



UNIVERSITÉ
DE REIMS
CHAMPAGNE-ARDENNE

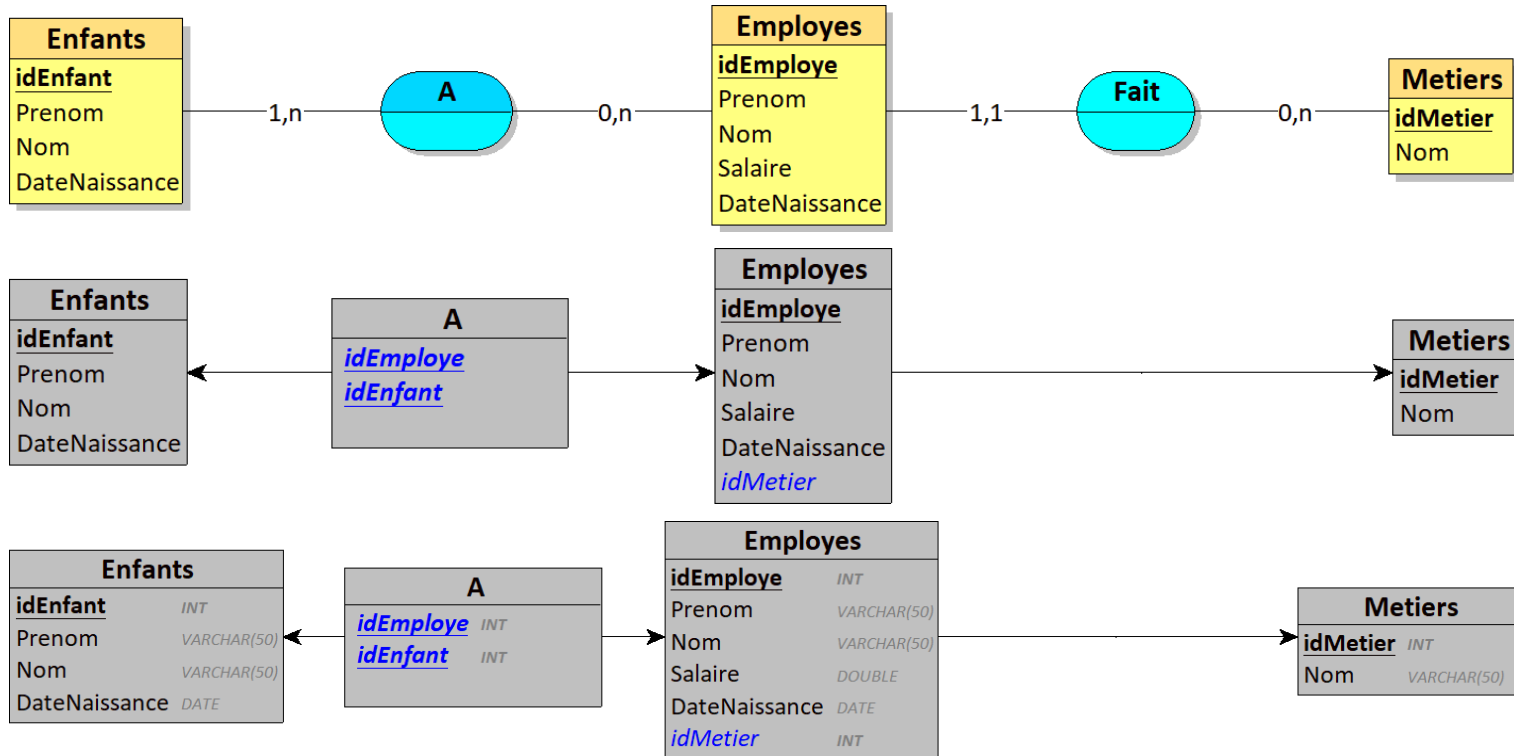
SQL – Partie 2

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

Requêtes basées sur plusieurs tables

- ❑ Une requête nécessite souvent plusieurs tables
- ❑ Si on veut afficher les informations des employés tout en affichant leur métier, il est nécessaire de travailler avec les 2 tables Employés et Métiers
- ❑ Nous allons commencer par réaliser un produit cartésien (CROSS JOIN), c'est-à-dire créer un ensemble de données composé de toutes les lignes d'une table mise en face de toutes les lignes de l'autre table
- ❑ Exemple
 - `SELECT * FROM Employes CROSS JOIN Metiers`

