

Documentation Formules Spreadsheet Evolution

Sylvain Boutet

17 mai 2025

Odoo Version 18.0

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 3 |
| 2 | Liste des formules disponibles | 3 |
| 2.1 | IROKOO.GET_FIELD | 3 |
| 2.1.1 | Syntaxe | 3 |
| 2.1.2 | Paramètres | 3 |
| 2.1.3 | Retourne | 4 |
| 2.1.4 | Exemple | 4 |
| 2.1.5 | Exemple avec champ relationnel | 4 |
| 2.2 | IROKOO.GET_IDS | 4 |
| 2.2.1 | Syntaxe | 4 |
| 2.2.2 | Paramètres | 4 |
| 2.2.3 | Formats de filtre supportés | 5 |
| 2.2.4 | Retourne | 5 |
| 2.2.5 | Exemple | 5 |
| 2.3 | IROKOO.GET_SUM | 5 |
| 2.3.1 | Syntaxe | 5 |
| 2.3.2 | Paramètres | 5 |
| 2.3.3 | Retourne | 6 |
| 2.3.4 | Exemple | 6 |
| 2.4 | IROKOO.GET_GROUPED_IDS | 6 |
| 2.4.1 | Syntaxe | 6 |
| 2.4.2 | Paramètres | 6 |
| 2.4.3 | Retourne | 6 |
| 2.4.4 | Exemple | 7 |
| 2.5 | IROKOO.SUM_BY_DOMAIN | 7 |
| 2.5.1 | Syntaxe | 7 |
| 2.5.2 | Paramètres | 7 |
| 2.5.3 | Retourne | 7 |
| 2.5.4 | Exemple | 7 |
| 3 | Utilisation avancée | 8 |
| 3.1 | Combinaison de formules | 8 |
| 3.2 | Filtres avec l'opérateur IN | 8 |
| 3.3 | Conseils pour les performances | 8 |
| 3.4 | Utilisation de cellules de référence comme filtres | 9 |
| 3.5 | Conseils pour l'optimisation | 9 |
| 4 | Dépannage | 9 |
| 4.1 | Erreurs courantes | 9 |
| 4.2 | Problèmes d'accès | 10 |
| 5 | Exemple de tableau de bord | 10 |
| 5.1 | Objectif du tableau de bord | 10 |
| 5.2 | Implémentation étape par étape | 10 |
| 5.2.1 | Étape 1 : Préparation des données de référence | 10 |
| 5.2.2 | Étape 2 : Évolution mensuelle des ventes | 11 |

| | | |
|-------|---|----|
| 5.2.3 | Étape 3 : Top 5 clients | 11 |
| 5.2.4 | Étape 4 : Top 5 produits | 11 |
| 5.2.5 | Étape 5 : Taux de conversion des devis | 12 |
| 5.2.6 | Étape 6 : Répartition par équipe de vente | 12 |
| 5.3 | Ajout de graphiques | 13 |

1 Introduction

Le module **Spreadsheet Evolution** étend les fonctionnalités des feuilles de calcul Odoo en ajoutant des formules avancées permettant d'exploiter pleinement les données du système directement dans vos spreadsheets sans export préalable.

Ces formules permettent :

- L'accès direct aux champs de n'importe quel enregistrement dans Odoo
- La recherche conditionnelle d'enregistrements avec des filtres avancés
- L'agrégation de données (sommations, moyennes, etc.)
- Les regroupements sophistiqués
- Le calcul direct de sommes avec des domaines de filtrage

