

# Exploitation d'une Base de Données

## Cours 1

Anaïs Durand

24 Janvier 2023

# Organisation du module

- ▶ CM : 1h toutes les 2 semaines
- ▶ TP : 2h par semaine
- ▶ Quizz sur Moodle : environ 1 toutes les 2 semaines

## *Enseignants :*

- ▶ Groupe 1-2 : Anaïs Durand
- ▶ Groupe 3-4 : Julie Rossignol
- ▶ Groupe 5-6 : Franck Glaziou
- ▶ Groupe 7-8 : Raphaël Delage
- ▶ Groupe 9-10 : Jules Azemar

# Organisation du module

## *Notions abordées :*

- ▶ SQL avancé (sous-requêtes, **GROUP BY**, ...)
- ▶ Visualisation de données
- ▶ PL/pgSQL
- ▶ Gestion des droits

# Organisation du module

## *Notions abordées :*

- ▶ SQL avancé (sous-requêtes, **GROUP BY**, ...)
- ▶ Visualisation de données
- ▶ PL/pgSQL
- ▶ Gestion des droits

## *Evaluation :* 2 examens

- ▶ semaine 8
- ▶ semaine 16

Début :  $\approx$  mi Mars

### *Objectifs :*

- ▶ Manipulation de jeux de données libres
- ▶ Visualisation de données
- ▶ Statistiques

### *Evaluation :*

- ▶ Livrables
- ▶ Soutenance en anglais

# Rappels

# SELECT

```
SELECT [DISTINCT] ...  
[FROM ...]  
[WHERE ... [(SELECT ... )]]  
[ORDER BY ...];
```

# SELECT

```
SELECT [DISTINCT] ...  
[FROM ...]  
[WHERE ... [(SELECT ... )]]  
[ORDER BY ...];
```

## *Exemple.*

```
SELECT e.nom, e.prenom  
FROM Employe e, Service s  
WHERE e.noService=s.noService  
      AND s.idChef NOT IN (SELECT idEmploye  
                           FROM Employe  
                           WHERE salaire >= 3000)  
ORDER BY e.nom;
```

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `count(*)` : compte les lignes

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `count(*)` : compte les lignes
- ▶ `count(col)` : compte les lignes où `col`  $\neq$  `NULL`

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `count(*)` : compte les lignes
- ▶ `count(col)` : compte les lignes où `col`  $\neq$  `NULL`
- ▶ `count(DISTINCT col)` : compte les valeurs distinctes  $\neq$  `NULL` pour `col`

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `count(*)` : compte les lignes
- ▶ `count(col)` : compte les lignes où `col`  $\neq$  `NULL`
- ▶ `count(DISTINCT col)` : compte les valeurs distinctes  $\neq$  `NULL` pour `col`
- ▶ `sum(col)` : somme des lignes

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `avg(col)` : moyenne des lignes

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `avg(col)` : moyenne des lignes
- ▶ `max(col)` : maximum des lignes

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `avg(col)` : moyenne des lignes
- ▶ `max(col)` : maximum des lignes
- ▶ `min(col)` : minimum des lignes

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `avg(col)` : moyenne des lignes
- ▶ `max(col)` : maximum des lignes
- ▶ `min(col)` : minimum des lignes
- ▶ `stddev(col)` : écart-type des lignes

# Agrégation

Permet de faire un calcul statistique sur toutes les lignes sélectionnées.

- ▶ `avg(col)` : moyenne des lignes
- ▶ `max(col)` : maximum des lignes
- ▶ `min(col)` : minimum des lignes
- ▶ `stddev(col)` : écart-type des lignes
- ▶ `variance(col)` : variance des lignes

*Exemple.* Salaire minimum, maximum et moyen dans l'entreprise :

```
SELECT min(salaire), max(salaire), avg(salaire)
FROM Employe;
```

*Exemple.* Salaire minimum, maximum et moyen dans l'entreprise :

```
SELECT min(salaire), max(salaire), avg(salaire)
FROM Employe;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

# Agrégation

*Exemple.* Salaire minimum, maximum et moyen dans l'entreprise :

```
SELECT min(salaire), max(salaire), avg(salaire)
FROM Employe;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

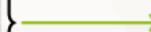


min	max	avg
?	?	?

*Exemple.* Salaire minimum, maximum et moyen dans l'entreprise :

```
SELECT min(salaire), max(salaire), avg(salaire)
FROM Employe;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400



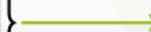
min	max	avg
1500	?	?

# Agrégation

*Exemple.* Salaire minimum, maximum et moyen dans l'entreprise :

```
SELECT min(salaire), max(salaire), avg(salaire)
FROM Employe;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400



min	max	avg
1500	3800	2250

# GROUP BY & HAVING

# GROUP BY

Permet de faire des calculs statistiques par “paquets” de lignes.

*Exemple.* Salaire moyen par service

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService;
```

# GROUP BY

Permet de faire des calculs statistiques par “paquets” de lignes.

*Exemple.* Salaire moyen par service

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

# GROUP BY

Permet de faire des calculs statistiques par “paquets” de lignes.

*Exemple.* Salaire moyen par service

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

noService	moy
S01	
S02	
S03	
S04	

# GROUP BY

Permet de faire des calculs statistiques par “paquets” de lignes.

*Exemple.* Salaire moyen par service

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400



noService	moy
S01	
S02	
S03	
S04	

# GROUP BY

Permet de faire des calculs statistiques par “paquets” de lignes.

*Exemple.* Salaire moyen par service

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400



noService	moy
S01	2000
S02	
S03	
S04	

# GROUP BY

Permet de faire des calculs statistiques par “paquets” de lignes.

*Exemple.* Salaire moyen par service

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400



noService	moy
S01	2000
S02	1800
S03	
S04	

# GROUP BY

Permet de faire des calculs statistiques par “paquets” de lignes.

*Exemple.* Salaire moyen par service

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

noService	moy
S01	2000
S02	1800
S03	3800
S04	2300

# HAVING

Impossible de filtrer les lignes en fonction du résultat d'une agrégation dans le **WHERE** **MAIS** possible dans le **HAVING**.

*Exemple.*

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService
HAVING min(salaire) >= 2000;
```

# HAVING

Impossible de filtrer les lignes en fonction du résultat d'une agrégation dans le **WHERE** **MAIS** possible dans le **HAVING**.

*Exemple.*

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService
HAVING min(salaire) >= 2000;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

noService	moy
S01	2000
S02	1800
S03	3800
S04	2300

# HAVING

Impossible de filtrer les lignes en fonction du résultat d'une agrégation dans le **WHERE** **MAIS** possible dans le **HAVING**.

*Exemple.*

```
SELECT noService, avg(salaire) moy
FROM Employe
GROUP BY noService
HAVING min(salaire) >= 2000;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

noService	moy
S01	2000
S02	1800
S03	3800
S04	2300

# HAVING

Impossible de filtrer les lignes en fonction du résultat d'une agrégation dans le **WHERE** **MAIS** possible dans le **HAVING**.

*Exemple.*

```
SELECT noService , avg(salaire) moy
FROM Employe
GROUP BY noService
HAVING min(salaire) >= 2000;
```

noEmploye	noService	salaire
E0001	S01	2000
E0002	S01	1500
E0003	S01	2500
E0004	S02	1900
E0005	S02	1700
E0006	S03	3800
E0007	S04	2200
E0008	S04	2400

noService	moy
<del>S01</del>	<del>2000</del>
<del>S02</del>	<del>1800</del>
S03	3800
S04	2300