Produit cartésien

idEmploye	Prenom	Nom	Salaire	DateNaissance	idMetier	idMetier	Nom
1	Pierre	Durand	30000		4	1	Commercial
1	Pierre	Durand	30000		4	2	Manager
1	Pierre	Durand	30000		4	3	Comptable
1	Pierre	Durand	30000		4	4	Ingénieur
1	Pierre	Durand	30000		4	5	Communication
2	John	Doe	60000	1999-07-12	2	1	Commercial
2	John	Doe	60000	1999-07-12	2	2	Manager
2	John	Doe	60000	1999-07-12	2	3	Comptable
2	John	Doe	60000	1999-07-12	2	4	Ingénieur
2	John	Doe	60000	1999-07-12	2	5	Communication
3	Amélie	Dup	55000	1997-03-12	1	1	Commercial
3	Amélie	Dup	55000	1997-03-12	1	2	Manager
3	Amélie	Dup	55000	1997-03-12	1	3	Comptable
3	Amélie	Dup	55000	1997-03-12	1	4	Ingénieur
3	Amélie	Dup	55000	1997-03-12	1	5	Communication
4	Julie	Durand	60000	2000-04-05	4	1	Commercial
4	Julie	Durand	60000	2000-04-05	4	2	Manager
4	Julie	Durand	60000	2000-04-05	4	3	Comptable
4	Julie	Durand	60000	2000-04-05	4	4	Ingénieur
4	Julie	Durand	60000	2000-04-05	4	5	Communication
5	Caroline	Dupont	20000		5	1	Commercial
5	Caroline	Dupont	20000		5	2	Manager
5	Caroline	Dupont	20000		5	3	Comptable

Produit cartésien

idEmploye	Prenom	Nom	Salaire	DateNaissance	idMetier	idMetier	Nom
1	Pierre	Durand	30000		4	1	Commercial
1	Pierre	Durand	30000		4	2	Manager
1	Pierre	Durand	30000		4	3	Comptable
1	Pierre	Durand	30000		4	4	Ingénieur
1	Pierre	Durand	30000		4	5	Communication
2	John	Doe	60000	1999-07-12	2	1	Commercial
2	John	Doe	60000	1999-07-12	2	2	Manager
2	John	Doe	60000	1999-07-12	2	3	Comptable
2	John	Doe	60000	1999-07-12	2	4	Ingénieur
2	John	Doe	60000	1999-07-12	2	5	Communication
3	Amélie	Dup	55000	1997-03-12	1	1	Commercial
3	Amélie	Dup	55000	1997-03-12	1	2	Manager
3	Amélie	Dup	55000	1997-03-12	1	3	Comptable
3	Amélie	Dup	55000	1997-03-12	1	4	Ingénieur
3	Amélie	Dup	55000	1997-03-12	1	5	Communication
4	Julie	Durand	60000	2000-04-05	4	1	Commercial
4	Julie	Durand	60000	2000-04-05	4	2	Manager
4	Julie	Durand	60000	2000-04-05	4	3	Comptable
4	Julie	Durand	60000	2000-04-05	4	4	Ingénieur
4	Julie	Durand	60000	2000-04-05	4	5	Communication
5	Caroline	Dupont	20000		5	1	Commercial
5	Caroline	Dupont	20000		5	2	Manager
5	Caroline	Dupont	20000		5	3	Comptable

Mécanisme de jointure

- ❑ Mots-clés SQL : JOIN, ON, USING
- ❑ Permet de réaliser un produit cartésien en filtrant les lignes pour ne conserver que celles qui ont du sens
 - Critère de jointure (souvent relation clé primaire / clé étrangère)
- ❑ Exemple (2 possibilités)
 - `SELECT *`
`FROM Employes`
`JOIN Metiers ON Employes.idMetier=Metiers.idMetier`
 - `SELECT *`
`FROM Employes`
`JOIN Metiers USING(idMetier)`

Mécanisme de jointure

- ❑ Bien entendu, il est possible d'utiliser plusieurs fois le mot-clé JOIN si on veut réaliser une jointure avec plus que 2 tables
- ❑ Chaque JOIN fusionne une table avec l'ensemble précédent
- ❑ Exemple générique
 - FROM A *On commence avec la table A*
 - JOIN B *On fusionne la table B avec A pour produire AB*
 - JOIN C *On fusionne la table C avec AB pour produire ABC*
 - JOIN D *On fusionne la table D avec ABC pour produire ABCD*

Mécanisme de jointure

- ❑ Si on veut afficher les noms des employés et les noms de leurs enfants, il est nécessaire de travailler avec les 3 tables Employés, A et Enfants
- ❑ Exemples (2 possibilités)
 - ```
SELECT Enfants.Prenom, Enfants.Nom, Employes.Prenom,
 Employes.Nom
FROM Employes
JOIN A ON Employes.idEmploye=A.idEmploye
JOIN Enfants ON A.idEnfant=Enfants.idEnfant
```
  - ```
SELECT Enfants.Prenom, Enfants.Nom, Employes.Prenom,  
       Employes.Nom  
FROM Employes  
JOIN A USING(idEmploye)  
JOIN Enfants USING(idEnfant)
```


Requêtes basées sur plusieurs tables

□ Différentes jointures

- FROM A **JOIN** B : garde seulement les lignes fusionnées entre A et B qui respectent le critère de jointure
- FROM A **LEFT JOIN** B : garde les lignes fusionnées entre A et B qui respectent le critère de jointure et les lignes de la table A n'ayant pas d'association avec une ligne de la table B
- FROM A **RIGHT JOIN** B : garde les lignes fusionnées entre A et B qui respectent le critère de jointure et les lignes de la table B n'ayant pas d'association avec une ligne de la table A
- FROM A **FULL JOIN** B : garde les lignes fusionnées entre A et B qui respectent le critère de jointure, les lignes de la table A n'ayant pas d'association avec une ligne de la table B et les lignes de la table B n'ayant pas d'association avec une ligne de la table A

Ordre des mots-clés

□ Un seul ordre possible

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