Cette infrastructure de **6 modèles spécialisés** permet une valorisation précise de tous types de collatéral bancaire!

PARAMÈTRES ET COEFFICIENTS PAR TYPE DE COLLATÉRAL

REAL ESTATE - PARAMÈTRES DÉTAILLÉS

Paramètres Cap Rate par Type/Localisation

```
vba
' === PARAMÈTRES CAP RATE (Feuille "Database") ===
Type CapRateParameters As Object
 .OFFICE\_CBD\_LONDON = 0.04 ' 4.0\%
 .OFFICE_CBD_FRANKFURT = 0.045 '4.5%
 .OFFICE\_SUBURBAN\_MAJOR = 0.055 ' 5.5%
 .OFFICE_SUBURBAN_SECONDARY = 0.07 '7.0%
 .RETAIL_HIGH_STREET_PRIME = 0.035 ' 3.5%
 .RETAIL_HIGH_STREET_GOOD = 0.045 ' 4.5%
 .RETAIL_SHOPPING_CENTER = 0.065 '6.5%
 .RESIDENTIAL\_PRIME = 0.025
                           ' 2.5%
 .RESIDENTIAL_GOOD = 0.035
                            ' 3.5%
 .RESIDENTIAL_AVERAGE = 0.045 ' 4.5\%
 .RESIDENTIAL_SOCIAL = 0.055
                            5.5%
 .INDUSTRIAL_LOGISTICS = 0.055 '5.5%
 .INDUSTRIAL MANUFACTURING = 0.08 '8.0%
 .INDUSTRIAL_SPECIALIZED = 0.095 '9.5%
 .HOSPITALITY_5_STAR = 0.06
                            6.0%
 .HOSPITALITY_4_STAR = 0.075
                            7.5%
 .HOSPITALITY_3_STAR = 0.09
                           ' 9.0%
End Type
```

Décotes de Liquidation par Type

```
' === DÉCOTES LIQUIDATION (Feuille "Collateral") ===
Type LiquidationDiscounts As Object
 .OFFICE_PRIME = 0.15 '15% décote
 .OFFICE_SECONDARY = 0.25 '25% décote
 .OFFICE_TERTIARY = 0.35 '35% décote
 .RETAIL_PRIME = 0.2 '20% décote
                           ' 35% décote
 .RETAIL\_SECONDARY = 0.35
 .RETAIL_STRUGGLING = 0.5 '50% décote
 .RESIDENTIAL_PRIME = 0.1 '10% décote
 .RESIDENTIAL_AVERAGE = 0.15 '15% décote
 .RESIDENTIAL_SOCIAL = 0.25 '25% décote
 .INDUSTRIAL_MODERN = 0.25 '25% décote
 .INDUSTRIAL\_STANDARD = 0.3
                             ' 30% décote
 .INDUSTRIAL_OBSOLETE = 0.45
                              ' 45% décote
 .LAND_DEVELOPMENT = 0.4 '40% décote
 .LAND_AGRICULTURAL = 0.2 '20% décote
End Type
```

Ajustements Conditions de Marché

Coûts de Réalisation

Formules Real Estate et Impacts

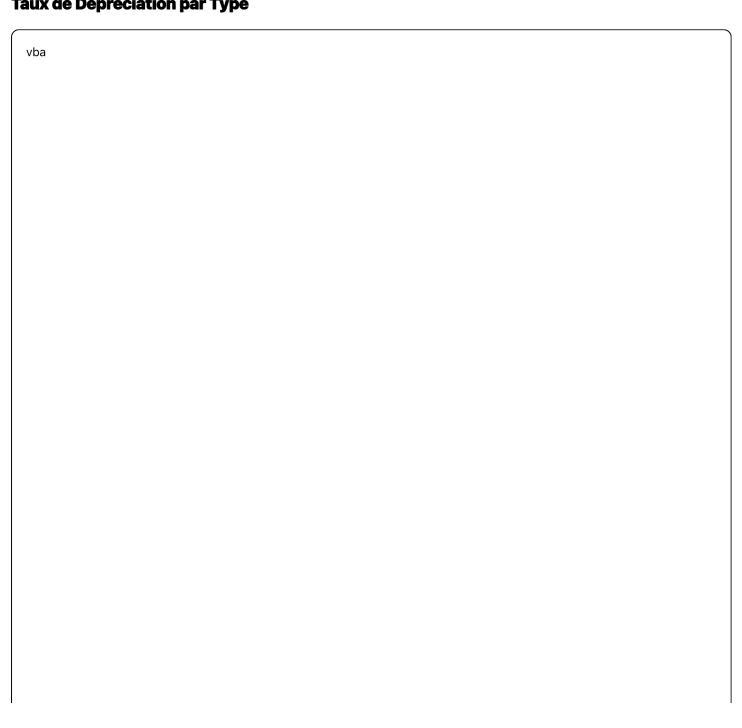
vba		

```
Function CalculateRealEstateLGDImpact(loanAmount As Double, _
                  propertyValue As Double, _
                  propertyType As String, _
                  marketCondition As String) As LGDResult
  Dim result As LGDResult
  ' === CALCUL VALEUR LIQUIDATION ===
  Dim baseDiscount As Double
  Dim marketMultiplier As Double
  Dim realizationCosts As Double
  baseDiscount = GetLiquidationDiscount(propertyType) 'Ex: 0.25 pour office secondary
  marketMultiplier = GetMarketMultiplier(marketCondition) 'Ex: 1.3 pour weak market
  realizationCosts = 0.08
                                        '8% frais
  ' Décote finale = Base × Marché
  Dim finalDiscount As Double
  ' Valeur de liquidation
  Dim liquidationValue As Double
  liquidationValue = propertyValue * (1 - finalDiscount) '10M \times (1-0.325) = 6.75M
  ' Valeur nette (après frais)
  Dim netRecoveryValue As Double
  netRecoveryValue = liquidationValue * (1 - realizationCosts) ' 6.75M × 0.92 = 6.21M
  ' === CALCUL LGD FINAL ===
  Dim Igd As Double
  If netRecoveryValue >= loanAmount Then
    lgd = 0
                                  ' Pas de perte
  Else
    Igd = (IoanAmount - netRecoveryValue) / IoanAmount ' (8M - 6.21M) / 8M = 22.4%
  End If
  ' Application minimums réglementaires
  If Igd < 0.1 Then Igd = 0.1 '10% minimum Bâle III
  If lgd > 0.95 Then lgd = 0.95 ' 95% maximum
  result.FinalLGD = lgd
  result.LiquidationValue = liquidationValue
  result.NetRecoveryValue = netRecoveryValue
  result.EffectiveDiscount = 1 - (netRecoveryValue / propertyValue)
  ' === IMPACTS AVAL ===
```

'Impact RWA (si garantie éligible) result.RWAImpact = CalculateRWAReduction(Igd)	
'Impact IFRS 9 ECL result.ECLImpact = CalculateECLReduction(lgd)	
'Impact pricing result.PricingImpact = CalculatePricingAdjustment(Igd)	
CalculateRealEstateLGDImpact = result	
End Function	

A VÉHICULES/ÉQUIPEMENTS - PARAMÈTRES DÉTAILLÉS

Taux de Dépréciation par Type



```
' === TAUX DÉPRÉCIATION ANNÉE 1 (Feuille "Database") ===
Type DepreciationRatesYear1 As Object
 .LUXURY_CAR = 0.25
                         ' 25% première année
                          ' 22% première année
 .PREMIUM_CAR = 0.22
 .STANDARD_CAR = 0.2
                          ' 20% première année
                          ' 18% première année
 .ECONOMY_CAR = 0.18
 .ELECTRIC_VEHICLE = 0.3 '30% (obsolescence tech)
 .COMMERCIAL_VAN = 0.15 '15% première année
 .COMMERCIAL_TRUCK = 0.12
                            ' 12% première année
 .HEAVY_TRUCK = 0.1 '10% première année
 .SPECIALIZED_VEHICLE = 0.2 '20% première année
 .CONSTRUCTION_EQUIPMENT = 0.08 '8% première année
 .MANUFACTURING_EQUIPMENT = 0.12 '12% première année
 .IT_EQUIPMENT = 0.4 '40% (obsolescence rapide)
 .MEDICAL_EQUIPMENT = 0.15 '15% première année
 .AIRCRAFT = 0.05 '5% première année
End Type
' === TAUX DÉPRÉCIATION ANNÉES SUIVANTES ===
Type DepreciationRatesSubsequent As Object
 .LUXURY_CAR = 0.15 '15% par an après année 1
 .STANDARD\_CAR = 0.12 ' 12% par an
 .COMMERCIAL_VEHICLE = 0.08 '8% par an
 .CONSTRUCTION_EQUIPMENT = 0.06 '6% par an
 .IT_EQUIPMENT = 0.25 ' 25% par an (tech)
End Type
```

