

1

Maîtrisez les bases du langage de formule DAX et ses fonctions les plus utilisées.

2

Enrichissez vos modèles de données avec de nouvelles colonnes calculées et des mesures.

3

Utilisez des filtres dans vos formules DAX.

4

Prenez en main des outils pour gagner du temps et améliorer vos développements DAX.

Code

```

1 -- Conversion des salaires USD en EUR
2 salaire_max_EUROS = VAR taux_USD_EUROS = 0.91 RETURN salaries[max_salary] *
  :_USD_EUROS
3
4 -- Calcul de la popularité des entreprises
5 MESURE_job_count = DISTINCTCOUNT(job_industries[job_id])
6
7 -- Somme conditionnelle
8 SUMX(table, table[column] * condition)
9
10 -- Utilisation de CALCULATE pour un filtre spécifique
11 PERCENTAGE_REMOTE_ALLOWED = CALCULATE(AVERAGE(postings[remote_allowed]),
  :ings[work_type] = "FULL_TIME")

```

## Définitions

### DAX (Data Analysis Expressions)

Langage de formules pour la manipulation des données dans Power BI.

### Mesures

Calculs dynamiques évalués à chaque interaction utilisateur dans Power BI.

### Colonnes calculées

Colonnes ajoutées à une table et stockées en mémoire. Elles sont calculées au moment du chargement des données.

### Fonctions itératives

Fonctions telles que 'SUMX', qui effectuent un calcul ligne par ligne, puis réalisent l'agrégation de ces calculs sur toute la colonne à la fin.

### CALCULATE

Fonction permettant d'appliquer des filtres spécifiques à une expression DAX.

## Bonnes pratiques

- ✓ Clarifier les besoins de l'utilisateur final pour comprendre le code à développer et visualisations attendues.
- ✓ Planifier le pipeline de calculs et structurer les étapes et les dépendances avant de se lancer dans le développement DAX.
- ✓ Tester progressivement les formules DAX en validant chaque partie avant des calculs complexes.
- ✓ Utiliser des variables pour simplifier le code et réduire la répétition avec des 'VAR'.
- ✓ Commenter le code DAX pour faciliter la maintenance avec des annotations.
- ✓ Protéger les données sensibles en utilisant des filtres
- ✓ Partager rapidement des prototypes et collecter des retours utilisateur rapidement.
- ✓ Optimiser les performances DAX : Utiliser DAX Studio pour diagnostiquer et optimiser votre code DAX.

## Erreurs classiques

- ✗ Négliger de clarifier les besoins des utilisateurs finaux et foncer dans le développement.
- ✗ Sauter les tests itératifs dans le développement de formules complètes et se retrouver avec des bugs en cascade
- ✗ Rendre le code illisible en répétant les mêmes formules plusieurs fois.
- ✗ Omettre les commentaires dans son code
- ✗ Surcharger le modèle de données des colonnes et tables calculées inutiles
- ✗ Ignorer de surveiller la performance de vos mesures DAX au risque d'avoir un rapport Power BI lent à l'utilisation.
- ✗ Oublier de concevoir et mettre à jour correctement les relations entre les tables de votre modèle de données.
- ✗ Ne pas mettre en place les filtres nécessaires à la sécurité de vos données confidentielles