

# **Manuel de Préparation: Business Intelligence - Contexte Banque Commerciale**

## **Introduction**

La Business Intelligence (BI) dans le secteur bancaire consiste à transformer les données en insights actionnables pour la prise de décision. Ce manuel couvre les concepts, métriques et applications spécifiques au contexte des banques commerciales, avec un focus sur le marché haïtien.

---

## **Partie 1: Fondamentaux de la BI Bancaire**

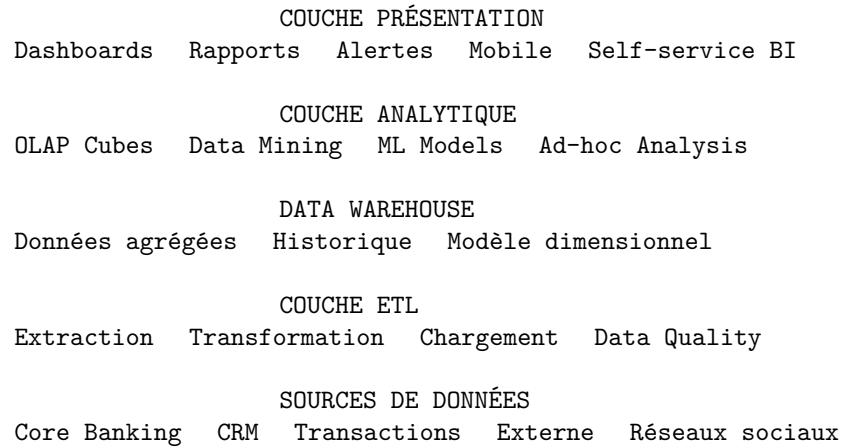
### **1.1 Définition et Objectifs**

**Business Intelligence:** Ensemble des technologies, applications et pratiques pour collecter, intégrer, analyser et présenter les informations business.

#### **Objectifs en Banque**

- **Pilotage de la performance:** Suivi des KPIs financiers et opérationnels
- **Gestion des risques:** Détection précoce des problèmes
- **Conformité réglementaire:** Reporting BRH et autres autorités
- **Connaissance client:** Segmentation et personnalisation
- **Optimisation opérationnelle:** Efficacité des processus

### **1.2 Architecture BI Bancaire**



---

## **Partie 2: Métriques Bancaires Essentielles (KPIs)**

### **2.1 Indicateurs de Rentabilité**

#### **Return on Equity (ROE)**

ROE = Résultat Net / Capitaux Propres Moyens × 100

Benchmark: 12-15% pour une banque performante

Interprétation: Rendement des fonds investis par les actionnaires

### **Return on Assets (ROA)**

ROA = Résultat Net / Total Actifs Moyens × 100

Benchmark: 1-2%

Interprétation: Efficacité de l'utilisation des actifs

### **Net Interest Margin (NIM)**

NIM = (Revenus d'intérêts - Charges d'intérêts) / Actifs Productifs × 100

Benchmark: 3-4%

Interprétation: Marge sur l'activité d'intermédiation

### **Cost-to-Income Ratio (CIR)**

CIR = Charges d'exploitation / Produit Net Bancaire × 100

Benchmark: 50-60%

Interprétation: Efficacité opérationnelle (plus bas = mieux)

### **Fee Income Ratio**

Fee Ratio = Commissions Nettes / Produit Net Bancaire × 100

Interprétation: Diversification des revenus

## **2.2 Indicateurs de Qualité des Actifs**

### **Non-Performing Loans Ratio (NPL Ratio)**

NPL Ratio = Prêts Non Performants / Total des Prêts × 100

Benchmark: < 5%

Classification NPL: Généralement > 90 jours d'impayés

### **Provision Coverage Ratio**

Coverage = Provisions pour Pertes / NPL × 100

Benchmark: > 100%

Interprétation: Capacité à absorber les pertes

### **Cost of Risk**

Cost of Risk = Dotations aux Provisions / Encours Moyen de Prêts × 100

Interprétation: Coût de la prise de risque crédit

## **Write-off Rate**

Write-off = Prêts Radiés / Encours Moyen × 100

Interprétation: Pertes définitives

## **2.3 Indicateurs de Liquidité**

### **Loan-to-Deposit Ratio (LDR)**

LDR = Total des Prêts / Total des Dépôts × 100

Benchmark: 80–90%

Interprétation: Utilisation des dépôts pour les prêts

### **Liquidity Coverage Ratio (LCR)**

LCR = Actifs Liquides de Haute Qualité / Sorties Nettes de Trésorerie 30 jours × 100

