

**Transformation Data-Driven**



# **De la location DVD au Streaming**

**Fonctionnalité: Analytics et Métriques de Performance**

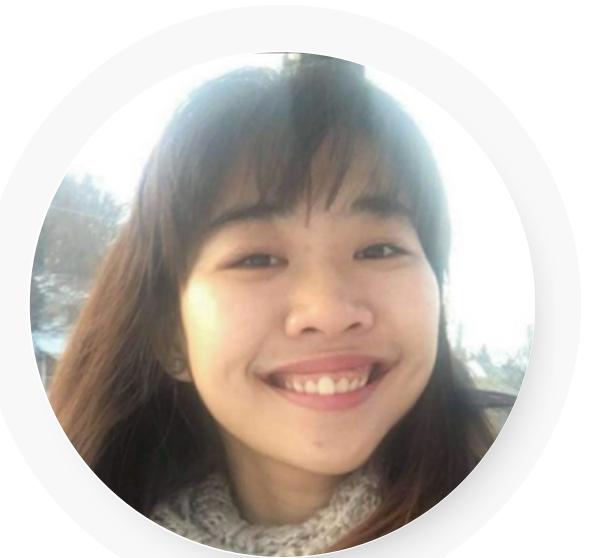
# Task Force



Thomas  
MARIE-ANNE



Irmeline  
TEGOMO GUEVOU



Thuy-Linh  
TO



Jeff  
SAMEDY



Raouf  
OROU-GOURA



Kodzo  
LIMA

**NETFLIX**



**1ère  
PARTIE**

# Contexte & enjeux

## Transition numérique

Passage d'un modèle transactionnel physique (DVD) à un flux continu numérique (Streaming)

## Problématique

La base de données (DVD) est opérationnelle mais ne permet aucune analyse stratégique

## Objectif

Déployer une couche de Business Intelligence pour piloter la croissance et la rétention client

## Périmètre

Analyse focalisée sur la base fournie

## Livrable

- Requêtes SQL pour analyser les données brutes
- Tableau de bord PowerBI pour afficher nos KPIs

## Les 4 piliers de la solution



### Performance du contenus

- Fonction : Tableau de bord des top visionnages
- Impact Métier : Optimisation du catalogue et ROI des acquisitions de films



### Comportement

- Fonction : analyse des pics de visionnage
- Impact métier : dimensionnement infrastructurel et ciblage marketing horaire



### Rétention client

- Fonction : détection du risque de Churn
- Impact métier : Sécurisation des revenus récurrents et fidélisation proactive