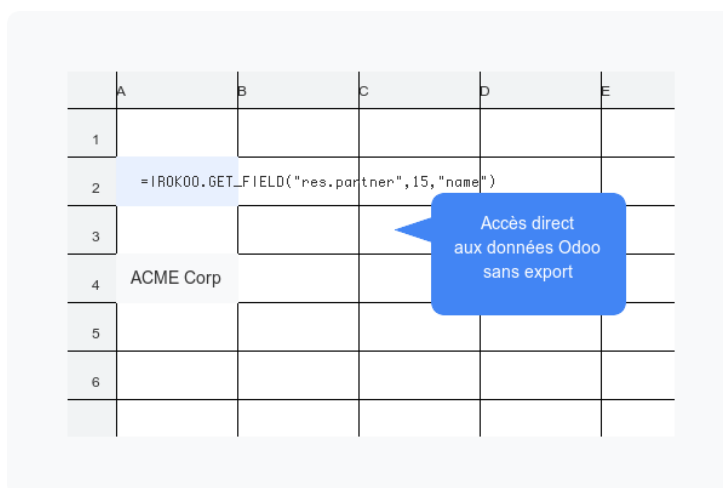


FIGURE 1 – Exemple d'utilisation des formules dans une feuille de calcul

2 Liste des formules disponibles

2.1 IROKOO.GET_FIELD

Description

Récupère la valeur d'un champ spécifique pour un enregistrement donné dans n'importe quel modèle Odoo.

2.1.1 Syntaxe

```
=IROKOO.GET_FIELD(model, id, field)
```

2.1.2 Paramètres

- **model** (chaîne) : Nom technique du modèle (ex : "res.partner")

- **id** (nombre) : ID de l'enregistrement
- **field** (chaîne) : Nom technique du champ (ex : "name", "email", "phone")

2.1.3 Retourne

La valeur du champ demandé.

2.1.4 Exemple

```
=IROKOO.GET_FIELD("res.partner", 1, "name")
```

Retourne le nom du partenaire ayant l'ID 1.

2.1.5 Exemple avec champ relationnel

```
=IROKOO.GET_FIELD("sale.order", 123, "partner_id.name")
```

Retourne directement le nom du client associé à la commande de vente ayant l'ID 123, en accédant au champ **name** à travers la relation **partner_id**.

Astuce

Cette formule fonctionne avec tous les types de champs, y compris les champs relationnels. Pour les champs relationnels, le contenu retourné sera automatiquement formaté de manière appropriée.

2.2 IROKOO.GET_IDS

Description

Récupère les IDs des enregistrements d'un modèle qui correspondent aux critères de filtre spécifiés.

2.2.1 Syntaxe

```
=IROKOO.GET_IDS(model, order, direction, limit, filters)
```

2.2.2 Paramètres

- **model** (chaîne) : Nom technique du modèle (ex : "sale.order")
- **order** (chaîne) : Champ pour le tri (ex : "date_order")
- **direction** (chaîne) : Direction du tri ("asc" ou "desc")
- **limit** (nombre) : Nombre maximum d'enregistrements à retourner (0 pour pas de limite)
- **filters** (chaîne) : Filtres séparés par des points-virgules

2.2.3 Formats de filtre supportés

- **Format explicite** : `field:operator:value`
 - Exemple : `state:in:draft,sent`
 - Opérateurs : `=`, `!=`, `>`, `<`, `>=`, `<=`, `in`, `not in`, `ilike`, `like`
- **Format simple** : `field=value`, `field>value`, etc.
 - Exemple : `amount_total>1000`
- **Format recherche texte** : `field value` pour recherches insensibles à la casse
 - Exemple : `name Comptabilité`
- **Format avec %** : `field=%valeur%` pour recherches partielles
 - Exemple : `name=%Dupont%`

2.2.4 Retourne

Une chaîne contenant les IDs séparés par des virgules.

2.2.5 Exemple

```
=IROKOO.GET_IDS("sale.order", "date_order", "desc", 10, "state:in:draft,sent;date_order>2023-01-01")
```

Retourne les 10 derniers devis/commandes en état brouillon ou envoyé créés après le 1er janvier 2023.

Attention !

Pour les filtres contenant des caractères spéciaux, assurez-vous d'utiliser le format explicite avec des deux-points comme séparateurs.

2.3 IROKOO.GET_SUM

Description

Calcule la somme des valeurs d'un champ pour une liste d'IDs.