Exigence Bâle III: 100%

### **Net Stable Funding Ratio (NSFR)**

NSFR = Financement Stable Disponible / Financement Stable Requis × 100

Exigence: 100%

## **2.4 Indicateurs de Capital**

### **Capital Adequacy Ratio (CAR)**

CAR = Fonds Propres Réglementaires / Actifs Pondérés par les Risques × 100

Exigence BRH: 12%

Exigence Bâle III: 8%

### **Tier 1 Ratio**

Tier 1 = Capital Tier 1 / RWA × 100

Core capital: Capital de base (actions, réserves)

### **Leverage Ratio**

Leverage = Tier 1 / Total des Expositions × 100

Exigence: 3%

## **2.5 Indicateurs Commerciaux**

### **Customer Acquisition Cost (CAC)**

CAC = Coûts Marketing et Ventes / Nombre de Nouveaux Clients

## **Customer Lifetime Value (CLV/LTV)**

LTV =  $\text{Revenu Moyen} \times \text{Durée de Relation} \times \text{Marge}$

## **Attrition Rate (Churn)**

Churn =  $\text{Clients Perdus} / \text{Clients Début de Période} \times 100$

## **Cross-sell Ratio**

Cross-sell =  $\text{Nombre de Produits} / \text{Nombre de Clients}$

## **Net Promoter Score (NPS)**

NPS = % Promoteurs - % Détracteurs

Score: -100 à +100

---

## **Partie 3: Reporting Bancaire**

### **3.1 Types de Rapports**

Type	Fréquence	Audience	Contenu
<b>Exécutif</b>	Mensuel	Direction	KPIs stratégiques, tendances
<b>Opérationnel</b>	Quotidien/Hebdo	Managers	Métriques détaillées
<b>Réglementaire</b>	Variable	BRH, Auditeurs	Conformité, ratios
<b>Ad-hoc</b>	Sur demande	Analystes	Analyses spécifiques

### **3.2 Dashboard Exécutif Type**

TABLEAU DE BORD DIRECTION  
[Mois - Année]

ROE	ROA	NPL	CAR
14.2%	1.8%	4.1%	15.2%

[Évolution PNB - 12 derniers mois]

Répartition Prêts  
[Pie Chart]

Performance Agences  
[Bar Chart]

### 3.3 Reporting Réglementaire BRH

#### Rapports Obligatoires (Contexte Haïti)

- **Bilan et Compte de Résultat:** Mensuel
  - **Ratios Prudentiels:** CAR, Liquidité, Grands risques
  - **État des Provisions:** Classification du portefeuille
  - **Reporting AML:** Transactions suspectes
  - **État des Changes:** Position de change
- 

## Partie 4: Analyse par Domaine

### 4.1 Analyse du Portefeuille de Crédit

```
def credit_portfolio_analysis(df_loans):
    """Analyse complète du portefeuille de crédit"""

    analysis = {}

    # Répartition par statut
    analysis['status_distribution'] = df_loans.groupby('statut').agg({
        'montant': ['sum', 'count'],
        'solde_restant': 'sum'
    })

    # NPL par secteur
    npl_loans = df_loans[df_loans['jours_retard'] > 90]
    analysis['npl_by_sector'] = npl_loans.groupby('secteur')['solde_restant'].sum() / \
        df_loans.groupby('secteur')['solde_restant'].sum() * 100

    # Concentration des risques
    total_exposure = df_loans['solde_restant'].sum()
    top_10 = df_loans.nlargest(10, 'solde_restant')['solde_restant'].sum()
    analysis['concentration_top10'] = top_10 / total_exposure * 100

    # Vintage analysis (cohortes)
    df_loans['vintage'] = df_loans['date_octroi'].dt.to_period('M')
    analysis['vintage'] = df_loans.groupby('vintage').apply(
        lambda x: (x['jours_retard'] > 90).mean() * 100
    )

    return analysis
```

