nom et prénom du meilleur élève.
  8. Afficher pour chaque matière, le nom et prénom des absents.
- 

# Bases de Données

Opérations de mises à jour :

- **INSERT** *Insérer des enregistrement*
- **DELETE** *Supprimer des enregistrement*
- **UPDATE** *Modifier des valeurs dans les enregistrement*

Transaction : Commit / Rollback

- **Commit** *Validation des modifications*
- **Rollback** *Annulation des modifications*