



UNIVERSITÉ
DE REIMS
CHAMPAGNE-ARDENNE

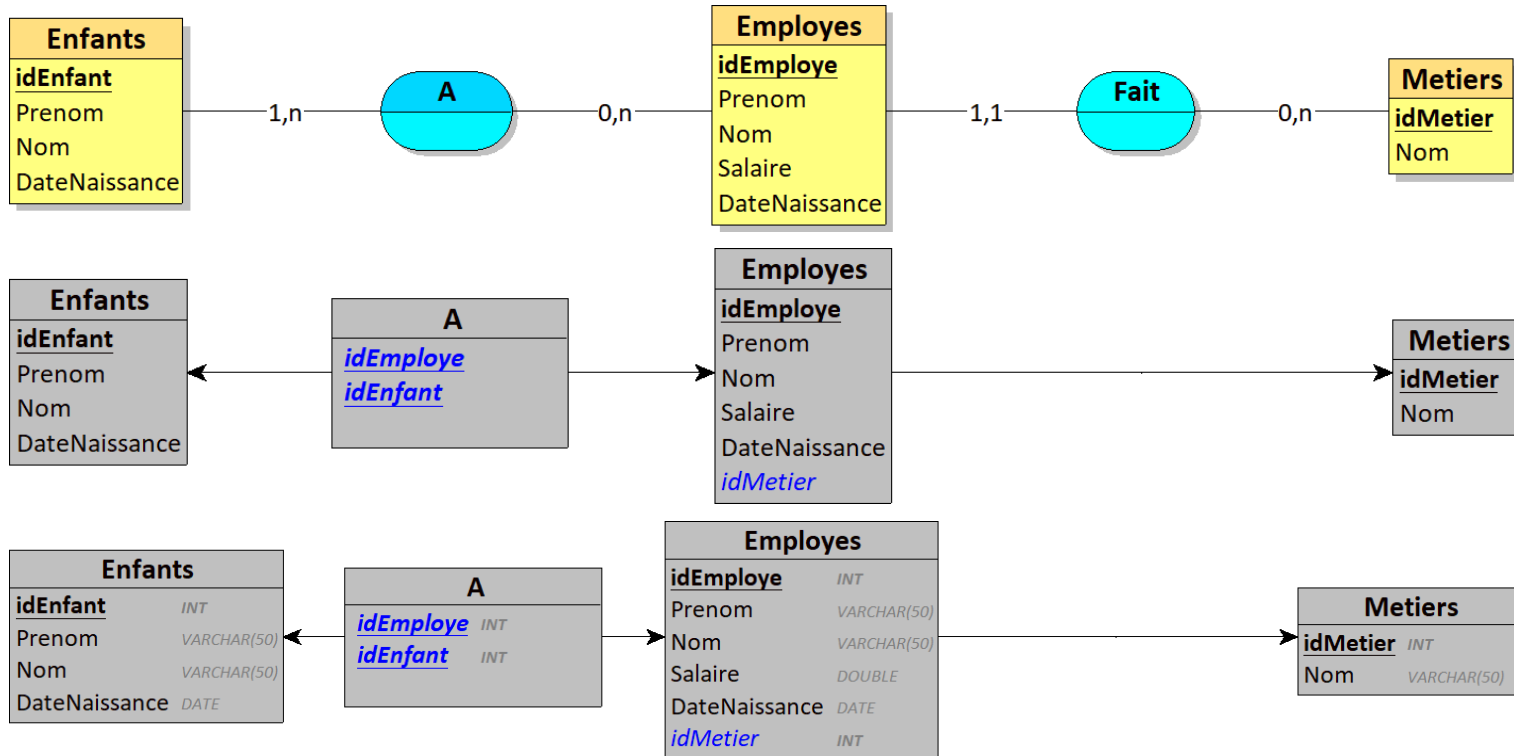
SQL – Partie 1

Nils Schaefer

nils.schaefer@snicw.fr

Base de données d'exemple

- Voici la base de données qui sera utilisée dans ce cours à titre d'exemple



- Lien pour accéder aux tables et tester les requêtes SQL du cours

- https://www.sniciw.fr/r/bd/bd.php?bd=c0_exemple8_BD

Concept de requête

- ❑ Lire des données dans la base de données
 - Aucune modification du contenu
- ❑ Nombreuses possibilités pour structurer les données
 - Important pour répondre aux besoins
- ❑ Une requête ressemble à une question
 - Qui sont les employés qui...
 - Quel employé a le montant maximum de...
 - Combien d'employés ont...
 - Quel est l'âge moyen des enfants des employés qui ont...
- ❑ Une requête sans résultat n'est pas fausse !
 - Il n'y a juste aucune donnée correspondant à la question posée