2.3.1 Syntaxe

```
=IROKOO.GET_SUM(model, field, ids)
```

2.3.2 Paramètres

- **model** (chaîne) : Nom technique du modèle (ex : "sale.order")
- **field** (chaîne) : Nom technique du champ à sommer (ex : "amount_untaxed")
- **ids** (chaîne) : Liste d'IDs séparés par des virgules (généralement le résultat de GET_IDS)

2.3.3 Retourne

La somme des valeurs du champ pour les enregistrements spécifiés.

2.3.4 Exemple

```
=IROKOO.GET_SUM("sale.order", "amount_untaxed", IROKOO.GET_IDS("sale.order", "date_order", "desc", 0, "state=sale"))
```

Calcule la somme des montants HT de toutes les commandes confirmées.

Note

Cette formule est particulièrement utile combinée avec GET_IDS pour créer des rapports dynamiques.

2.4 IROKOO.GET_GROUPED_IDS

Description

Regroupe les enregistrements selon un champ et calcule une valeur agrégée pour chaque groupe.

2.4.1 Syntaxe

```
=IROKOO.GET_GROUPED_IDS(model, group_by, aggregate_field, aggregate_function, filters, limit)
```

2.4.2 Paramètres

- **model** (chaîne) : Nom technique du modèle
- **group_by** (chaîne) : Champ utilisé pour le regroupement
- **aggregate_field** (chaîne) : Champ à agréger
- **aggregate_function** (chaîne) : Fonction d'agrégation : "sum", "avg", "count", "min", "max"
- **filters** (chaîne) : Filtres (même format que GET_IDS)
- **limit** (nombre) : Nombre maximum de groupes à retourner

2.4.3 Retourne

Une chaîne contenant les valeurs du champ de regroupement, séparées par des virgules, pour les groupes classés par valeur agrégée décroissante.

2.4.4 Exemple

```
=IROKOO.GET_GROUPED_IDS("sale.order", "partner_id", "amount_total", "sum", "state=sale", 5)
```

Retourne les IDs des 5 clients ayant généré le plus de chiffre d'affaires à travers les commandes confirmées.

Astuce

Cette formule est idéale pour créer des analyses de type "Top N" ou pour identifier des tendances comme les clients les plus actifs, les produits les plus vendus, etc.

2.5 IROKOO.SUM_BY_DOMAIN

Description

Calcule directement la somme d'un champ pour les enregistrements correspondant à un domaine, sans nécessiter d'étape intermédiaire.

2.5.1 Syntaxe

```
=IROKOO.SUM_BY_DOMAIN(model, field, filters)
```

2.5.2 Paramètres

- **model** (chaîne) : Nom technique du modèle
- **field** (chaîne) : Champ à sommer
- **filters** (chaîne) : Filtres (même format que GET_IDS)

2.5.3 Retourne

La somme des valeurs du champ pour les enregistrements correspondant aux filtres.

2.5.4 Exemple

```
=IROKOO.SUM_BY_DOMAIN("account.move.line", "balance", "account_id=401100;date>=2023-01-01;date<=2023-12-31")
```

Calcule le solde total des écritures comptables sur le compte fournisseurs 401100 pour l'année 2023.

| | A | B | C | D | E |
|---|--|---|---|---|---|
| 1 | | | | | |
| 2 | =IROK00.SUM_BY_DOMAIN("sale.order", "amount_untaxed", "state:in:draft,sent | | | | |
| 3 | | | | | |
| 4 | 15,240.00 € | | | | |
| 5 | | | | | |
| 6 | | | | | |

Filtres avancés
Multiple conditions
Opérateur IN

FIGURE 2 – Exemple d'utilisation de SUM_BY_DOMAIN avec filtres complexes

3 Utilisation avancée

3.1 Combinaison de formules

Les formules peuvent être combinées pour créer des analyses complexes :

```
=IROK00.GET_SUM("sale.order", "amount_untaxed",
  IROK00.GET_IDS("sale.order", "date_order", "desc", 0,
    "state=sale;partner_id=" & IROK00.GET_GROUPED_IDS("sale.order", "
    partner_id", "amount_total", "sum", "state=sale", 1)
  )
)
```

Cette formule calcule le montant total des ventes pour le meilleur client.