Ajustements Kilométrage/Usage

```
vba

' === FACTEURS KILOMÉTRAGE (Feuille "Collateral") ===

Type MileageAdjustments As Object

.STANDARD_ANNUAL_KM = 15000   '15,000 km/an standard

.LOW_MILEAGE_BONUS = 0.05    '+5% si -50% kilomètres

.HIGH_MILEAGE_PENALTY_PER_1000KM = 0.005   '-0.5% par 1000km excès

.MAXIMUM_MILEAGE_PENALTY = 0.3    'Maximum -30%

' Pour équipements (heures d'usage)

.STANDARD_ANNUAL_HOURS = 2000    '2,000h/an standard

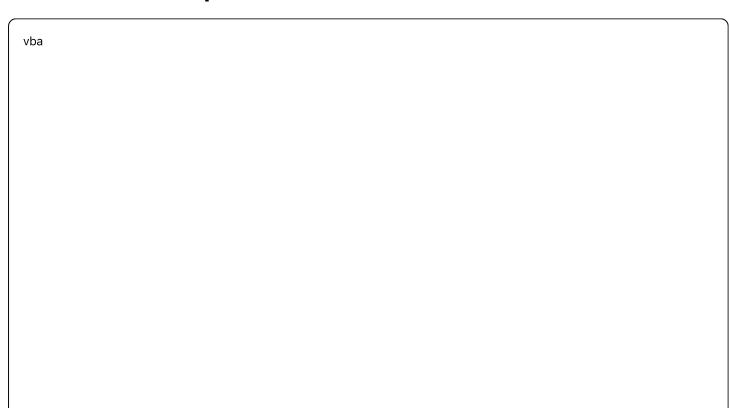
.INTENSIVE_USE_PENALTY = 0.003    '-0.3% par 100h excès

End Type
```

Ajustements Condition

Valeurs Résiduelles Minimales

Formules Véhicules et Impacts



```
Function CalculateVehicleLGDImpact(loanAmount As Double, _
                 vehicleCurrentValue As Double, _
                 vehicleType As String) As LGDResult
  Dim result As LGDResult
  ' === DÉCOTES LIQUIDATION VÉHICULES ===
  Dim liquidationDiscount As Double
  Select Case UCase(vehicleType)
    Case "LUXURY_CAR"
      liquidationDiscount = 0.3 '30% décote (marché étroit)
    Case "STANDARD_CAR"
      liquidationDiscount = 0.2 '20% décote
    Case "COMMERCIAL_VEHICLE"
      liquidationDiscount = 0.25 '25% décote
    Case "CONSTRUCTION_EQUIPMENT"
      liquidationDiscount = 0.35 '35% décote (spécialisé)
    Case "IT_EQUIPMENT"
      liquidationDiscount = 0.5 '50% décote (obsolescence)
  End Select
  'Coûts de réalisation
  Dim realizationCosts As Double
  realizationCosts = 0.1 '10% (transport, vente, paperasse)
  ' Valeur nette récupérable
  Dim netRecoveryValue As Double
  netRecoveryValue = vehicleCurrentValue * (1 - liquidationDiscount) * (1 - realizationCosts)
  ' LGD final
  Dim Igd As Double
  lgd = Application.WorksheetFunction.Max(0, (loanAmount - netRecoveryValue) / loanAmount)
  If lgd < 0.15 Then lgd = 0.15 '15% minimum véhicules
  result.FinalLGD = lgd
  result.NetRecoveryValue = netRecoveryValue
  CalculateVehicleLGDImpact = result
End Function
```

PARAMÈTRES DÉTAILLÉS

```
vba
' === DÉCOTES LIQUIDITÉ (Feuille "Database") ===
Type CommodityLiquidityDiscounts As Object
 ' Métaux précieux (haute liquidité)
 .GOLD_LBMA_GOOD = 0.02 '2% décote
 .SILVER_999 = 0.05 '5% décote
                       ' 8% décote
 .PLATINUM = 0.08
 .PALLADIUM = 0.1 '10% décote
 ' Énergie (liquidité variable)
 .CRUDE_OIL_BRENT = 0.05
                          ' 5% décote
 .CRUDE_OIL_WTI = 0.05 '5% décote
 .NATURAL_GAS = 0.15 '15% décote
 .COAL = 0.2 '20% décote
 ' Métaux industriels
 .COPPER_LME_GRADE_A = 0.08 '8% décote
 .ALUMINUM_PRIMARY = 0.1 '10% décote
               ' 12% décote
' 15% décote
 .ZINC = 0.12
 .NICKEL = 0.15
 ' Agricole (périssable)
 .WHEAT_FEED_GRADE = 0.15 '15% décote
 .CORN = 0.18 ' 18% décote
 .SOYBEANS = 0.2 '20% décote
 .COFFEE_ARABICA = 0.25 '25% décote
 ' Produits manufacturés
 .ELECTRONICS = 0.4 '40% décote (obsolescence)
 .TEXTILES = 0.35 '35% décote (mode)
 .AUTOMOTIVE_PARTS = 0.3 '30% décote
 .PHARMACEUTICALS = 0.5 '50% décote (réglementation)
End Type
```

Ajustements Qualité par Grade

```
' === MULTIPLICATEURS QUALITÉ (Feuille "Collateral") ===
Type QualityGradeMultipliers As Object
 ' Or
  .GOLD_999_FINE = 1.0
                            ' 100% (référence)
                             ' 91.7% (22/24)
  .GOLD_22_CARAT = 0.917
  .GOLD_18_CARAT = 0.75
                             ' 75% (18/24)
  .GOLD_14_CARAT = 0.583
                             ' 58.3% (14/24)
  ' Pétrole
  .CRUDE_BRENT_DATED = 1.0 ' 100% (référence)
                          ' 102% (premium léger)
  .CRUDE_WTI = 1.02
  .CRUDE_HEAVY = 0.9
                            ' 90% (décote lourd)
  .CRUDE_SOUR = 0.85
                            ' 85% (décote soufre)
 ' Agricole
 .WHEAT_GRADE_1 = 1.0
                             ' 100% (premium)
 .WHEAT_GRADE_2 = 0.95
                              ' 95% (standard)
 .WHEAT_GRADE_3 = 0.85
                              ' 85% (feed)
 .WHEAT_REJECTED = 0.7
                              '70% (rejected)
End Type
```

Ajustements Localisation

```
vba

' === AJUSTEMENTS LOCALISATION (Feuille "Database") ===

Type LocationAdjustments As Object

.MAJOR_EXCHANGE_DELIVERY = 1.0 '100% (LME, CBOT, etc.)

.MAJOR_PORT = 0.98 '98% (Rotterdam, Houston)

.INLAND_TERMINAL = 0.95 '95% (transport +2-3%)

.CERTIFIED_WAREHOUSE = 0.97 '97% (certifié)

.PRIVATE_STORAGE = 0.9 '90% (non certifié)

.REMOTE_LOCATION = 0.85 '85% (transport coûteux)

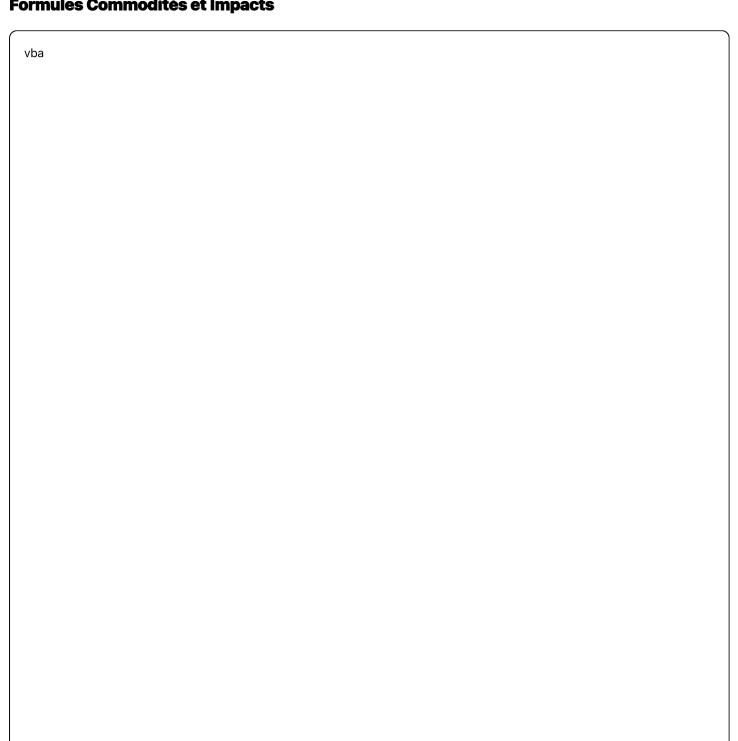
.HIGH_RISK_COUNTRY = 0.8 '80% (risque politique)

End Type
```

Facteurs Péremption/Détérioration

```
' === FACTEURS EXPIRATION (Feuille "Collateral") ===
Type ExpirationAdjustments As Object
  ' Produits périssables (décote par jours restants)
 .FRESH_PRODUCE_PER_DAY = 0.02 '-2% par jour
 .DAIRY_PRODUCTS_PER_DAY = 0.03 '-3% par jour
  .PHARMACEUTICALS_PER_MONTH = 0.05 '-5% par mois avant expiry
  ' Seuils critiques
 .PERISHABLE_30_DAYS = 0.7 '-30% si <30 jours
 .PERISHABLE_7_DAYS = 0.5 '-50% si <7 jours
 .PERISHABLE_1_DAY = 0.2 '-80% si <1 jour
End Type
```

