Liens rapides :

- Exemples de jointures

Classeurs

Toutes les requêtes doivent être placées dans un classeur pour être utilisables coté Utilisateur.

Vous pouvez ajouter dans un classeur autant de requêtes que nécessaire.

Vous pouvez également définir les droits utilisateur dans la partie Sécurités.

Attention Une requête interdite d'accès dans un classeur peut toujours être accessible dans un autre endroit comme un dossier ou un autre classeur

Dossiers

Ce module permet d'obtenir une vue de tous les documents correspondant à une ou plusieurs requêtes filtrées par un ou plusieurs critères .

Vous définissez les critères disponibles, le filtrage sera effectué par les utilisateurs dans leur interface

1. Vous pouvez ajouter autant de requêtes que nécessaires.
2. Au niveau du dossier, vous ajoutez un ou plusieurs champs communs aux requêtes, ce sont ces champs qui vont servir de critères de filtrage aux utilisateurs. Pour plus d'infos, voir la partie un peu plus bas Options des champs communs
3. Dans chaque requête ajoutée dans le dossier, vous associez le champ qui correspond à un des champs communs.

Suite à la configuration, l'utilisateur peut faire des recherches sur les différents champs que vous aurez définis et l'outil lui génèrera la vue correspondante avec tous les documents dans la vue document.

A partir de cette vue , l'utilisateur peut, en temps normal, générer un document PDF des documents ainsi trouvés.

Options des champs communs

- Description
- Type : pour permettre à l'utilisateur de saisir une valeur valide, le champ de formulaire pour ce champs commun sera affiché à l'utilisateur en fonction du type choisi ici
- Opérateur par défaut : sur l'interface utilisateur, l'opérateur par défaut s'affichera en premier opérateur sélectionné parmi ceux disponibles pour le critère
- Opérateur forcé : l'opérateur définis par défaut sera en lecture seule pour l'utilisateur
- champs requis : l'utilisateur doit remplir les champs requis au minimum pour générer la vue.

Requêtes

Dans la partie **Description**, deux parties :

- **Informations** ; formulaire de paramétrage général de la requête
- **Sécurité** ; pour paramétrer qui a quels droits sur la requête

Requêtes : Informations

- **Clé primaire** : permet à EzGED de repérer la colonne primaire. En principe, à la création de la requête, elle est paramétrée automatiquement. **Evitez de la modifier en cas d'incertitude des conséquences de la modification.**
- **Publier sur mobile** : permet à la requête d'être utilisée et affichée dans l'application mobile EzGED.
- **Distinct** : voir plus bas
- **Champ mail** : permet de générer une liste d'emails dans l'envoi de document par email coté Utilisateur
- **Publier dans le centre d'Indexation/Correction**: voir plus bas

Distinct

Ce paramètre s'applique sur les lignes affichées.

Il réduit les doublons à une seule ligne en ne gardant qu'une seule des lignes trouvées.

Si un enregistrement apparait plusieurs fois dans la vue, probablement à cause d'une jointure, soit ce paramètre soit appliquer une ou des conditions peut régler le problème.

A vous d'essayer

Une condition peut avoir un meilleur impact, voir les conditions.

Publier dans le centre d'Indexation/Correction

Les deux premiers modes créent une notification dans l'acceuil pour l'utilisateur qui y a le droit lorsqu'une nouvelle fiche est en attente d'une action.

Publier rend toutes les fiches libres (non en cours d'édition par un utilisateur) accessibles et éditables à qui y a le droit depuis le Centre d'Indexation / Correction.

Publier (Accès réservé) rend la première fiche libre éditable à qui y a le droit. Dans ce mode, seule la toute première fiche éditable est accessible à la fois. Tant qu'elle n'est pas complétée et validée, elle reste la première et seule accessible depuis le Centre d'Indexation / Correction.

Ne pas publier fait que les fiches de cette vue ne seront pas accessibles depuis le Centre d'Indexation / Correction.

Classement

Classeurs

C'est ici que vous pouvez dire dans quels classeurs, vous avez besoin que la requête soit rangée et accessible.

Arborescence

Ca permet de créer un triage par critère dans le classeur. Ce triage est créé sous forme d'une arborescence.

Vous pouvez ajouter autant de critères que vous voulez mais il est recommandé de limiter à un ou deux critères pour des questions de performance.

Exemple : si vous ajoutez un critère pour trier des factures par année puis par mois. Dans le classeur associé, vous verrez apparaitre des vues nommées avec les années et à l'intérieur, des vues nommées avec le mois.

Chaque vue affine l'affichage en ne vous montrant que ce que vous voulez voir donc la requête avec le filtre correspondant.

Sources

Ici, deux parties :

- Tables
- Jointures

Tables

C'est ici que vous définisssez sur quelles tables se base la recherche.