### 4.2 Analyse de la Clientèle

```
def customer_analysis(df_customers, df_transactions):
    """Analyse de la base clients"""

    analysis = {}

    # Segmentation par valeur
    customer_value = df_transactions.groupby('client_id').agg({
        'montant': ['sum', 'mean', 'count'],
        'solde_restant': ['sum', 'mean', 'count']
    })

    # Concentration des risques
    total_exposure = df_transactions['solde_restant'].sum()
    top_10 = df_transactions.nlargest(10, 'solde_restant')['solde_restant'].sum()
    analysis['concentration_top10'] = top_10 / total_exposure * 100

    # Vintage analysis (cohorte)
    df_transactions['vintage'] = df_transactions['date_octroi'].dt.to_period('M')
    analysis['vintage'] = df_transactions.groupby('vintage').apply(
        lambda x: (x['jours_retard'] > 90).mean() * 100
    )

    return analysis
```

```

        'date': ['min', 'max']
    })
customer_value.columns = ['total_tx', 'avg_tx', 'nb_tx', 'first_tx', 'last_tx']

# RFM Scoring
customer_value['recency'] = (pd.Timestamp.now() - customer_value['last_tx']).dt.days
customer_value['frequency'] = customer_value['nb_tx']
customer_value['monetary'] = customer_value['total_tx']

# Segmentation
analysis['segments'] = segment_customers(customer_value)

# Métriques clés
analysis['active_rate'] = (customer_value['recency'] < 90).mean() * 100
analysis['avg_products'] = df_customers['nb_produits'].mean()
analysis['churn_rate'] = calculate_churn(df_customers)

return analysis

```

### 4.3 Analyse de la Liquidité

```

def liquidity_analysis(df_balance_sheet):
    """Analyse de la position de liquidité"""

    analysis = {}

    # Loan-to-Deposit
    analysis['ldr'] = (df_balance_sheet['total_loans'] /
                       df_balance_sheet['total_deposits']) * 100

    # Structure des dépôts
    analysis['deposit_mix'] = {
        'demand': df_balance_sheet['demand_deposits'] / df_balance_sheet['total_deposits'],
        'savings': df_balance_sheet['savings_deposits'] / df_balance_sheet['total_deposits'],
        'term': df_balance_sheet['term_deposits'] / df_balance_sheet['total_deposits']
    }

    # Gap de liquidité
    analysis['liquidity_gap'] = calculate_maturity_gap(df_balance_sheet)

    # Coût des ressources
    analysis['cost_of_funds'] = (df_balance_sheet['interest_expense'] /
                                 df_balance_sheet['total_deposits']) * 100

    return analysis

```

---

## Partie 5: Segmentation Client Bancaire

### 5.1 Modèles de Segmentation

#### Par Valeur (RFM)

```

def rfm_segmentation(df, customer_id, date_col, amount_col):
    """Segmentation RFM"""

    # Calcul des métriques
    rfm = df.groupby(customer_id).agg({
        date_col: lambda x: (df[date_col].max() - x.max()).days,
        customer_id: 'count',
        amount_col: 'sum'
    })
    rfm.columns = ['Recency', 'Frequency', 'Monetary']

    # Scoring (quintiles)
    for col in ['Frequency', 'Monetary']:
        rfm[f'{col[0]}_Score'] = pd.qcut(rfm[col], 5, labels=[1,2,3,4,5])
    rfm['R_Score'] = pd.qcut(rfm['Recency'], 5, labels=[5,4,3,2,1]) # Inversé

    # Segment final
    rfm['RFM_Segment'] = rfm.apply(lambda x: f"{x['R_Score']}{x['F_Score']}{x['M_Score']}", axis=1)

    return rfm