Formules Commodités et Impacts

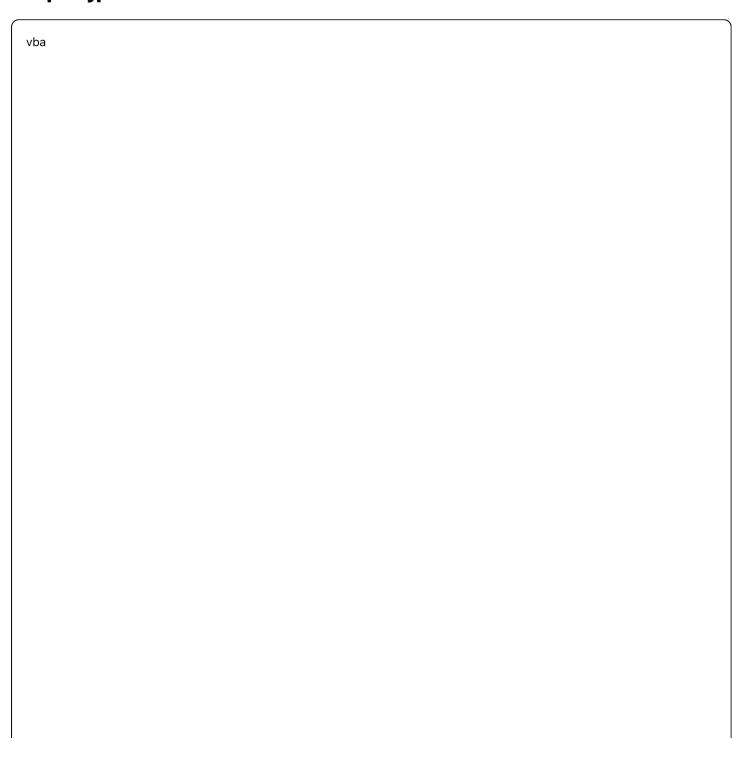


```
Function CalculateCommodityLGDImpact(loanAmount As Double, _
                  commodityValue As Double, _
                  commodityType As String) As LGDResult
  Dim result As LGDResult
  ' === DÉCOTES SPÉCIFIQUES COMMODITÉS ===
  Dim volatilityDiscount As Double
  Dim liquidityDiscount As Double
  Dim storageRisk As Double
  Select Case UCase(commodityType)
    Case "GOLD", "SILVER"
      volatilityDiscount = 0.05 '5% volatilité
      liquidityDiscount = 0.02 '2% liquidité excellente
      storageRisk = 0.01 '1% risque stockage
    Case "CRUDE_OIL"
      volatilityDiscount = 0.1 '10% volatilité
      liquidityDiscount = 0.05 '5% liquidité bonne
      storageRisk = 0.02 '2% risque stockage
    Case "AGRICULTURAL"
      volatilityDiscount = 0.15 '15% volatilité
      liquidityDiscount = 0.15 '15% liquidité moyenne
      storageRisk = 0.05 '5% risque détérioration
    Case "ELECTRONICS"
      volatilityDiscount = 0.2 '20% obsolescence
      liquidityDiscount = 0.3 '30% marché étroit
      storageRisk = 0.02 '2% risque stockage
  End Select
  ' Décote totale
  Dim totalDiscount As Double
  totalDiscount = 1 - ((1 - volatilityDiscount) * (1 - liquidityDiscount) * (1 - storageRisk))
  ' Coûts de réalisation
  Dim realizationCosts As Double
  realizationCosts = 0.05 '5% frais vente commodités
  ' Valeur nette
  Dim netRecoveryValue As Double
  netRecoveryValue = commodityValue * (1 - totalDiscount) * (1 - realizationCosts)
  ' LGD final
```

Dim Igd As Double Igd = Application.WorksheetFunction.Max(0, (IoanAmount - netRecoveryValue) / IoanAmount) result.FinalLGD = Igd result.TotalDiscount = totalDiscount result.NetRecoveryValue = netRecoveryValue CalculateCommodityLGDImpact = result End Function

GARANTIES PERSONNELLES - PARAMÈTRES DÉTAILLÉS

LGD par Type de Garant



```
' === LGD GARANTS (Feuille "Database") ===
Type GuarantorLGD As Object
 ' Souverains
 .SOVEREIGN_AAA = 0.05
                          ' 5% (quasi-zéro risque)
                        ' 10%
 .SOVEREIGN_AA = 0.1
                        ' 15%
 .SOVEREIGN_A = 0.15
                         ' 25%
 .SOVEREIGN_BBB = 0.25
 .SOVEREIGN_BB = 0.4 ' 40\%
 .SOVEREIGN_B_AND_BELOW = 0.6 '60%
 ' Corporates
 .CORPORATE_AAA = 0.15
                          ' 15%
                         '20%
 .CORPORATE_AA = 0.2
 .CORPORATE_A = 0.25
                         ' 25%
                          ′ 35%
 .CORPORATE\_BBB = 0.35
 .CORPORATE_BB = 0.5
                          ' 50%
 .CORPORATE_B = 0.65
                          ' 65%
 .CORPORATE_CCC_AND_BELOW = 0.8 '80%
 ' Banques
 .BANK_SYSTEMICALLY_IMPORTANT = 0.1 ' 10% (TBTF)
 .BANK_LARGE_DOMESTIC = 0.2 '20%
 .BANK_REGIONAL = 0.3 '30%
 .BANK_COOPERATIVE = 0.25
 ' Individuels
 .INDIVIDUAL_ULTRA_HIGH_NET_WORTH = 0.3 ′30% (>50M€)
 .INDIVIDUAL_HIGH_NET_WORTH = 0.45 ' 45% (5-50M€)
 .INDIVIDUAL_AFFLUENT = 0.6 ' 60% (1-5M€)
 .INDIVIDUAL STANDARD = 0.75 '75% (<1M€)
End Type
```

Taux d'Exécution par Juridiction

```
' === ENFORCEMENT RATES (Feuille "Database") ===
Type LegalEnforcementRates As Object
 ' Juridictions de référence
 .ENGLAND_WALES = 0.95
                      ' 95% (common law)
 .NEW_YORK = 0.95 '95% (commercial courts)
                    ' 93% (arbitrage)
 .SINGAPORE = 0.93
 .SWITZERLAND = 0.92 '92% (stabilité)
 ' Europe développée
                   ' 90%
 .FRANCE = 0.9
 .GERMANY = 0.91 '91%
                    ′ 89%
 .NETHERLANDS = 0.89
 .IRELAND = 0.88 '88%
 ' Europe du Sud
                 ' 80% (délais longs)
 .ITALY = 0.8
 .SPAIN = 0.82
                 ' 82%
 ' 78%
 ' Pays émergents
 .POLAND = 0.75 ' 75%
 .CZECH_REPUBLIC = 0.73 '73%
 .RUSSIA = 0.6
             ' 60%
 .TURKEY = 0.65
                   ' 65%
 ' Asie
 .JAPAN = 0.85
                   ' 90%
 .HONG_KONG = 0.9
 .SOUTH_KOREA = 0.8 '80%
 .CHINA = 0.7 ' 70%
 ' Autres
 .CANADA = 0.88
                   '88%
 .AUSTRALIA = 0.87 '87%
End Type
```

Facteurs de Corrélation

```
' === CORRÉLATIONS EMPRUNTEUR-GARANT (Feuille "Collateral") ===
Type CorrelationAdjustments As Object
 ' Même groupe/contrôle
 .SAME_ULTIMATE_PARENT = 0.3 '-70% (très corrélé)
 .SAME_GROUP_SISTER = 0.4 '-60% (même groupe)
 .SIGNIFICANT_SHAREHOLDING = 0.7 '-30% (participation)
 ' Même secteur
 .SAME_INDUSTRY_SUBSECTOR = 0.7 '-30% (sous-secteur)
 .SAME_INDUSTRY_BROAD = 0.85 '-15% (secteur large)
 .RELATED_INDUSTRIES = 0.9 '-10% (secteurs liés)
 ' Même géographie
 .SAME_COUNTRY_REGION = 0.85 '-15% (même région)
 .SAME_COUNTRY_DIFFERENT_REGION = 0.9 '-10%
 .DIFFERENT_COUNTRY_SAME_ZONE = 0.95 '-5% (UE, NAFTA)
 'Indépendant
 .DIFFERENT_SECTOR_GEOGRAPHY = 1.0 '0% (indépendant)
 .SOVEREIGN_VS_CORPORATE = 0.95 '-5% (corrélation macro)
End Type
```

Formules Garanties et Impacts

```
Function CalculateGuaranteeLGDImpact(loanAmount As Double, _
                  guaranteeAmount As Double, _
                  guarantorRating As String, _
                  guarantorType As String, _
                  jurisdiction As String, _
                  correlationFactor As Double) As LGDResult
  Dim result As LGDResult
  ' === CALCUL VALEUR ATTENDUE GARANTIE ===
  Dim guarantorPD As Double
  Dim guarantorLGD As Double
  Dim enforcementRate As Double
  guarantorPD = GetPDFromMapping(guarantorRating, "EXTERNAL")
  guarantorLGD = GetGuarantorLGD(guarantorType, guarantorRating)
  enforcementRate = GetEnforcementRate(jurisdiction)
  ' Valeur attendue brute
  Dim expectedGuaranteeValue As Double
  expectedGuaranteeValue = guaranteeAmount * (1 - guarantorPD) * (1 - guarantorLGD) * enforcementRate
  ' Ajustement corrélation
  Dim correlationAdjustedValue As Double
  correlationAdjustedValue = expectedGuaranteeValue * correlationFactor
  ' Taux de couverture effectif
  Dim effectiveCoverage As Double
  effectiveCoverage = correlationAdjustedValue / loanAmount
  ' === CALCUL LGD AVEC GARANTIE ===
  Dim baseLGD As Double
  baseLGD = 0.45
                           ' LGD de base sans garantie
  ' LGD ajustée selon formule Bâle III
  Dim adjustedLGD As Double
  If effectiveCoverage >= 1.0 Then
    adjustedLGD = guarantorLGD 'Substitution complète
  Else
    'LGD = LGD_base × (1 - Coverage) + LGD_garant × Coverage
    adjustedLGD = baseLGD * (1 - effectiveCoverage) + guarantorLGD * effectiveCoverage
  End If
  ' Minimum réglementaire
  If adjustedLGD < 0.1 Then adjustedLGD = 0.1
```

```
result.FinalLGD = adjustedLGD
result.ExpectedGuaranteeValue = correlationAdjustedValue
result.EffectiveCoverage = effectiveCoverage
result.LGDReduction = baseLGD - adjustedLGD

CalculateGuaranteeLGDImpact = result

End Function
```