Il est conseillé de n'ajouter qu'**une seule table principale** et de rajouter les autres via des jointures, c'est plus propre.

Jointures

Les jointures permettent d'associer des champs issus d'autres tables que la table principale dans la recherche. Ces champs peuvent être affichés dans la vue comme servir un autre objectif (par exemple, un calcul simple).

Si vous avez besoin d'une jointure, demandez vous s'il y a plusieurs correspondances pour un enregistrement dans l'autre table :

Si OUI ; un enregistrement s'affichera plusieurs fois à l'utilisateur alors qu'en réalité, il n'existe qu'une fois. Dans ce cas, soit un DISTINCT soit une condition règlera le problème du doublon.

A la fin de cette partie, une liste de précautions à propos des jointures.

Trois types de jointures:

- Jointure gauche (LEFT JOIN en SQL)
- Jointure droite (RIGHT JOIN en SQL)
- Jointure classique (JOIN en SQL)

Le **rang** définit l'ordre de prise en compte des jointures.

Quelques points :

- Bien commencer par une première jointure sur la table source principale
- Bien jointer des types de colonnes compatibles entre eux
    - Exemples : jointer un ID avec un ID et non un texte
- **Attention** à ne pas jointer DEUX fois la même table

Pour vérifier le résultat, vous pouvez aller voir l'**Aperçu**.

Exemples ci-dessous sur deux tables, factures et bonsdelivraison, du résultat de chaque jointure :

1. Une jointure gauche LEFT JOIN en SQL
2. Une jointure droite RIGHT JOIN en SQL
3. Une jointure JOIN

Exemple de jointure gauche LEFT JOIN en SQL

Cas d'une jointure LEFT JOIN

1-3 / 3

FACTURES ID	FACTURES DATE	FACTURES CLIENT
1 *Facture 1*	01-03-18	1
2 *Facture 2*	03-03-18	1
3 *Facture 3*	15-03-18	1

Table factures

Jointure : factures.FACTURES_ID = bonsdelivraison.BONSDELIVRAISON_FACTURESID

Table bonsdelivraison

1-4 / 4

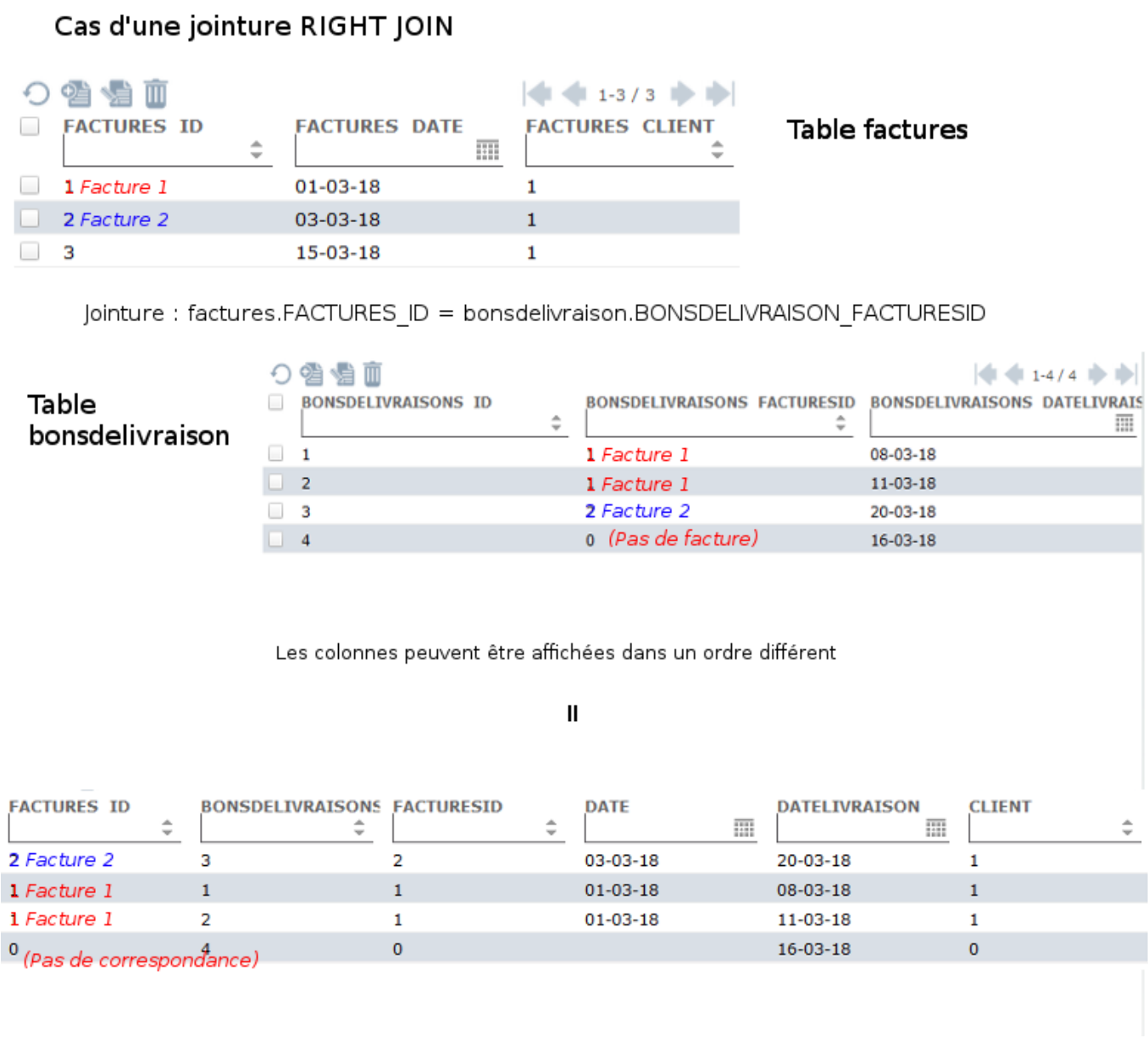
BONSDELIVRAISONS ID	BONSDELIVRAISONS FACTURESID	BONSDELIVRAISONS DATELIVRAIS
1	1 *Facture 1*	08-03-18
2	1 *Facture 1*	11-03-18
3	2 *Facture 2*	20-03-18
4 *Pas de correspondance*	0	16-03-18

