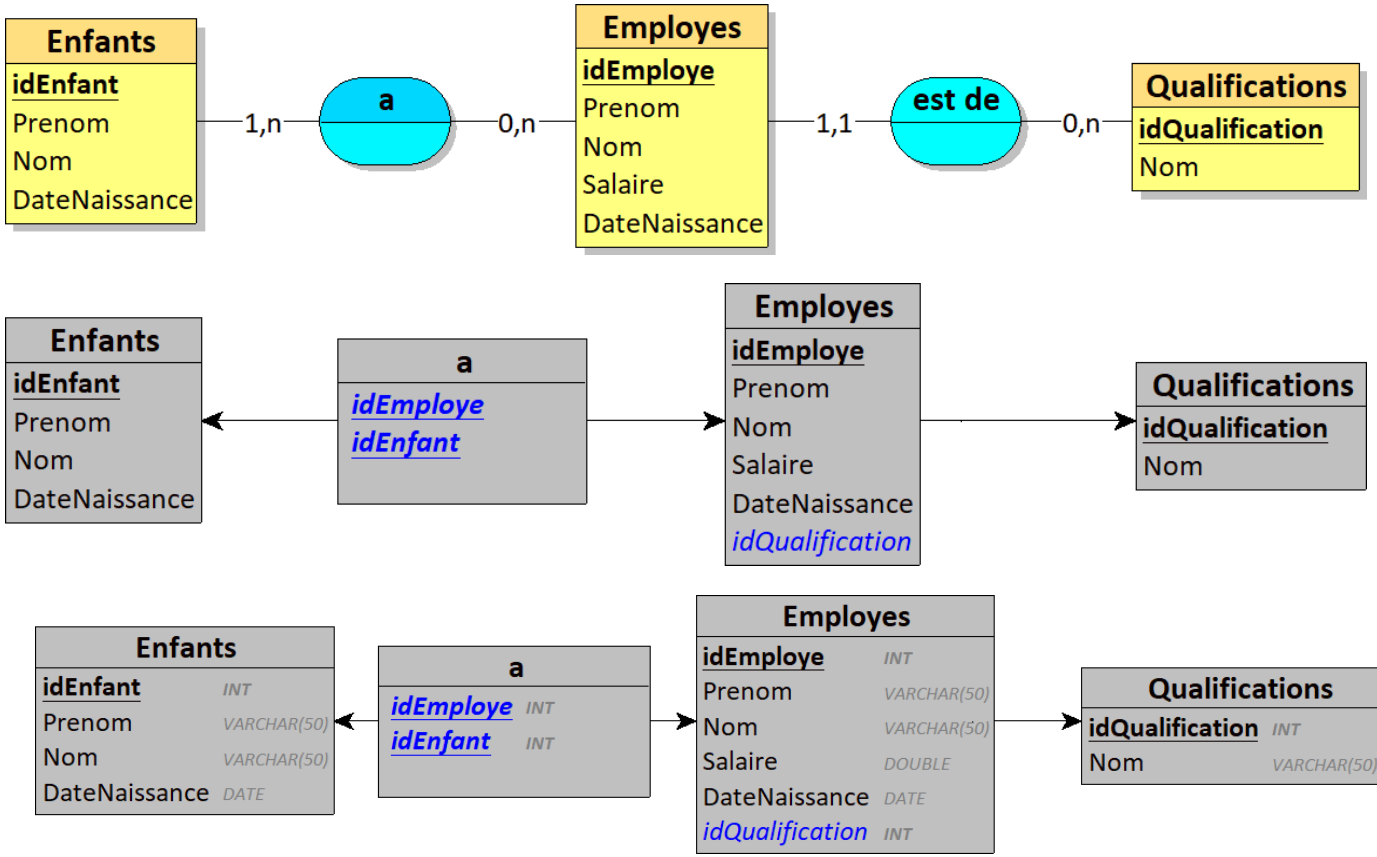
Requêtes SQL

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



Concept de requête

- Lire des données dans la base de données
  - Aucune modification du contenu
- Nombreuses possibilités pour structure 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

Bases d'un 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 Qualifications

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 Qualifications WHERE Nom='A1'

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 Qualifications WHERE Nom IN ('A1','A2','A3')

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 '%e%'
  - SELECT * FROM Employes WHERE Nom NOT LIKE '%e%'
  - SELECT * FROM Employes WHERE Nom NOT LIKE 'D%e'
  - SELECT * FROM Qualifications WHERE Nom LIKE 'A%'

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
- 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 idQualification=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
- Exemples
  - SELECT Nom,COUNT(*) FROM Employes GROUP BY Nom
  - SELECT Nom,AVG(Salaire) FROM Employes GROUP BY Nom
  - SELECT Salaire,COUNT(*) FROM Employes GROUP BY Salaire

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
- Exemples
  - SELECT Nom,COUNT(*) FROM Employes
    GROUP BY Nom HAVING COUNT(*)>5
  - SELECT Nom,AVG(Salaire) FROM Employes
    GROUP BY Nom HAVING AVG(Salaire)>=20000 AND
    AVG(Salaire)<=40000
  - SELECT Salaire,COUNT(*) FROM Employes
    GROUP BY Salaire HAVING COUNT(*)=10

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
- Le 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 idQualification=5 ORDER BY Nom
  - SELECT * FROM Qualifications ORDER BY Nom DESC
  - SELECT * FROM Enfants WHERE DateNaissance<'2000-01-01' ORDER BY DateNaissance DESC

Requêtes basées sur plusieurs tables

- Mot-clé SQL : JOIN
- Une requête nécessite souvent plusieurs tables
- Il faut donc « fusionner » les tables ensembles
- 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

Requêtes basées sur plusieurs tables

- Quand on fusionne une table avec un ensemble il faut expliquer le lien qui existe entre la table et l'ensemble (critère de jointure)
  - Souvent un lien clé primaire/clé étrangère
  - Mots-clés SQL : ON, USING
- Exemple avec 2 tables (versions ON et USING)
  - SELECT Prenom,Employes.Nom,Qualifications.Nom
    FROM Employes
    JOIN Qualifications ON
    Employes.idQualification=Qualifications.idQualification
  - SELECT Prenom,Employes.Nom,Qualifications.Nom
    FROM Employes
    JOIN Qualifications USING(idQualification)

Requêtes basées sur plusieurs tables

- Exemple avec 3 tables (verions ON et USING)
- 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