TRADE FINANCE - PARAMÈTRES DÉTAILLÉS

Décotes par Type de Marchandises

```
vba
' === DÉCOTES TRADE FINANCE (Feuille "Database") ===
Type TradeGoodsDiscounts As Object
  ' Commodités standard
  .BULK_COMMODITIES_LIQUID = 0.05 '5% (pétrole, produits chimiques)
  .BULK_COMMODITIES_DRY = 0.1 '10% (minerai, céréales)
  .PRECIOUS_METALS = 0.02
                               ' 2% (or, argent)
  .BASE\_METALS = 0.08
                             ' 8% (cuivre, aluminium)# Modèles de Valorisation du Collatéral - Analyse Comp
## FEUILLES EXCEL DÉDIÉES AU COLLATÉRAL
### **1. Feuille "Collateral" - Valorisation des Sûretés**
### **2. Feuille "Guarantee" - Gestion des Garanties**
### **3. Feuille "LGD" - Impact sur Loss Given Default**
## 🏚 CATÉGORIE 1: COLLATÉRAL IMMOBILIER (REAL ESTATE)
### **Métriques de Valorisation**
```

Variables Obligatoires:

```
    Property Value [DECIMAL(15,2)] - Valeur expertise
    Property Address [VARCHAR(200)] - Localisation complète
    Property Type [VARCHAR(30)] - OFFICE/RETAIL/RESIDENTIAL/INDUSTRIAL
    Gross Floor Area [DECIMAL(10,2)] - Surface totale (m²)
    Net Operating Income [DECIMAL(12,2)] - Revenus nets annuels
    Operating Expenses [DECIMAL(10,2)] - Charges exploitation
    Last Valuation Date [DATE] - Date dernière expertise
```

Variables Facultatives:

├── Occupancy Rate [DECIMAL(5,2)] - Taux occupation (%)

├── Average Rent PSM [DECIMAL(8,2)] - Loyer moyen/m²

├── Lease Expiry Profile [VARCHAR(100)] - Échéancier baux

├── Capital Expenditures [DECIMAL(10,2)] - CapEx annuel

├── Property Manager [VARCHAR(100)] - Gestionnaire

└── Insurance Value [DECIMAL(15,2)] - Valeur assurance

```
### **Modèles de Valorisation Immobilière**
#### **1. MÉTHODE PAR CAPITALISATION (Income Approach)**
```vba
Function CalculatePropertyValueIncome(noi As Double, capRate As Double) As Double
 ' Valeur = Revenus Nets / Taux de Capitalisation
 If capRate <= 0 Then
 CalculatePropertyValueIncome = 0
 Exit Function
 End If
 Dim propertyValue As Double
 propertyValue = noi / capRate
 ' Validation limites
 If propertyValue < 0 Then propertyValue = 0
 CalculatePropertyValueIncome = propertyValue
End Function
' Calcul Cap Rate de marché
Function CalculateMarketCapRate(propertyType As String, _
 location As String, _
 quality As String) As Double
 Dim baseCapRate As Double
 Dim locationAdjustment As Double
 Dim qualityAdjustment As Double
 ' Cap Rate de base par type
 Select Case UCase(propertyType)
 Case "OFFICE CBD"
 baseCapRate = 0.045 ' 4.5% (prime)
 Case "OFFICE_SUBURBAN"
 baseCapRate = 0.055 ' 5.5%
 Case "RETAIL HIGH STREET"
 baseCapRate = 0.04 ' 4.0% (prime)
 Case "RETAIL_SHOPPING_CENTER"
 baseCapRate = 0.065 ' 6.5%
 Case "RESIDENTIAL RENTAL"
 baseCapRate = 0.035 ' 3.5%
 Case "INDUSTRIAL_LOGISTICS"
 baseCapRate = 0.055 ' 5.5%
```

```
Case "INDUSTRIAL_MANUFACTURING"
 baseCapRate = 0.08 '8.0%
 Case Else
 baseCapRate = 0.06 ' 6.0% par défaut
 End Select
 ' Ajustement localisation
 Select Case UCase(location)
 Case "PARIS_CBD", "LONDON_CITY", "MANHATTAN"
 locationAdjustment = -0.01 ' Prime location -100bp
 Case "MAJOR_CITY_CENTER"
 locationAdjustment = -0.005 ' -50bp
 Case "SUBURBAN"
 locationAdjustment = 0.005 ' +50bp
 Case "RURAL", "SECONDARY_CITY"
 locationAdjustment = 0.015 ' +150bp
 Case Else
 locationAdjustment = 0
 End Select
 ' Ajustement qualité
 Select Case UCase(quality)
 Case "GRADE_A", "PRIME"
 qualityAdjustment = -0.005 '-50bp
 Case "GRADE_B", "GOOD"
 qualityAdjustment = 0 'Neutral
 Case "GRADE_C", "FAIR"
 qualityAdjustment = 0.01 ' +100bp
 Case "DISTRESSED"
 qualityAdjustment = 0.02 '+200bp
 Case Else
 qualityAdjustment = 0
 End Select
 CalculateMarketCapRate = baseCapRate + locationAdjustment + qualityAdjustment
End Function
```

# 2. MÉTHODE COMPARATIVE (Sales Comparison Approach)

```
Function CalculatePropertyValueComparative(subjectProperty As PropertyData, _
 comparable Properties As Collection) As Double
 Dim adjusted Values As Collection
 Set adjustedValues = New Collection
 Dim comp As PropertyData
 For Each comp In comparable Properties
 ' Ajustements pour différences
 Dim adjusted Value As Double
 adjustedValue = comp.SalePrice
 ' Ajustement taille
 Dim sizeAdjustment As Double
 If subjectProperty.GFA <> comp.GFA Then
 sizeAdjustment = (subjectProperty.GFA - comp.GFA) / comp.GFA * 0.1 ' 10% per size difference
 adjustedValue = adjustedValue * (1 + sizeAdjustment)
 End If
 ' Ajustement âge/condition
 Dim ageAdjustment As Double
 Dim ageDifference As Integer
 ageDifference = subjectProperty.YearBuilt - comp.YearBuilt
 ageAdjustment = ageDifference * 0.005 ' 0.5% per year
 adjustedValue = adjustedValue * (1 + ageAdjustment)
 ' Ajustement localisation
 Dim locationAdjustment As Double
 locationAdjustment = GetLocationAdjustment(subjectProperty.Location, comp.Location)
 adjustedValue = adjustedValue * (1 + locationAdjustment)
 ' Ajustement date de vente
 Dim marketAdjustment As Double
 marketAdjustment = GetMarketAppreciation(comp.SaleDate, Date)
 adjustedValue = adjustedValue * (1 + marketAdjustment)
 adjustedValues.Add adjustedValue
 Next comp
```

' Calcul valeur moyenne pondérée

Dim totalValue As Double
Dim totalWeight As Double
Dim i As Integer

```
For i = 1 To adjustedValues.Count
Dim weight As Double
weight = GetComparableWeight(comparableProperties(i)) 'Poids selon fiabilité

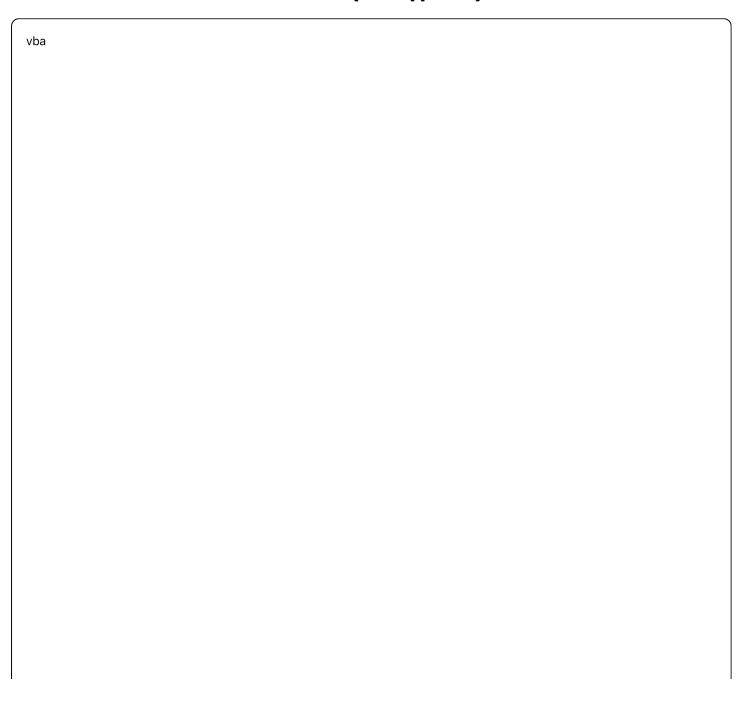
totalValue = totalValue + adjustedValues(i) * weight
totalWeight = totalWeight + weight
Next i