Les colonnes peuvent être affichés dans un ordre différent mais dans une requête, vous pouvez les ré-ordonnez

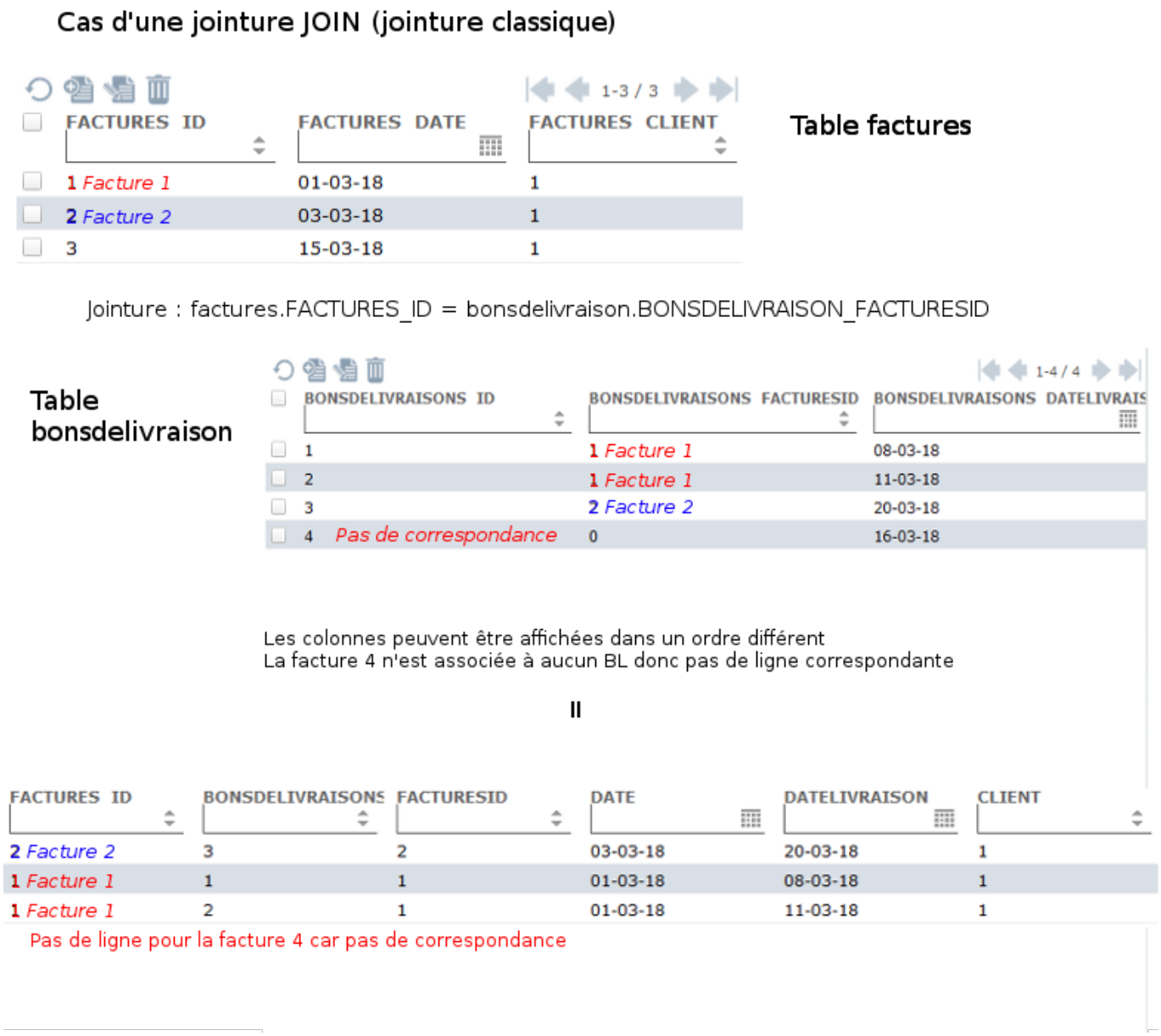
Résultat

FACTURES ID	BONSDELIVRAISONS	FACTURESID	DATE	DATELIVRAISON	CLIENT
3 *Facture 3*	0 (Pas de bordereau)	0	15-03-18		1
2 *Facture 2*	3	2	03-03-18	20-03-18	1
1 *Facture 1*	2	1	01-03-18	11-03-18	1
1 *Facture 1*	1	1	01-03-18	08-03-18	1

Exemple de jointure droite RIGHT JOIN en SQL



Exemple de jointure JOIN



Champs

Champs

Pour définir les champs affichés dans la vue et leur mode d'utilisation par l'utilisateur.

- **Champ de table** ou **Divers** : le champs peut être un **champ de la table** OU une expression SQL dans **Divers**.
  - Le champ **Divers** permet de créer un champ à partir d'une expression SQL spécifique
- L'**alias** sert à renommer un champ de la requête SQL

Si vous avez besoin d'avoir deux fois au moins un champs d'une même table :

1. indiquez le champs de la table
2. attribuez un alias à l'un des deux

Les autres params :

- **Afficher dans la grille** : si activé, le contenu des cases de cette colonne s'affichera, sinon il sera masqué mais la colonne reste visible

- Insertion/Modification
  - **Afficher** : mode normal, liste associée au champs accessible e éditable
  - **Masquer** : liste accessible mais pop up liste verrouillée
  - **Sans éditer la liste** : pop up liste en lecture seule
  - **Afficher en lecture seule** : champs en affichage uniquement

Remarque : si vous mettez un mode spécial sur un champ d'une table, mettez le sur tous les champs de la table pour qu'il soit pris en compte

- **Désactiver le mode liste** : si le champs est normalement associé à une liste, ce mode désactive la liste, on se retrouve donc avec un simple champs texte

RAD

Mode spécial d'édition qui permet d'apprendre au système à reconnaitre une valeur constante.

Valeurs par défaut

Sert à définir une valeur par défaut lorsqu'un utilisateur décide d'insérer un nouvel enregistrement