Bases d'une requête

- ❑ Mots-clés SQL : SELECT, FROM
- ❑ FROM permet de choisir une table
- ❑ SELECT permet de choisir les colonnes
 - Le symbole * veut dire « toutes les colonnes »
- ❑ Exemples
 - SELECT * FROM Employes
 - SELECT * FROM Enfants
 - SELECT Prenom,Nom FROM Employes
 - SELECT Prenom,Nom,Salaire FROM Employes
 - SELECT Nom FROM Metiers

Filtrage de lignes

- ❑ Mot-clé SQL : WHERE
- ❑ Toujours associé à un test booléen (Vrai/Faux)
- ❑ Conserve uniquement les lignes pour lesquelles le test est vrai
- ❑ Opérateurs de comparaison : < <= >= > = <>
- ❑ Exemples
 - SELECT * FROM Employes WHERE Salaire>50000
 - SELECT * FROM Enfants WHERE DateNaissance>='2000-01-01'
 - SELECT * FROM Employes WHERE Nom='Doe'
 - SELECT * FROM Employes WHERE Nom<>'Durand'
 - SELECT * FROM Metiers WHERE Nom='Manager'

Filtrage de lignes

❑ Mots-clés SQL : AND, OR, NOT

❑ Pour des tests plus complexes

- Eventuellement avec parenthèses

❑ Exemples

- `SELECT * FROM Employes WHERE Salaire >= 20000 AND Salaire <= 30000`
- `SELECT * FROM Employes WHERE Nom = 'Doe' OR Nom = 'Dup'`
- `SELECT * FROM Enfants WHERE NOT (DateNaissance >= '2000-01-01' AND DateNaissance <= '2000-12-31')`
- `SELECT * FROM Employes WHERE Salaire >= 20000 AND (Nom = 'Doe' OR Nom = 'Dup')`
- `SELECT * FROM Employes WHERE (Salaire >= 20000 AND Nom = 'Doe') OR Nom = 'Dup'`

Filtrage de lignes

❑ Mot-clé SQL : IN

❑ Tester si une valeur est comprise dans une liste

- Plus court que d'utiliser plusieurs OR

❑ Exemples

- `SELECT * FROM Employes
WHERE Salaire=10000 OR Salaire=20000 OR Salaire=30000`
- `SELECT * FROM Employes
WHERE Salaire IN (10000,20000,30000)`
- `SELECT * FROM Employes
WHERE Nom='Doe' OR Nom='Dup'`
- `SELECT * FROM Employes WHERE Nom IN ('Doe','Dup')`
- `SELECT * FROM Metiers WHERE Nom NOT IN
('Commercial','Ingénieur')`

Filtrage de lignes

❑ Mot-clé SQL : LIKE

❑ Test si un texte ressemble à un modèle

- Le symbole % représente n'importe quel(s) caractère(s)

❑ Exemples

- `SELECT * FROM Employes WHERE Nom LIKE 'D%'`
- `SELECT * FROM Enfants WHERE Nom LIKE '%o%'`
- `SELECT * FROM Employes WHERE Nom NOT LIKE '%o%'`
- `SELECT * FROM Employes WHERE Nom NOT LIKE 'D%d'`

Filtrage de lignes

❑ Mot-clé SQL : NULL

❑ NULL représente une absence de valeur

- `NULL ≠ ""` : un texte vide est une donnée (donc non NULL)

❑ Exemples

- `SELECT * FROM Employes
WHERE DateNaissance IS NULL`
- `SELECT * FROM Enfants
WHERE DateNaissance IS NOT NULL`

Agrégation de lignes

❑ Mot-clé SQL : SUM, AVG, MIN, MAX, COUNT

❑ Différentes fonctions d'agrégation

- La plupart du temps utilisées avec des données numériques même s'il est possible d'agréger des textes
 - GROUP_CONCAT

❑ Agrégation des données d'une colonne

❑ Exemples

- SELECT MIN(DateNaissance) FROM Enfants
- SELECT MAX(Salaire),MIN(DateNaissance) FROM Employes
- SELECT AVG(Salaire) FROM Employes WHERE idMetier=5
- SELECT SUM(Salaire) FROM Employes WHERE DateNaissance>='2000-01-01'

Agrégation de lignes

- ❑ Mot-clé SQL : GROUP BY
- ❑ Regroupes des lignes par rapport à une ou plusieurs colonnes ayant la même valeur
- ❑ Il est alors possible d'utiliser une fonction d'agrégation sur les colonnes non concernées par le regroupement
- ❑ Etudions quelques exemples...