If totalWeight > 0 Then
CalculatePropertyValueComparative = totalValue / totalWeight

Else
CalculatePropertyValueComparative = 0
End If

End Function
```

# 3. MÉTHODE DU COÛT DE REMPLACEMENT (Cost Approach)



```
Function CalculatePropertyValueCost(landValue As Double, _
 buildingCostPSM As Double, _
 gfa As Double, _
 ageBuilding As Integer, _
 condition As String) As Double
 'Coût de construction actuel
 Dim replacementCost As Double
 replacementCost = buildingCostPSM * gfa
 ' Facteur de dépréciation selon âge
 Dim depreciationRate As Double
 Select Case condition
 Case "EXCELLENT"
 depreciationRate = 0.015 ' 1.5% par an
 Case "GOOD"
 depreciationRate = 0.02 ' 2.0% par an
 Case "AVERAGE"
 depreciationRate = 0.025 ' 2.5% par an
 Case "POOR"
 depreciationRate = 0.035 ' 3.5% par an
 Case Else
 depreciationRate = 0.025
 End Select
 ' Dépréciation accumulée
 Dim accumulated Depreciation As Double
 accumulatedDepreciation = 1 - (depreciationRate * ageBuilding)
 If accumulatedDepreciation < 0.2 Then accumulatedDepreciation = 0.2 'Minimum 20% valeur
 ' Valeur dépréciée du bâtiment
 Dim depreciatedBuildingValue As Double
 depreciatedBuildingValue = replacementCost * accumulatedDepreciation
 ' Valeur totale = Terrain + Bâtiment déprécié
 CalculatePropertyValueCost = landValue + depreciatedBuildingValue
End Function
```

# **Calcul LGD avec Collatéral Immobilier**

```
Function CalculateRealEstateLGD(loanAmount As Double, _
 propertyValue As Double, _
 propertyType As String, __
 marketCondition As String) As Double
 ' Calcul LTV
 Dim Itv As Double
 Itv = IoanAmount / propertyValue
 ' Décotes de liquidation par type
 Dim liquidationDiscount As Double
 Select Case UCase(propertyType)
 Case "OFFICE_PRIME"
 liquidationDiscount = 0.15 '15% décote
 Case "OFFICE_SECONDARY"
 liquidationDiscount = 0.25 '25% décote
 Case "RETAIL_PRIME"
 liquidationDiscount = 0.2 '20% décote
 Case "RETAIL_SECONDARY"
 liquidationDiscount = 0.35 '35% décote
 Case "RESIDENTIAL"
 liquidationDiscount = 0.1 '10% décote
 Case "INDUSTRIAL"
 liquidationDiscount = 0.3 '30% décote
 Case Else
 liquidationDiscount = 0.25
 End Select
 ' Ajustement conditions de marché
 Select Case UCase(marketCondition)
 Case "STRONG"
 liquidationDiscount = liquidationDiscount * 0.8 '-20%
 Case "NORMAL"
 ' Pas d'ajustement
 Case "WEAK"
 liquidationDiscount = liquidationDiscount * 1.3 '+30%
 Case "DISTRESSED"
 liquidationDiscount = liquidationDiscount * 1.6 '+60%
 End Select
 ' Valeur de liquidation
 Dim liquidationValue As Double
 liquidationValue = propertyValue * (1 - liquidationDiscount)
 'Coûts de réalisation (frais juridiques, commissaire-priseur, etc.)
 Dim realizationCosts As Double
```

# A CATÉGORIE 2: COLLATÉRAL VÉHICULES/ÉQUIPEMENTS

### Variables de Valorisation

```
Variables Obligatoires:
 — Asset Type
 [VARCHAR(50)] - VEHICLE/MACHINERY/EQUIPMENT
 — Make and Model
 [VARCHAR(100)] - Marque et modèle
 — Year of Manufacture [INTEGER] - Année fabrication
 — Purchase Price
 [DECIMAL(12,2)] - Prix d'achat initial

 Current Market Value [DECIMAL(12,2)] - Valeur marché actuelle

 Condition
 [VARCHAR(20)] - EXCELLENT/GOOD/FAIR/POOR
 - Mileage/Hours
 [INTEGER] - Kilométrage/heures usage
Variables Facultatives:
 VIN/Serial Number [VARCHAR(50)] - Numéro série

 Maintenance Records [BOOLEAN] - Historique entretien

 Insurance Coverage [DECIMAL(10,2)] - Couverture assurance

 Registration Status [VARCHAR(20)] - Statut immatriculation

 Location
 [VARCHAR(100)] - Localisation actuelle
```

# Modèle de Dépréciation Véhicules

```
Function CalculateVehicleValue(purchasePrice As Double, _
 ageYears As Integer, _
 mileage As Long, _
 vehicleType As String, _
 condition As String) As Double
 Dim baseDepreciationRate As Double
 Dim mileageAdjustment As Double
 Dim conditionAdjustment As Double
 ' Taux de dépréciation de base par type
 Select Case UCase(vehicleType)
 Case "LUXURY_CAR"
 baseDepreciationRate = 0.25 '25% première année, puis 15%/an
 Case "STANDARD_CAR"
 baseDepreciationRate = 0.2 '20% première année, puis 12%/an
 Case "COMMERCIAL_VEHICLE"
 baseDepreciationRate = 0.15 '15% première année, puis 10%/an
 Case "TRUCK_HEAVY"
 baseDepreciationRate = 0.12 '12% première année, puis 8%/an
 Case "CONSTRUCTION_EQUIPMENT"
 baseDepreciationRate = 0.1 '10% première année, puis 7%/an
 Case Else
 baseDepreciationRate = 0.15
 End Select
 ' Calcul dépréciation cumulative
 Dim depreciated Value As Double
 If ageYears = 0 Then
 depreciatedValue = purchasePrice
 Elself ageYears = 1 Then
 depreciatedValue = purchasePrice * (1 - baseDepreciationRate)
 Else
 ' Première année + années suivantes
 depreciatedValue = purchasePrice * (1 - baseDepreciationRate)
 depreciatedValue = depreciatedValue * ((1 - baseDepreciationRate * 0.6) ^ (ageYears - 1))
 End If
 ' Ajustement kilométrage (pour véhicules)
 If UCase(vehicleType) Like "*CAR*" Or UCase(vehicleType) Like "*VEHICLE*" Then
 Dim expectedMileage As Long
 expectedMileage = ageYears * 15000 ' 15k km/an standard
 If mileage > expectedMileage Then
 Dim excessMileage As Long
 excessMileage = mileage - expectedMileage
```

```
mileageAdjustment = -(excessMileage / 1000) * 0.005 ' -0.5% par 1000km excès
 Else
 mileageAdjustment = 0
 End If
 depreciatedValue = depreciatedValue * (1 + mileageAdjustment)
 End If
 ' Ajustement condition
 Select Case UCase(condition)
 Case "EXCELLENT"
 conditionAdjustment = 1.1 '+10%
 Case "GOOD"
 conditionAdjustment = 1.0 'Baseline
 Case "FAIR"
 conditionAdjustment = 0.85 '-15%
 Case "POOR"
 conditionAdjustment = 0.65 '-35%
 Case Else
 conditionAdjustment = 0.9
 End Select
 depreciatedValue = depreciatedValue * conditionAdjustment
 ' Valeur résiduelle minimum
 Dim minimumValue As Double
 minimumValue = purchasePrice * 0.05 ' 5% minimum valeur résiduelle
 If depreciatedValue < minimumValue Then
 depreciatedValue = minimumValue
 End If
 CalculateVehicleValue = depreciatedValue
End Function
```

# **VALUE SE COLLATÉRAL MARCHANDISES/STOCKS**

### **Variables de Valorisation**

Variables Obligatoires:	
Commodity Type [VARCHAR(50)] - Type marchandise	
Quantity [DECIMAL(12,3)] - Quantité	
Unit of Measure [VARCHAR(20)] - Unité (MT, BBL, OZ, etc.)	
Current Market Price [DECIMAL(10,4)] - Prix marché unitaire	
Location [VARCHAR(100)] - Lieu stockage	
—— Quality Grade [VARCHAR(30)] - Grade qualité	
Storage Costs [DECIMAL(8,2)] - Coûts stockage/mois	
Variables Facultatives:	
Expiry Date [DATE] - Date péremption	
Insurance Coverage [DECIMAL(12,2)] - Couverture assurance	
Warehouse Receipts [BOOLEAN] - Récépissés entrepôt	
Certification [VARCHAR(50)] - Certifications qualité	
L—— Minimum Sale Quantity [DECIMAL(10,3)] - Lot minimum vente	

# **Modèle de Valorisation Commodités**

vba		
123		
1		ı