```

## Par Comportement

- **Transactionnels:** Beaucoup de transactions, soldes bas
- **Épargnants:** Peu de transactions, soldes élevés
- **Emprunteurs:** Utilisateurs de crédit
- **Digitaux:** Préfèrent les canaux numériques
- **Traditionnels:** Préfèrent l'agence

## Par Cycle de Vie

- **Nouveaux:** < 1 an d'ancienneté
- **En croissance:** Augmentation des produits/activité
- **Matures:** Relation stable
- **À risque:** Diminution de l'activité
- **Inactifs:** Pas d'activité récente

## 5.2 Application Commerciale

```

def segment_action_plan(segment):
    """Plan d'action par segment"""

    actions = {
        'Champions': {
            'objectif': 'Fidéliser et récompenser',
            'actions': ['Programme VIP', 'Cross-sell premium', 'Parrainage']
        },
        'Fidèles': {
            'objectif': 'Maintenir et développer',
            'actions': ['Offres personnalisées', 'Augmentation limite', 'Nouveaux produits']
        },
        'Nouveaux': {
            'objectif': 'Activer et engager',
            'actions': ['Onboarding', 'Tutoriels', 'Offre de bienvenue']
        }
    }

```

```

        },
        'À risque': {
            'objectif': 'Réactiver',
            'actions': ['Contact proactif', 'Offre de rétention', 'Enquête satisfaction']
        },
        'Perdus': {
            'objectif': 'Reconquérir si rentable',
            'actions': ['Campagne win-back', 'Offre exceptionnelle', 'Feedback']
        }
    }

    return actions.get(segment, 'Segment non reconnu')

```

---

## Partie 6: Analyse des Risques

### 6.1 Risque de Crédit

```

def credit_risk_dashboard(df_loans):
    """Dashboard risque de crédit"""

    metrics = {}

    # Classification du portefeuille
    def classify_loan(days_overdue):
        if days_overdue == 0:
            return 'Performing'
        elif days_overdue <= 30:
            return 'Watch'
        elif days_overdue <= 90:
            return 'Substandard'
        elif days_overdue <= 180:
            return 'Doubtful'
        else:
            return 'Loss'

    df_loans['classification'] = df_loans['jours_retard'].apply(classify_loan)

    # Distribution par classe
    metrics['portfolio_classification'] = df_loans.groupby('classification')['solde_restant'].sum()

    # NPL ratio
    npl_amount = df_loans[df_loans['jours_retard'] > 90]['solde_restant'].sum()
    total_loans = df_loans['solde_restant'].sum()
    metrics['npl_ratio'] = npl_amount / total_loans * 100

    # Expected Loss
    metrics['expected_loss'] = df_loans.apply(
        lambda x: x['solde_restant'] * x['pd'] * x['lgd'], axis=1
    ).sum()

    return metrics

```

## 6.2 Risque de Liquidité

```
def liquidity_risk_analysis(df_cashflows):
    """Analyse du risque de liquidité"""

    # Gap par maturité
    maturity_buckets = ['<1M', '1-3M', '3-6M', '6-12M', '>12M']

    gap_analysis = df_cashflows.groupby('maturity_bucket').agg({
        'inflows': 'sum',
        'outflows': 'sum'
    })
    gap_analysis['net_gap'] = gap_analysis['inflows'] - gap_analysis['outflows']
    gap_analysis['cumulative_gap'] = gap_analysis['net_gap'].cumsum()

    return gap_analysis
```

## 6.3 Risque Opérationnel

```
def operational_risk_metrics(df_incidents):
    """Métriques de risque opérationnel"""

    metrics = {}

    # Fréquence des incidents
    metrics['incident_frequency'] = df_incidents.groupby('type').size()

    # Pertes par catégorie
    metrics['loss_by_category'] = df_incidents.groupby('categorie')['perte'].sum()

    # Tendance
    metrics['trend'] = df_incidents.set_index('date').resample('M')['perte'].sum()

    # Top incidents
    metrics['top_incidents'] = df_incidents.nlargest(10, 'perte')[['type', 'description', 'perte']]

    return metrics
```