Agrégation de lignes

Exemple 1

```
SELECT Nom  
FROM Employes
```

| Nom |
|---------|
| Durand |
| Doe |
| Dup |
| Durand |
| Dupont |
| Antoine |
| Masson |
| Dup |

```
SELECT Nom,COUNT(*)  
FROM Employes  
GROUP BY Nom
```

| Nom | COUNT(*) |
|---------|----------|
| Antoine | 1 |
| Doe | 1 |
| Dup | 2 |
| Dupont | 1 |
| Durand | 2 |
| Masson | 1 |

Agrégation de lignes

Exemple 2

```
SELECT Nom,Salaire  
FROM Employes
```

| Nom | Salaire |
|---------|---------|
| Durand | 30000 |
| Doe | 60000 |
| Dup | 55000 |
| Durand | 60000 |
| Dupont | 20000 |
| Antoine | 15000 |
| Masson | 80000 |
| Dup | 15000 |

```
SELECT  
Nom,AVG(Salaire)  
FROM Employes  
GROUP BY Nom
```

| Nom | AVG(Salaire) |
|---------|--------------|
| Antoine | 15000 |
| Doe | 60000 |
| Dup | 35000 |
| Dupont | 20000 |
| Durand | 45000 |
| Masson | 80000 |

Agrégation de lignes

Exemple 3

```
SELECT Nom,Salaire  
FROM Employes
```

| Nom | Salaire |
|---------|---------|
| Durand | 30000 |
| Doe | 60000 |
| Dup | 55000 |
| Durand | 60000 |
| Dupont | 20000 |
| Antoine | 15000 |
| Masson | 80000 |
| Dup | 15000 |

```
SELECT Salaire,COUNT(*)  
FROM Employes  
GROUP BY Salaire
```

| Salaire | COUNT(*) |
|---------|----------|
| 15000 | 2 |
| 20000 | 1 |
| 30000 | 1 |
| 55000 | 1 |
| 60000 | 2 |
| 80000 | 1 |

Filtrage de lignes agrégées

- ❑ Mot-clé SQL : HAVING
- ❑ Filtre les lignes après regroupement
- ❑ Même principe qu'avec WHERE
 - Avec la possibilité d'utiliser des fonctions d'agrégation appliquées aux groupements
- ❑ Reprenons les exemples précédents...

Filtrage de lignes agrégées

Exemple 1

```
SELECT Nom,COUNT(*)  
FROM Employes  
GROUP BY Nom
```

| Nom | COUNT(*) |
|---------|----------|
| Antoine | 1 |
| Doe | 1 |
| Dup | 2 |
| Dupont | 1 |
| Durand | 2 |
| Masson | 1 |

```
SELECT Nom,COUNT(*)  
FROM Employes  
GROUP BY Nom  
HAVING COUNT(*)=2
```

| Nom | COUNT(*) |
|--------|----------|
| Dup | 2 |
| Durand | 2 |

Filtrage de lignes agrégées

Exemple 2

```
SELECT  
Nom,AVG(Salaire)  
FROM Employes  
GROUP BY Nom
```

| Nom | AVG(Salaire) |
|---------|--------------|
| Antoine | 15000 |
| Doe | 60000 |
| Dup | 35000 |
| Dupont | 20000 |
| Durand | 45000 |
| Masson | 80000 |

```
SELECT  
Nom,AVG(Salaire)  
FROM Employes  
GROUP BY Nom  
HAVING  
AVG(Salaire)>=20000  
AND  
AVG(Salaire)<=40000
```

| Nom | AVG(Salaire) |
|--------|--------------|
| Dup | 35000 |
| Dupont | 20000 |

Filtrage de lignes agrégées

Exemple 3

```
SELECT Salaire,COUNT(*)  
FROM Employes  
GROUP BY Salaire
```

```
SELECT Salaire,COUNT(*)  
FROM Employes  
GROUP BY Salaire  
HAVING COUNT(*)=1
```

| Salaire | COUNT(*) |
|---------|----------|
| 15000 | 2 |
| 20000 | 1 |
| 30000 | 1 |
| 55000 | 1 |
| 60000 | 2 |
| 80000 | 1 |

| Salaire | COUNT(*) |
|---------|----------|
| 20000 | 1 |
| 30000 | 1 |
| 55000 | 1 |
| 80000 | 1 |

Trier les lignes

- ❑ Mot-clé SQL : ORDER BY, DESC
- ❑ Choix des colonnes sur lesquelles appliquer le tri
- ❑ La première colonne sera utilisée pour le tri principal
- ❑ La seconde colonne sera utilisée pour les lignes ayant la même valeur pour la première colonne...
- ❑ Exemples
 - `SELECT * FROM Employes ORDER BY Nom,Prenom`
 - `SELECT * FROM Employes WHERE idMetier=5 ORDER BY Nom`
 - `SELECT * FROM Metiers ORDER BY Nom DESC`
 - `SELECT * FROM Enfants
WHERE DateNaissance>'2000-01-01'
ORDER BY DateNaissance DESC`

Ordre des mots-clés

□ Un seul ordre possible

- 6 SELECT
- 1 FROM
- 2 WHERE
- 3 GROUP BY
- 4 HAVING
- 5 ORDER BY

□ Bien que placé en première position, le SELECT est en fait la dernière commande exécutée