```
Function CalculateCommodityValue(commodityType As String, _
 quantity As Double, _
 marketPrice As Double, _
 qualityGrade As String, _
 location As String, _
 daysToExpiry As Integer) As Double
 Dim baseValue As Double
 Dim qualityAdjustment As Double
 Dim locationAdjustment As Double
 Dim liquidityAdjustment As Double
 Dim expiryAdjustment As Double
 ' Valeur de base
 baseValue = quantity * marketPrice
 ' Ajustement qualité par type de commodité
 Select Case UCase(commodityType)
 Case "CRUDE_OIL"
 Select Case UCase(qualityGrade)
 Case "BRENT", "WTI"
 qualityAdjustment = 1.0 'Premium grade
 Case "HEAVY_CRUDE"
 qualityAdjustment = 0.9 '-10%
 Case "SOUR_CRUDE"
 qualityAdjustment = 0.85 '-15%
 Case Else
 qualityAdjustment = 0.95
 End Select
 Case "GOLD"
 Select Case UCase(qualityGrade)
 Case "999_FINE", "LBMA_GOOD"
 qualityAdjustment = 1.0
 Case "22_CARAT"
 qualityAdjustment = 0.92
 Case "18_CARAT"
 qualityAdjustment = 0.75
 Case Else
 qualityAdjustment = 0.9
 End Select
 Case "WHEAT", "CORN", "SOYBEANS"
 Select Case UCase(qualityGrade)
 Case "GRADE_1", "PREMIUM"
 qualityAdjustment = 1.0
```

```
Case "GRADE_2", "STANDARD"
 qualityAdjustment = 0.95
 Case "GRADE_3", "FEED"
 qualityAdjustment = 0.85
 Case Else
 qualityAdjustment = 0.9
 End Select
 Case Else
 qualityAdjustment = 0.95 ' Décote générale 5%
End Select
' Ajustement localisation (accès aux marchés)
Select Case UCase(location)
 Case "MAJOR_PORT", "EXCHANGE_DELIVERY_POINT"
 locationAdjustment = 1.0 'Prix de référence
 Case "INLAND_TERMINAL"
 locationAdjustment = 0.97 '-3% transport
 Case "REMOTE_STORAGE"
 locationAdjustment = 0.9 '-10% transport + risque
 Case Else
 locationAdjustment = 0.95
End Select
' Ajustement liquidité (facilité de vente)
Select Case UCase(commodityType)
 Case "GOLD", "SILVER", "CRUDE_OIL"
 liquidityAdjustment = 0.95 'Très liquide, -5% seulement
 Case "COPPER", "ALUMINUM", "WHEAT"
 liquidityAdjustment = 0.9 'Liquide, -10%
 Case "SPECIALTY_METALS", "RARE_EARTHS"
 liquidityAdjustment = 0.8 'Moins liquide, -20%
 Case Else
 liquidityAdjustment = 0.85 'Liquidité moyenne, -15%
End Select
' Ajustement péremption/détérioration
If daysToExpiry > 0 Then
 Select Case UCase(commodityType)
 Case "AGRICULTURAL_PERISHABLE"
 If daysToExpiry <= 30 Then
 expiryAdjustment = 0.7 '-30% si < 1 mois
 Elself daysToExpiry <= 90 Then
 expiryAdjustment = 0.85 '-15% si < 3 mois
 expiryAdjustment = 1.0
 End If
```

```
Case "PETROLEUM_PRODUCTS"
 If daysToExpiry <= 60 Then
 expiryAdjustment = 0.9 '-10% si < 2 mois
 Else
 expiryAdjustment = 1.0
 End If
 Case Else
 expiryAdjustment = 1.0 'Pas de péremption
 End Select
 Else
 expiryAdjustment = 1.0
 End If
 ' Calcul valeur finale
 Dim finalValue As Double
 finalValue = baseValue * qualityAdjustment * locationAdjustment * _
 liquidityAdjustment * expiryAdjustment
 CalculateCommodityValue = finalValue
End Function
```

# **CATÉGORIE 4: GARANTIES PERSONNELLES**

# Variables de Valorisation

```
Variables Obligatoires:
 — Guarantor Name
 [VARCHAR(100)] - Nom du garant
 —— Guarantor Type
 [VARCHAR(30)] - CORPORATE/INDIVIDUAL/SOVEREIGN
 — Guarantor Rating
 [VARCHAR(10)] - Notation du garant
 — Guarantee Amount
 [DECIMAL(15,2)] - Montant garanti
 — Coverage Ratio
 [DECIMAL(5,4)] - Taux de couverture
 — Guarantee Type
 [VARCHAR(30)] - FULL/PARTIAL/SPECIFIC

 Legal Enforceability [BOOLEAN] - Opposabilité juridique

Variables Facultatives:
 - Guarantor Net Worth [DECIMAL(15,2)] - Patrimoine net
 — Guarantor Income
 [DECIMAL(12,2)] - Revenus annuels

 Cross Default Clauses [BOOLEAN] - Clauses défaut croisé

 Collateral Supporting [BOOLEAN] - Garantie contre-garantie

 — Jurisdiction
 [VARCHAR(50)] - Juridiction applicable
```

### **Modèle de Valorisation Garanties**

vba	

```
Function CalculateGuaranteeValue(guarantorRating As String, _
 guaranteeAmount As Double, _
 coverageRatio As Double, _
 guarantorType As String, _
 jurisdiction As String) As Double
 Dim guarantorPD As Double
 Dim guarantorLGD As Double
 Dim legalEnforcementRate As Double
 Dim correlationAdjustment As Double
 ' PD du garant selon sa notation
 guarantorPD = GetPDFromMapping(guarantorRating, "EXTERNAL")
 'LGD du garant selon son type
 Select Case UCase(guarantorType)
 Case "SOVEREIGN_AAA_TO_A"
 guarantorLGD = 0.15 '15% (souverain investment grade)
 Case "SOVEREIGN_BBB_TO_BB"
 guarantorLGD = 0.35 ' 35% (souverain speculative)
 Case "CORPORATE_INVESTMENT_GRADE"
 guarantorLGD = 0.25 ' 25% (corporate IG)
 Case "CORPORATE_SPECULATIVE"
 guarantorLGD = 0.55 ' 55% (corporate spec)
 Case "INDIVIDUAL_HIGH_NET_WORTH"
 guarantorLGD = 0.45 ' 45% (personne physique riche)
 Case "INDIVIDUAL_STANDARD"
 guarantorLGD = 0.75 ' 75% (personne physique standard)
 Case Else
 guarantorLGD = 0.6 '60% par défaut
 End Select
 ' Taux d'exécution juridique par juridiction
 Select Case UCase(jurisdiction)
 Case "ENGLAND", "NEW_YORK", "SINGAPORE"
 legalEnforcementRate = 0.95 '95% enforcement
 Case "FRANCE", "GERMANY", "SWITZERLAND"
 legalEnforcementRate = 0.9 '90% enforcement
 Case "ITALY", "SPAIN", "EMERGING_EUROPE"
 legalEnforcementRate = 0.8 '80% enforcement
 Case "EMERGING_ASIA", "LATIN_AMERICA"
 legalEnforcementRate = 0.7 '70% enforcement
 Case "AFRICA", "MIDDLE_EAST"
 legalEnforcementRate = 0.6 '60% enforcement
 Case Else
 legalEnforcementRate = 0.75 '75% par défaut
```

```
End Select
 ' Ajustement corrélation emprunteur-garant
 'Si même secteur/pays, corrélation élevée = décote
 correlationAdjustment = GetCorrelationAdjustment(guarantorType)
 ' Valeur attendue de la garantie
 Dim expectedRecovery As Double
 expectedRecovery = guaranteeAmount * coverageRatio * (1 - guarantorPD) * _
 (1 - guarantorLGD) * legalEnforcementRate * correlationAdjustment
 CalculateGuaranteeValue = expectedRecovery
End Function
Private Function GetCorrelationAdjustment(guarantorType As String) As Double
 ' Ajustement pour corrélation emprunteur-garant
 Select Case UCase(guarantorType)
 Case "SOVEREIGN_SAME_COUNTRY"
 ' -20% corrélation élevée
 GetCorrelationAdjustment = 0.8
 Case "CORPORATE_SAME_GROUP"
 GetCorrelationAdjustment = 0.7 '-30% très corrélé
 Case "CORPORATE_SAME_SECTOR"
 GetCorrelationAdjustment = 0.85 '-15% corrélé
 Case "CORPORATE_DIFFERENT_SECTOR"
 GetCorrelationAdjustment = 0.95 '-5% peu corrélé
 Case "INDIVIDUAL UNRELATED"
 GetCorrelationAdjustment = 1.0 'Pas de corrélation
 Case Else
 GetCorrelationAdjustment = 0.9 '-10% par défaut
 End Select
End Function
```

# **CATÉGORIE 5: COLLATÉRAL TRADE FINANCE**

### Variables de Valorisation

Variables Obligatoires:
Document Type [VARCHAR(30)] - BL/INVOICE/LC/INSURANCE
Goods Description [VARCHAR(200)] - Description marchandises
Goods Value [DECIMAL(15,2)] - Valeur déclarée
Country of Origin [VARCHAR(50)] - Pays d'origine
Country of Destination [VARCHAR(50)] - Pays destination
Transportation Mode [VARCHAR(20)] - SEA/AIR/LAND/MULTIMODAL
Insurance Coverage [DECIMAL(15,2)] - Couverture assurance
Variables Facultatives:
Inspection Certificate [BOOLEAN] - Certificat inspection
Phytosanitary Cert [BOOLEAN] - Certificat sanitaire
Warehouse Receipts [BOOLEAN] - Récépissés entrepôt
Force Majeure Coverage [BOOLEAN] - Couverture force majeure
Political Risk Ins [BOOLEAN] - Assurance risque politique

# **Modèle de Valorisation Trade Finance**

vba			