3.2 Filtres avec l'opérateur IN

Utilisez l'opérateur IN pour filtrer sur plusieurs valeurs :

```
=IROK00.SUM_BY_DOMAIN("sale.order", "amount_total", "state:in:draft,sent
,sale;date>=2023-01-01")
```

3.3 Conseils pour les performances

- Limiter le nombre d'appels de formules en utilisant SUM_BY_DOMAIN directement quand possible
- Utiliser des limites appropriées dans GET_IDS pour réduire le volume de données
- Réutiliser les résultats de GET_IDS dans plusieurs cellules plutôt que de répéter l'appel

- Préférer le format explicite `field :operator :value` pour les filtres complexes

3.4 Utilisation de cellules de référence comme filtres

Il est important de noter que les formules `IROKOO.*` ne sont pas directement compatibles avec les filtres globaux natifs d'Odoo Spreadsheet. Cependant, vous pouvez créer un système de filtrage efficace en utilisant des cellules de référence :

- **Pour l'année** : Créez une cellule avec validation de données (liste déroulante) contenant les années disponibles, et référencez cette cellule dans vos formules
- **Pour la période** : Utilisez des cellules pour définir les dates de début et de fin, qui serviront de paramètres à toutes vos formules
- **Pour d'autres filtres** : Créez des cellules contenant les IDs des équipes, produits, etc., que vous souhaitez filtrer

Exemple de mise en place :

```
D1: "Filtre par équipe"
D2: [liste déroulante avec IDs des équipes]
```

Puis dans vos formules, référencez cette cellule :

```
=IROKOO.SUM_BY_DOMAIN("sale.order", "amount_untaxed", "state=sale;
team_id=" & D2 & ";date_order>=" & TEXT(B2,"yyyy-mm-dd"))
```

Cette approche permet de mettre à jour dynamiquement toutes vos formules lorsque l'utilisateur change la valeur d'une cellule de filtrage.

3.5 Conseils pour l'optimisation

- Utilisez des cellules de référence pour les paramètres communs (dates, filtres)
- Minimisez le nombre d'appels à `IROKOO.GET_IDS` en stockant les résultats dans des cellules intermédiaires
- Privilégiez `IROKOO.SUM_BY_DOMAIN` pour les calculs directs plutôt que des formules imbriquées complexes
- Pour les tableaux très volumineux, segmentez vos requêtes par période pour améliorer les performances

Ce tableau de bord vous donne un point de départ pour créer vos propres analyses adaptées à vos besoins spécifiques. Les formules peuvent être modifiées et combinées pour répondre à différentes questions métier.

4 Dépannage

4.1 Erreurs courantes

- **"All parameters are required"** : Vérifiez que tous les arguments obligatoires sont fournis
- **"Loading data..."** : Patientez, les données sont en cours de chargement

- **"No results found"** : Les filtres ne correspondent à aucun enregistrement
- **"Error while executing search"** : Vérifiez la syntaxe des filtres et l'existence du modèle

4.2 Problèmes d'accès

- Les formules respectent les droits d'accès de l'utilisateur
- Un utilisateur ne peut accéder qu'aux données auxquelles il a droit
- Si une formule retourne une erreur pour certains utilisateurs, vérifiez leurs droits d'accès

Attention !

Si vous rencontrez des problèmes d'accès, assurez-vous que l'utilisateur a les droits nécessaires pour accéder au modèle et aux champs spécifiés dans la formule.

5 Exemple de tableau de bord

Cette section présente un exemple complet de construction d'un tableau de bord d'analyse des ventes à l'aide des formules du module Spreadsheet Evolution.

5.1 Objectif du tableau de bord

Notre tableau de bord vise à répondre aux questions suivantes :

- Quelle est l'évolution mensuelle des ventes sur l'année en cours ?
- Qui sont nos 5 meilleurs clients et leur chiffre d'affaires ?
- Quels sont nos 5 produits les plus vendus (en valeur et en quantité) ?
- Quel est le taux de conversion de nos devis en commandes ?
- Comment se répartissent nos ventes par équipe de vente ?

