

# Test Power BI et DAX - Test 2

**Sujet:** Power BI, DAX et Python

**Niveau:** Intermédiaire

**Nombre de questions:** 20

---

## Questions et Réponses

**Q1.** Comment utiliser ALLEXCEPT pour conserver certains filtres?

**R1.**

```
// Ignore tous les filtres SAUF ceux sur Agence
% Contribution Agence =
DIVIDE(
    SUM(Ventes[Montant]),
    CALCULATE(
        SUM(Ventes[Montant]),
        ALLEXCEPT(Ventes, Agences[Agence])
    )
)
```

---

**Q2.** Quelle est la différence entre ALL, ALLSELECTED et ALLEXCEPT?

**R2.** | Fonction | Effet | |----|----| | **ALL(Table)** | Ignore TOUS les filtres | | **ALLSELECTED(Table)** | Ignore les filtres sauf slicers | | **ALLEXCEPT(Table, Col)** | Ignore tous SAUF colonnes spécifiées | | **REMOVEFILTERS(Col)** | Retire le filtre d'une colonne |

---

**Q3.** Comment créer une mesure de comparaison période précédente?

**R3.**

```
Montant PP =
CALCULATE(
    SUM(Ventes[Montant]),
    PREVIOUSMONTH(Calendrier[Date])
)

Var vs PP = [Montant Actuel] - [Montant PP]

Var % PP = DIVIDE([Var vs PP], [Montant PP], 0)
```

---

**Q4.** Comment implémenter DATESINPERIOD pour des périodes glissantes?

**R4.**

```
// 12 derniers mois glissants
Montant 12M =
CALCULATE(
    SUM(Ventes[Montant]),
    DATESINPERIOD(
        Calendrier[Date],

```

```

        LASTDATE(Calendrier[Date]),
        -12,
        MONTH
    )
)

// 90 derniers jours
Montant 90J =
CALCULATE(
    SUM(Ventes[Montant]),
    DATESINPERIOD(
        Calendrier[Date],
        LASTDATE(Calendrier[Date]),
        -90,
        DAY
    )
)

```

---

**Q5.** Comment créer une table de paramètres pour sélection dynamique?

**R5.**

```

// 1. Créer la table
Parametres = DATATABLE(
    "ID", INTEGER,
    "Mesure", STRING,
    {
        {1, "Montant"},
        {2, "Quantité"},
        {3, "Marge"}
    }
)

// 2. Mesure dynamique
Valeur Sélectionnée =
SWITCH(
    SELECTEDVALUE(Parametres[Mesure]),
    "Montant", SUM(Ventes[Montant]),
    "Quantité", SUM(Ventes[Quantité]),
    "Marge", SUM(Ventes[Marge]),
    BLANK()
)

```

---

**Q6.** Comment utiliser USERRELATIONSHIP pour activer une relation inactive?

**R6.**

```

// Relations: Ventes[DateVente] → Calendrier[Date] (active)
//           Ventes[DateLivraison] → Calendrier[Date] (inactive)

Montant par Date Livraison =
CALCULATE(
    SUM(Ventes[Montant]),

```

```

    USERRELATIONSHIP(Ventes[DateLivraison], Calendrier[Date])
)

```

---

**Q7.** Comment créer un visuel semi-additif (ex: soldes)?

**R7.**

```

// Solde fin de période (pas somme!)
Solde =
CALCULATE(
    SUM(Comptes[Solde]),
    LASTDATE(Calendrier[Date])
)

// Ou avec LASTNONBLANK
Solde v2 =
CALCULATE(
    SUM(Comptes[Solde]),
    LASTNONBLANK(Calendrier[Date], SUM(Comptes[Solde]))
)

```

---

**Q8.** Comment utiliser TOPN dans une mesure?

**R8.**

```

Top 5 Clients Montant =
CALCULATE(
    SUM(Ventes[Montant]),
    TOPN(
        5,
        ALL(Clients[ClientID]),
        [Total Ventes],
        DESC
    )
)

```

---

**Q9.** Comment créer une mesure de rang avec filtres dynamiques?

**R9.**

```

Rang =
VAR ValeurActuelle = [Total Ventes]
VAR TableComplete =
    SUMMARIZE(
        ALLSELECTED(Produits),
        Produits[Produit],
        "Ventes", [Total Ventes]
    )
RETURN
COUNTROWS(
    FILTER(
        TableComplete,
        [Ventes] > ValeurActuelle
    )
)

```

```
)  
) + 1
```

---

**Q10.** Comment gérer les performances avec SUMX vs SUM?

**R10.**

```
// Préférer SUM quand possible (plus rapide)  
Total = SUM(Table[Colonne])  
  
// SUMX nécessaire pour calculs row-by-row  
Total Calcul = SUMX(Table, Table[Prix] * Table[Quantité])  
  
// Optimisation SUMX avec KEEPFILTERS  
Total Optimisé =  
SUMX(  
    KEEPFILTERS(Table),  
    Table[Prix] * Table[Quantité]  
)
```

---

**Q11.** Comment utiliser VALUES vs DISTINCT?

**R11.** | VALUES | DISTINCT | |———|———| | Inclut BLANK si présent | Exclut BLANK | | Résulte des filtres actifs | Résulte des filtres actifs |

```
Nb Clients = COUNTROWS(VALUES(Ventes[ClientID]))  
Nb Clients Sans Blank = COUNTROWS(DISTINCT(Ventes[ClientID]))
```

---

**Q12.** Comment créer un format conditionnel dynamique?

**R12.**

```
// Couleur basée sur performance  
Couleur Performance =  
SWITCH(  
    TRUE(),  
    [Var % YoY] >= 0.1, "#00FF00",    // Vert  
    [Var % YoY] >= 0, "#FFFF00",      // Jaune  
    [Var % YoY] >= -0.1, "#FFA500",   // Orange  
    "#FF0000"                        // Rouge  
)  
  
// Utiliser dans Format conditionnel > Règles > Par valeur de champ
```

---

**Q13.** Comment implémenter une mesure de moyenne pondérée?

**R13.**

```
Taux Moyen Pondéré =  
DIVIDE(  
    SUMX(Prets, Prets[Taux] * Prets[Montant]),  
    SUM(Prets[Montant])  
)
```

---

**Q14.** Comment utiliser GENERATE pour créer des combinaisons?

**R14.**

```
Matrice Combinaisons =  
GENERATE(  
    VALUES(Produits[Categorie]),  
    VALUES(Regions[Region])  
)  
// Crée toutes les combinaisons Catégorie × Région
```

---

**Q15.** Comment créer une mesure DAX pour le taux de rétention?

**R15.**

```
Taux Rétention =  
VAR ClientsDebut =  
    CALCULATE(  
        DISTINCTCOUNT(Transactions[ClientID]),  
        PREVIOUSMONTH(Calendrier[Date])  
    )  
VAR ClientsConserve =  
    CALCULATE(  
        DISTINCTCOUNT(Transactions[ClientID]),  
        FILTER(  
            VALUES(Transactions[ClientID]),  
            CALCULATE(  
                COUNTROWS(Transactions),  
                PREVIOUSMONTH(Calendrier[Date])  
            ) > 0  
        )  
    )  
RETURN  
DIVIDE(ClientsConserve, ClientsDebut)
```

---

**Q16-20.** [Questions sur l'intégration Python avancée, les custom visuals, et les bonnes pratiques de modélisation...]

---

## Scoring

Score	Niveau
0-8	À améliorer
9-13	Intermédiaire
14-17	Avancé
18-20	Expert