```
Function CalculateTradeFinanceCollateral(goodsValue As Double, _
 goodsType As String, _
 countryOrigin As String, _
 countryDestination As String, _
 transportMode As String, _
 insuranceCoverage As Double, _
 documentQuality As String) As Double
 Dim baseValue As Double
 Dim goodsRiskAdjustment As Double
 Dim countryRiskAdjustment As Double
 Dim transportRiskAdjustment As Double
 Dim documentRiskAdjustment As Double
 Dim insuranceAdjustment As Double
 baseValue = goodsValue
 ' === AJUSTEMENT TYPE DE MARCHANDISE ===
 Select Case UCase(goodsType)
 Case "BULK_COMMODITIES"
 goodsRiskAdjustment = 0.9 '-10% (volatilité prix)
 Case "MANUFACTURED_GOODS"
 goodsRiskAdjustment = 0.85 '-15% (dépréciation)
 Case "PERISHABLE_GOODS"
 goodsRiskAdjustment = 0.7 '-30% (périssable)
 Case "HAZARDOUS_MATERIALS"
 goodsRiskAdjustment = 0.75 '-25% (risques spéciaux)
 Case "LUXURY_ITEMS"
 goodsRiskAdjustment = 0.8 '-20% (marché étroit)
 Case "PRECIOUS METALS"
 goodsRiskAdjustment = 0.95 '-5% (très liquide)
 Case Else
 goodsRiskAdjustment = 0.85 '-15% par défaut
 End Select
 ' === AJUSTEMENT RISQUE PAYS ===
 Dim originRisk As Double
 Dim destinationRisk As Double
 originRisk = GetCountryRiskScore(countryOrigin) '0-100 scale
 destinationRisk = GetCountryRiskScore(countryDestination)
 ' Risque combiné (pire des deux)
 Dim worstCountryRisk As Double
 worstCountryRisk = Application.WorksheetFunction.Min(originRisk, destinationRisk)
```

```
If worstCountryRisk >= 80 Then
 countryRiskAdjustment = 1.0
 ' Pays développés
Elself worstCountryRisk >= 60 Then
 ' Pays émergents stables
 countryRiskAdjustment = 0.95
Elself worstCountryRisk >= 40 Then
 countryRiskAdjustment = 0.85 'Pays émergents volatils
Else
 countryRiskAdjustment = 0.7 'Pays à haut risque
End If
' === AJUSTEMENT MODE TRANSPORT ===
Select Case UCase(transportMode)
 Case "SEA_CONTAINER"
 transportRiskAdjustment = 0.95 '-5% (standard, sûr)
 Case "SEA_BULK"
 transportRiskAdjustment = 0.9 '-10% (risque météo)
 Case "AIR_CARGO"
 transportRiskAdjustment = 0.98 '-2% (rapide, sûr)
 Case "LAND_TRUCK"
 transportRiskAdjustment = 0.92 '-8% (accidents)
 Case "LAND_RAIL"
 transportRiskAdjustment = 0.94 '-6% (retards)
 Case "MULTIMODAL"
 transportRiskAdjustment = 0.88 '-12% (complexité)
 Case Else
 transportRiskAdjustment = 0.9 '-10% par défaut
End Select
' === AJUSTEMENT QUALITÉ DOCUMENTAIRE ===
Select Case UCase(documentQuality)
 Case "FULL_SET_CLEAN"
 Case "FULL_SET_CLAUSED"
 documentRiskAdjustment = 0.95 'Réserves mineures
 Case "PARTIAL SET"
 Case "TRUST RECEIPT"
 documentRiskAdjustment = 0.8 'Confiance seule
 Case "NO DOCUMENTS"
 documentRiskAdjustment = 0.6 'Pas de documents
 Case Else
 documentRiskAdjustment = 0.9 'Standard
End Select
' === AJUSTEMENT COUVERTURE ASSURANCE ===
Dim insuranceRatio As Double
insuranceRatio = insuranceCoverage / goodsValue
```

```
If insuranceRatio >= 1.1 Then
 insuranceAdiustment = 1.0
 'Surassurance (110%+)
 Elself insuranceRatio >= 1.0 Then
 insuranceAdjustment = 0.98
 ' Assurance complète
 Elself insuranceRatio >= 0.9 Then
 insuranceAdjustment = 0.95
 ' Assurance quasi-complète
 Elself insuranceRatio >= 0.7 Then
 insuranceAdjustment = 0.9
 ' Assurance partielle
 Else
 insuranceAdjustment = 0.8
 'Sous-assurance
 End If
 ' === CALCUL VALEUR FINALE ===
 Dim collateralValue As Double
 collateralValue = baseValue * goodsRiskAdjustment * countryRiskAdjustment *
 transportRiskAdjustment * documentRiskAdjustment * insuranceAdjustment
 CalculateTradeFinanceCollateral = collateralValue
End Function
```

# **CATÉGORIE 6: COLLATÉRAL STRUCTURED FINANCE**

### Variables de Valorisation

```
Variables Obligatoires:
 — Project Name
 [VARCHAR(100)] - Nom du projet
 — Project Phase
 [VARCHAR(20)] - CONSTRUCTION/OPERATION
 — Revenue Contracts
 [DECIMAL(15,2)] - Contrats revenus sécurisés
 — Physical Assets Value [DECIMAL(15,2)] - Valeur actifs physiques

 Reserve Account Size [DECIMAL(12,2)] - Comptes de réserve

 Technology Risk [VARCHAR(20)] - PROVEN/NEW/EXPERIMENTAL
 - Off-taker Rating
 [VARCHAR(10)] - Notation acheteur
Variables Facultatives:
 Construction Progress [DECIMAL(5,2)] - % avancement (si construction)

 Operating History

 [INTEGER] - Années opérationnelles

 Regulatory Approvals [BOOLEAN] - Autorisations obtenues

 Environmental Permits [BOOLEAN] - Permis environnement

 Sponsor Support [VARCHAR(20)] - FULL/LIMITED/NONE
 — Political Risk Ins
 [DECIMAL(12,2)] - Assurance risque politique
```

# **Modèle de Valorisation Structured Finance**

```
Function CalculateProjectCollateral(revenueContracts As Double, _
 physicalAssets As Double, _
 reserveAccounts As Double, _
 projectPhase As String, _
 technologyRisk As String, _
 offtakerRating As String, _
 constructionProgress As Double) As Double
 Dim contractValue As Double
 Dim assetValue As Double
 Dim reserveValue As Double
 Dim phaseAdjustment As Double
 Dim technologyAdjustment As Double
 Dim offtakerAdjustment As Double
 ' === VALORISATION CONTRATS DE REVENUS ===
 ' Actualisation selon qualité off-taker
 Dim offtakerPD As Double
 offtakerPD = GetPDFromMapping(offtakerRating, "EXTERNAL")
 ' Valeur actuarielle des contrats
 contractValue = revenueContracts * (1 - offtakerPD) * GetOfftakerAdjustment(offtakerRating)
 ' === VALORISATION ACTIFS PHYSIQUES ===
 ' Décote selon type et liquidité
 Select Case UCase(projectPhase)
 Case "CONSTRUCTION"
 ' Actifs en construction - forte décote
 assetValue = physicalAssets * 0.4 ' 40% de la valeur
 Case "COMMISSIONING"
 'Tests et mise en service
 assetValue = physicalAssets * 0.6 ' 60% de la valeur
 Case "OPERATION_EARLY"
 ' Début exploitation (< 2 ans)
 assetValue = physicalAssets * 0.75 ' 75% de la valeur
 Case "OPERATION MATURE"
 'Exploitation mature (> 2 ans)
 assetValue = physicalAssets * 0.85 ' 85% de la valeur
 Case Else
 assetValue = physicalAssets * 0.5 ' 50% par défaut
 End Select
 ' === AJUSTEMENT PHASE PROJET ===
 Select Case UCase(projectPhase)
 Case "CONSTRUCTION"
 If constructionProgress < 0.3 Then
```

```
Elself constructionProgress < 0.7 Then
 phaseAdjustment = 0.75 'Mi-construction
 Else
 phaseAdjustment = 0.9 'Fin construction
 End If
 Case "OPERATION"
 phaseAdjustment = 1.0 'Pleine valeur
 Case Else
 phaseAdjustment = 0.8
 End Select
 ' === AJUSTEMENT RISQUE TECHNOLOGIQUE ===
 Select Case UCase(technologyRisk)
 Case "PROVEN_TECHNOLOGY"
 technologyAdjustment = 1.0 'Technologie éprouvée
 Case "ENHANCED_TECHNOLOGY"
 technologyAdjustment = 0.95 'Technologie améliorée
 Case "NEW_TECHNOLOGY"
 technologyAdjustment = 0.85 'Nouvelle technologie
 Case "EXPERIMENTAL"
 Case Else
 technologyAdjustment = 0.9
 End Select
 ' === VALORISATION COMPTES DE RÉSERVE ===
 'Liquidité quasi-totale mais restrictions d'usage
 reserveValue = reserveAccounts * 0.95 ' 95% de liquidité
 ' === CALCUL VALEUR TOTALE ===
 Dim totalCollateralValue As Double
 totalCollateralValue = (contractValue + assetValue) * phaseAdjustment * _
 technologyAdjustment + reserveValue
 CalculateProjectCollateral = totalCollateralValue
End Function
Private Function GetOfftakerAdjustment(offtakerRating As String) As Double
 ' Ajustement selon notation de l'acheteur
 Select Case UCase(offtakerRating)
 Case "AAA", "AA+"
 GetOfftakerAdjustment = 0.98 'Quasi-souverain
 Case "AA", "AA-", "A+"
 GetOfftakerAdjustment = 0.95 'Investment grade élevé
 Case "A", "A-", "BBB+"
```