5.2 Implémentation étape par étape

5.2.1 Étape 1 : Préparation des données de référence

Commençons par créer des cellules de référence pour les dates et filtres :

```
A1: "Année en cours"
B1: =YEAR(TODAY())
A2: "Date début"
B2: =DATE(B1,1,1)
A3: "Date fin"
B3: =DATE(B1,12,31)
```

5.2.2 Étape 2 : Évolution mensuelle des ventes

Pour créer l'évolution mensuelle, nous allons calculer le total pour chaque mois :

```
A5: "Mois"
B5: "Total ventes HT"
C5: "Nombre de commandes"

A6: "Janvier"
B6: =IROK00.SUM_BY_DOMAIN("sale.order", "amount_untaxed", "state=sale;
    date_order>=" & TEXT(DATE(B1,1,1),"yyyy-mm-dd") & ";date_order<=" &
    TEXT(DATE(B1,1,31),"yyyy-mm-dd"))
C6: =IROK00.SUM_BY_DOMAIN("sale.order", "1", "state=sale;date_order>=" &
    TEXT(DATE(B1,1,1),"yyyy-mm-dd") & ";date_order<=" & TEXT(DATE(B1
    ,1,31),"yyyy-mm-dd"))

A7: "Février"
B7: =IROK00.SUM_BY_DOMAIN("sale.order", "amount_untaxed", "state=sale;
    date_order>=" & TEXT(DATE(B1,2,1),"yyyy-mm-dd") & ";date_order<=" &
    TEXT(DATE(B1,2,28),"yyyy-mm-dd"))
C7: =IROK00.SUM_BY_DOMAIN("sale.order", "1", "state=sale;date_order>=" &
    TEXT(DATE(B1,2,1),"yyyy-mm-dd") & ";date_order<=" & TEXT(DATE(B1
    ,2,28),"yyyy-mm-dd"))

# Répéter pour tous les mois...
```

5.2.3 Étape 3 : Top 5 clients

Identifions nos meilleurs clients par chiffre d'affaires :

```
E5: "Top 5 clients"
F5: "CA Total HT"

# Obtenir les IDs des 5 meilleurs clients
E6: =IROK00.GET_FIELD("res.partner", SPLIT(IROK00.GET_GROUPED_IDS("sale.
    order", "partner_id", "amount_total", "sum", "state=sale;date_order
    >=" & TEXT(B2,"yyyy-mm-dd") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd")
    , 1), ","), "name")
F6: =IROK00.SUM_BY_DOMAIN("sale.order", "amount_untaxed", "state=sale;
    partner_id=" & SPLIT(IROK00.GET_GROUPED_IDS("sale.order", "partner_id
    ", "amount_total", "sum", "state=sale;date_order>=" & TEXT(B2,"yyyy-
    mm-dd") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd"), 1), ",") & ";
    date_order>=" & TEXT(B2,"yyyy-mm-dd") & ";date_order<=" & TEXT(B3,"
    yyyy-mm-dd"))

# Répéter pour les 4 autres clients (rangs 2 à 5)...
```