---

## Partie 7: BI et Conformité Réglementaire

### 7.1 Reporting BRH

#### Ratios Prudentiels à Surveiller

```
def calculate_regulatory_ratios(balance_sheet, pnl):
    """Calcul des ratios réglementaires"""

    ratios = {}

    # Ratio de solvabilité (CAR)
    ratios['car'] = (balance_sheet['tier1'] + balance_sheet['tier2']) / \
                    balance_sheet['rwa'] * 100
    ratios['car_compliant'] = ratios['car'] >= 12 # Seuil BRH
```

```

# Ratio de liquidité
ratios['liquidity'] = balance_sheet['liquid_assets'] / \
                      balance_sheet['demand_deposits'] * 100
ratios['liquidity_compliant'] = ratios['liquidity'] >= 20 # Seuil BRH

# Ratio des grands risques
ratios['large_exposure'] = balance_sheet['largest_exposure'] / \
                           balance_sheet['equity'] * 100
ratios['large_exposure_compliant'] = ratios['large_exposure'] <= 15

return ratios

```

## 7.2 Anti-Money Laundering (AML)

```

def aml_monitoring(df_transactions):
    """Indicateurs AML"""

    alerts = {}

    # Transactions suspectes
    alerts['high_value'] = df_transactions[df_transactions['montant'] > 50000]

    # Structuration (smurfing)
    daily_totals = df_transactions.groupby(['client_id', 'date'])['montant'].sum()
    alerts['potential_structuring'] = daily_totals[
        (daily_totals > 40000) & (daily_totals < 50000)
    ]

    # Activité inhabituelle
    client_stats = df_transactions.groupby('client_id')['montant'].agg(['mean', 'std'])
    df_transactions = df_transactions.merge(client_stats, on='client_id')
    alerts['unusual_activity'] = df_transactions[
        df_transactions['montant'] > df_transactions['mean'] + 3 * df_transactions['std']
    ]

return alerts

```

---

## Partie 8: Contexte Haïtien Spécifique

### 8.1 Défis Spécifiques

- **Économie informelle:** Grande partie non bancarisée
- **Risque de change:** Volatilité HTG/USD
- **Infrastructure:** Connectivité variable
- **Inclusion financière:** Faible pénétration bancaire
- **Catastrophes naturelles:** Impact sur le portefeuille

### 8.2 Opportunités

- **Mobile banking:** Forte adoption du mobile
- **Transferts diaspora:** Flux importants

- **Microfinance:** Marché en croissance
- **PME:** Secteur sous-servi

### 8.3 KPIs Adaptés au Contexte

```
haiti_specific_kpis = {
    'digital_adoption': 'Utilisateurs mobile / Total clients',
    'diaspora_remittance': 'Volume transferts / Total dépôts',
    'currency_mix': 'Dépôts HTG / Total dépôts',
    'geographic_coverage': 'Clients hors PAP / Total clients',
    'msme_portfolio': 'Prêts PME / Total prêts'
}
```

---

### Questions d'Entretien

- Quels sont les KPIs les plus importants pour une banque commerciale?** → ROE, ROA, NPL ratio, CAR, NIM, CIR
  - Comment calculer le NPL ratio?** → Prêts > 90 jours impayés / Total des prêts × 100
  - Qu'est-ce que le ratio Loan-to-Deposit et pourquoi est-il important?** → Prêts/Dépôts; mesure l'utilisation des dépôts, idéalement 80-90%
  - Comment segmenter les clients d'une banque?** → Par valeur (RFM), comportement, cycle de vie, démographie
  - Quels sont les principaux risques bancaires à surveiller?** → Crédit, liquidité, marché, opérationnel, conformité
- 

### Checklist BI Bancaire

Maîtriser les KPIs de rentabilité (ROE, ROA, NIM)  
 Comprendre les indicateurs de qualité d'actifs (NPL, provisions)  
 Connaître les ratios réglementaires (CAR, liquidité)  
 Savoir segmenter les clients  
 Comprendre le reporting réglementaire  
 Maîtriser l'analyse du risque de crédit  
 Connaître les spécificités du marché haïtien

---

**Rappel:** La BI bancaire doit toujours équilibrer trois objectifs: rentabilité, gestion des risques et conformité réglementaire.