```
GetOfftakerAdjustment = 0.9 'Investment grade

Case "BBB", "BBB-"

GetOfftakerAdjustment = 0.85 'Investment grade faible

Case "BB+", "BB", "BB-"

GetOfftakerAdjustment = 0.75 'Speculative grade

Case "B+", "B", "B-"

GetOfftakerAdjustment = 0.6 'Haut risque

Case Else

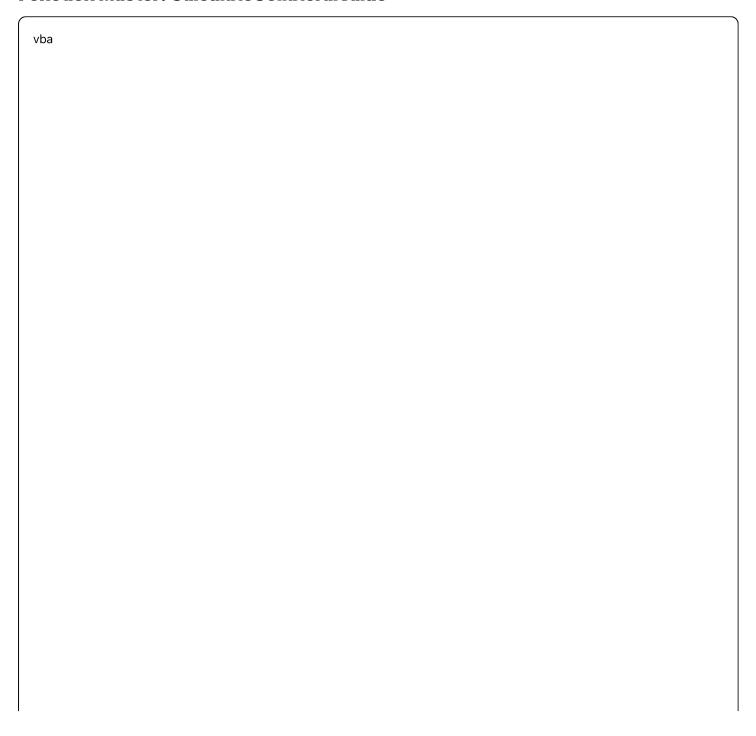
GetOfftakerAdjustment = 0.5 'Très haut risque

End Select

End Function
```

# **II** FONCTION PRINCIPALE D'ORCHESTRATION

# **Fonction Master: CalculateCollateralValue**



```
Public Function CalculateCollateralValue(applicationCode As String) As CollateralResult
 Dim result As CollateralResult
 Dim facilityData As FacilityData
 Dim collateralData As Collection
 ' === RÉCUPÉRATION DONNÉES ===
 facilityData = GetFacilityData(applicationCode)
 Set collateralData = GetCollateralData(applicationCode)
 ' === INITIALISATION ===
 result.TotalCollateralValue = 0
 result.WeightedAverageHaircut = 0
 result.CollateralCoverage = 0
 ' === VALORISATION PAR TYPE DE COLLATÉRAL ===
 Dim collateral As CollateralItem
 Dim itemValue As Double
 Dim totalWeight As Double
 For Each collateral In collateral Data
 Select Case UCase(collateral.CollateralType)
 Case "REAL_ESTATE"
 itemValue = CalculateRealEstateValue(collateral)
 Case "VEHICLE", "EQUIPMENT", "MACHINERY"
 itemValue = CalculateVehicleValue(collateral.PurchasePrice, _
 collateral.AgeYears, _
 collateral.Mileage, _
 collateral.AssetType, _
 collateral.Condition)
 Case "COMMODITIES", "INVENTORY", "STOCK"
 itemValue = CalculateCommodityValue(collateral.CommodityType, _
 collateral.Quantity, _
 collateral.MarketPrice, _
 collateral.QualityGrade, _
 collateral.Location, _
 collateral.DaysToExpiry)
 Case "PERSONAL_GUARANTEE", "CORPORATE_GUARANTEE"
 itemValue = CalculateGuaranteeValue(collateral.GuarantorRating, _
 collateral.GuaranteeAmount, _
```

collateral.CoverageRatio, \_

```
collateral.GuarantorType, _
 collateral.Jurisdiction)
 Case "TRADE_FINANCE_GOODS"
 itemValue = CalculateTradeFinanceCollateral(collateral.GoodsValue, _
 collateral.GoodsType, _
 collateral.CountryOrigin, _
 collateral.CountryDestination, _
 collateral.TransportMode, _
 collateral.InsuranceCoverage, _
 collateral.DocumentQuality)
 Case "PROJECT_ASSETS", "STRUCTURED_FINANCE"
 itemValue = CalculateProjectCollateral(collateral.RevenueContracts, _
 collateral.PhysicalAssets, _
 collateral.ReserveAccounts,
 collateral.ProjectPhase, _
 collateral.TechnologyRisk, _
 collateral.OfftakerRating,
 collateral.ConstructionProgress)
 Case Flse
 itemValue = collateral.BookValue * 0.5 ' 50% décote par défaut
 End Select
 ' === ACCUMULATION RÉSULTATS ===
 collateral.MarketValue = itemValue
 collateral.Haircut = (collateral.BookValue - itemValue) / collateral.BookValue
 result.TotalCollateralValue = result.TotalCollateralValue + itemValue
 result.WeightedAverageHaircut = result.WeightedAverageHaircut + _
 (collateral.Haircut * collateral.BookValue)
 totalWeight = totalWeight + collateral.BookValue
 ' Ajout au détail
 result.CollateralDetails.Add collateral
Next collateral
' === CALCULS FINAUX ===
If totalWeight > 0 Then
 result.WeightedAverageHaircut = result.WeightedAverageHaircut / totalWeight
End If
result.CollateralCoverage = result.TotalCollateralValue / facilityData.OutstandingAmount
result.IsFullyCovered = (result.CollateralCoverage >= 1.0)
```

' === CALCUL LGD AJUSTÉE ===

```
Dim baseLGD As Double
 baseLGD = GetBaseLGD(facilityData.Sector)
 If result.CollateralCoverage >= 1.0 Then
 ' Sur-collatéralisé
 result.AdjustedLGD = baseLGD * 0.3 ' 70% réduction LGD
 Elself result.CollateralCoverage >= 0.8 Then
 ' Bien collatéralisé
 result.AdjustedLGD = baseLGD * 0.5 ' 50% réduction LGD
 Elself result.CollateralCoverage >= 0.5 Then
 ' Partiellement collatéralisé
 result.AdjustedLGD = baseLGD * 0.75 ' 25% réduction LGD
 Else
 ' Mal collatéralisé
 result.AdjustedLGD = baseLGD * 0.95 ' 5% réduction LGD seulement
 End If
 ' === MISE À JOUR FEUILLES ===
 Call UpdateCollateralSheet(applicationCode, result)
 Call UpdateLGDSheet(applicationCode, result.AdjustedLGD)
 CalculateCollateralValue = result
End Function
```

# TABLEAU RÉCAPITULATIF DES MODÈLES

# Synthèse des 6 Catégories de Collatéral

Туре	Modèles	Variables Clés	Décotes Typiques	Complexité
Real Estate	3 méthodes (Income, Comparative, Cost)	NOI, Cap Rate, LTV, Location	10-35%	****
Véhicules/ Équip	Dépréciation + ajustements	Âge, Kilométrage, Condition, Type	15-60%	***
Commodités	Prix marché + ajustements qualité	Prix, Grade, Location, Expiry	5-30%	***
Garanties	Valeur actuarielle ajustée	PD Garant, LGD, Corrélation	20-80%	***
Trade Finance	Valeur marchandises ajustée	Pays, Transport, Documents	10-40%	***
Structured	Cash flows + actifs physiques	Contrats, Phase, Technologie	15-60%	***

# **Utilités dans le Système**

- Calcul LGD : Réduction selon taux de couverture
- RWA Bâle III : Substitution risque si garantie éligible
- IFRS 9 : Impact sur ECL via LGD ajustée
- **Pricing** : Ajustement marge selon collatéralisation
- Limites : Concentration par type de collatéral

# **Scripts VBA Associés**

vba ' Fonctions principales CalculateCollateralValue() - Orchestration générale CalculateRealEstateValue() - Immobilier (3 méthodes) CalculateVehicleValue() - Véhicules/équipements CalculateCommodityValue() - Marchandises/stocks CalculateGuaranteeValue() - Garanties personnelles CalculateTradeFinanceCollateral() - Trade finance CalculateProjectCollateral() - Structured finance ' Fonctions support - Lecture données collatéral GetCollateralData() UpdateCollateralSheet() - Mise à jour feuilles ApplyHaircuts() - Application décotes ValidateCollateralEligibility() - Éligibilité réglementaire

Cette infrastructure complète permet une valorisation sophistiquée de **6 types de collatéral** avec des modèles adaptés à chaque classe d'actifs ! ©