5.2.4 Étape 4 : Top 5 produits

Analysons les produits les plus performants :

```
H5: "Top 5 produits"
I5: "Quantité vendue"
J5: "CA Total HT"

# Obtenir les données du premier produit
```

```

H6: =IROK00.GET_FIELD("product.product", SPLIT(IROK00.GET_GROUPED_IDS("
    sale.order.line", "product_id", "price_subtotal", "sum", "order_id.
    state=sale;order_id.date_order>=" & TEXT(B2,"yyyy-mm-dd") & ";
    order_id.date_order<=" & TEXT(B3,"yyyy-mm-dd"), 1), ","), "name")
I6: =IROK00.SUM_BY_DOMAIN("sale.order.line", "product_uom_qty", "
    order_id.state=sale;product_id=" & SPLIT(IROK00.GET_GROUPED_IDS("sale
    .order.line", "product_id", "price_subtotal", "sum", "order_id.state=
    sale;order_id.date_order>=" & TEXT(B2,"yyyy-mm-dd") & ";order_id.
    date_order<=" & TEXT(B3,"yyyy-mm-dd"), 1), ",") & ";order_id.
    date_order>=" & TEXT(B2,"yyyy-mm-dd") & ";order_id.date_order<=" &
    TEXT(B3,"yyyy-mm-dd"))
J6: =IROK00.SUM_BY_DOMAIN("sale.order.line", "price_subtotal", "order_id
    .state=sale;product_id=" & SPLIT(IROK00.GET_GROUPED_IDS("sale.order.
    line", "product_id", "price_subtotal", "sum", "order_id.state=sale;
    order_id.date_order>=" & TEXT(B2,"yyyy-mm-dd") & ";order_id.
    date_order<=" & TEXT(B3,"yyyy-mm-dd"), 1), ",") & ";order_id.
    date_order>=" & TEXT(B2,"yyyy-mm-dd") & ";order_id.date_order<=" &
    TEXT(B3,"yyyy-mm-dd"))

# Répéter pour les 4 autres produits...

```

5.2.5 Étape 5 : Taux de conversion des devis

Calculons le taux de conversion des devis en commandes :

```

A18: "Taux de conversion des devis"
A19: "Nombre total de devis"
B19: =IROK00.SUM_BY_DOMAIN("sale.order", "1", "date_order>=" & TEXT(B2,"
    yyyy-mm-dd") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd"))
A20: "Nombre de commandes confirmées"
B20: =IROK00.SUM_BY_DOMAIN("sale.order", "1", "state=sale;date_order>="
    & TEXT(B2,"yyyy-mm-dd") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd"))
A21: "Taux de conversion"
B21: =B20/B19

```

5.2.6 Étape 6 : Répartition par équipe de vente

Analysons les performances par équipe de vente :

```

E18: "Répartition par équipe de vente"
F18: "CA Total HT"
G18: "Nombre de commandes"

# Obtenir les équipes de vente
E19: =IROK00.GET_FIELD("crm.team", SPLIT(IROK00.GET_GROUPED_IDS("sale.
    order", "team_id", "amount_total", "sum", "state=sale;date_order>=" &
    TEXT(B2,"yyyy-mm-dd") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd"), 1),
    ","), "name")
F19: =IROK00.SUM_BY_DOMAIN("sale.order", "amount_untaxed", "state=sale;
    team_id=" & SPLIT(IROK00.GET_GROUPED_IDS("sale.order", "team_id", "
    amount_total", "sum", "state=sale;date_order>=" & TEXT(B2,"yyyy-mm-dd
    ") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd"), 1), ",") & ";date_order
    >=" & TEXT(B2,"yyyy-mm-dd") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd")
    )
G19: =IROK00.SUM_BY_DOMAIN("sale.order", "1", "state=sale;team_id=" &
    SPLIT(IROK00.GET_GROUPED_IDS("sale.order", "team_id", "amount_total",
    "sum", "state=sale;date_order>=" & TEXT(B2,"yyyy-mm-dd") & ";

```

```
date_order<=" & TEXT(B3,"yyyy-mm-dd"), 1), ",") & ";date_order>=" &  
TEXT(B2,"yyyy-mm-dd") & ";date_order<=" & TEXT(B3,"yyyy-mm-dd")  
# Répéter pour les autres équipes...
```

5.3 Ajout de graphiques

Pour rendre le tableau de bord plus visuel, ajoutez des graphiques :

- Un graphique en colonnes pour l'évolution mensuelle des ventes
- Un graphique en secteurs pour la répartition des ventes par équipe
- Un graphique en barres horizontales pour les top 5 clients et produits

Ce tableau de bord vous donne un point de départ pour créer vos propres analyses adaptées à vos besoins spécifiques. Les formules peuvent être modifiées et combinées pour répondre à différentes questions métier.