Filtres

Conditions

Attention Dans cette partie, pour sauvegarder **toute** modification, cliquer sur la disquette juste en haut à coté du bouton de rafraichissement

Aperçu

Ici, vous pouvez voir le résultat de la recherche telle que vous en aurez le droit dans l'interface utilisateurs.

Liens

Dossiers et requêtes

Cet outil permet d'associer d'autres vues ou dossiers à la vue actuelle en créant des liens web dans la vue.

Exemple : une requête B sera associée à une requête A en fonction des correspondances que vous

aurez définies. En fonction de ces correspondances, dans l'affichage utilisateur, un lien url est ajouté à chaque ligne de A qui possède une association existante. Cliquer sur une ligne A dirige vers une vue filtrée de B en fonction de la ligne cliquée dans A.

Un lien “Dossiers ” ouvrira le dossier lié tandis que le lien “Requêtes” ouvrira la requête correspondante filtrée.

Fusion

Cet outil permet d'associer des documents provenant d'une autre Recherche aux enregistrements de la vue actuelle. Les documents apparaissent dans la vue documents en-dessous de la grille de la Requête.

Avancés

GROUP BY

Triage par regroupement

- GROUP BY retourne un enregistrement par groupe
- GROUP BY peut être utilisé avec des fonctions comme COUNT, MAX, SUM, AVG, etc.
- Vous pouvez créer un ou plusieurs critères GROUP BY

HAVING

Filtre sur les groupes

- HAVING re-filtre les résultats filtrés par un GROUP BY ⇒ HAVING nécessite la présence d'un GROUP BY
- HAVING s'applique à des résultats synthétiques générés par la présence d'un GROUP BY alors que WHERE s'applique à des résultats individuels
- Seules les lignes dans les groupes satisfaisant le ou les critères GROUP BY seront affichées au final
- HAVING et WHERE peuvent être dans la même requête

From:
http://wiki.ezdev.fr/ - **EzGED Wiki**

Permanent link:
http://wiki.ezdev.fr/doku.php?id=doc:v3:presentation:archives&rev=1522759763

Last update: **2023/03/17 09:56**