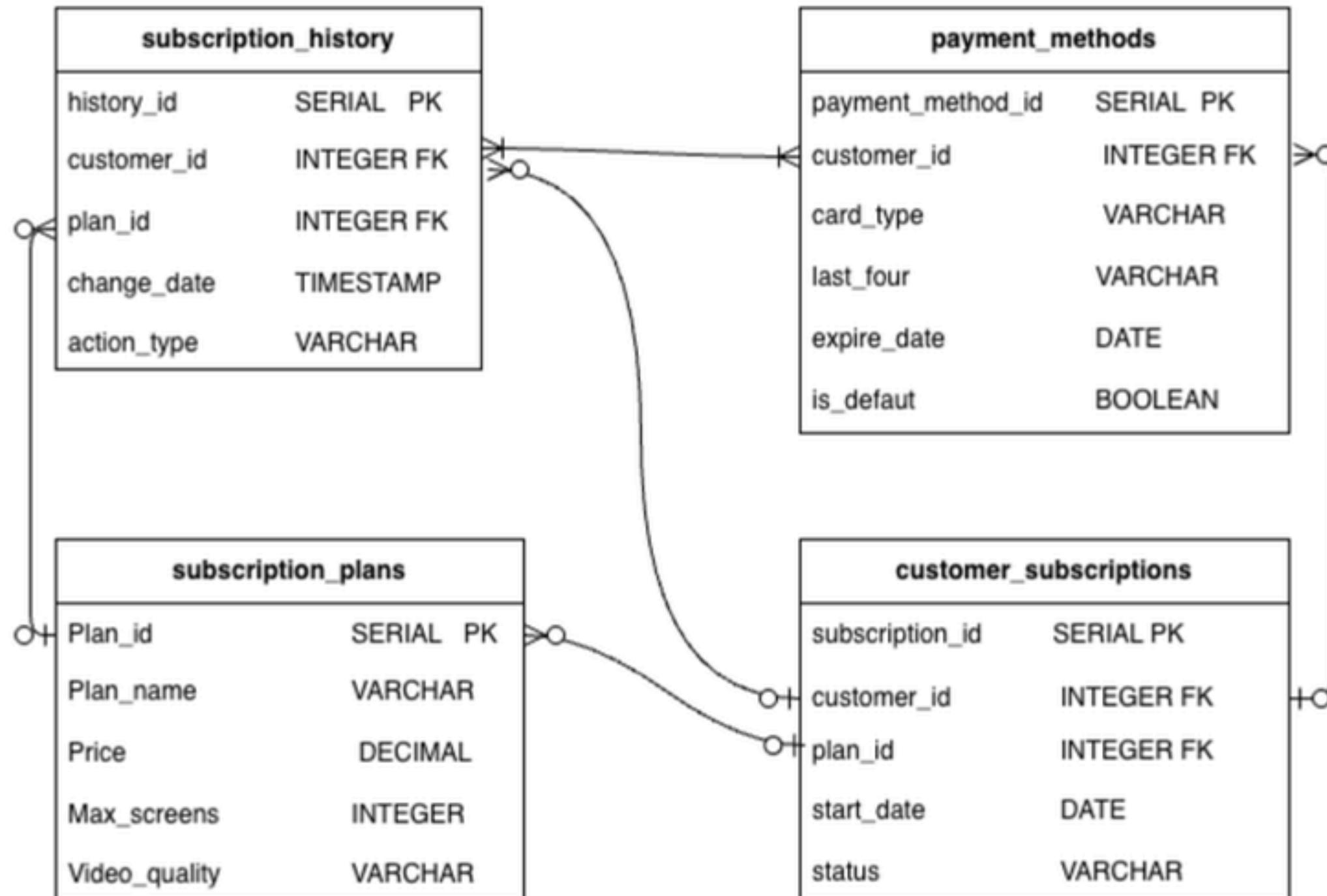
### Monétisation

- Fonction : Monitoring du ARPU (Revenu Moyen par Utilisateur)
- Impact métier : Pilotage de la rentabilité par segment d'abonnement

# Analytics et Métriques de Performance

# Modélisation des nouvelles tables

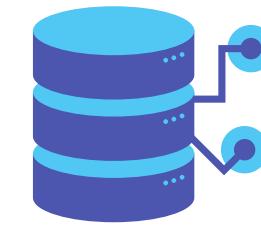
## Diagramme ERD



**NETFLIX**



**2ème  
PARTIE**



## CRÉATION DES TABLES

# TABLE FORMULES D'ABONNEMENT

## REQUÊTE SQL

```
CREATE TABLE subscription_plans (
    plan_id SERIAL PRIMARY KEY,
    plan_name VARCHAR(50) NOT NULL,
    monthly_fee NUMERIC(5,2) NOT NULL,
    description TEXT
);
```

	plan_id [PK] integer	plan_name character varying (50)	monthly_fee numeric (5,2)	description text
1	1	Découverte	9.90	Basé sur le tarif film 0.99€
2	2	Standard	19.90	Basé sur le tarif film 2.99€
3	3	Premium	29.90	Basé sur le tarif film 4.99€

## TABLE ABOUNNEMENTS CLIENTS

## REQUÊTE SQL

```
CREATE TABLE customer_subscriptions (
    subscription_id SERIAL PRIMARY KEY,
    customer_id INTEGER NOT NULL,
    plan_id INTEGER NOT NULL,
    start_date DATE NOT NULL,
    status VARCHAR(20) DEFAULT 'Active',
    CONSTRAINT fk_sub_customer FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
    CONSTRAINT fk_sub_plan FOREIGN KEY (plan_id) REFERENCES subscription_plans(plan_id)
);
```

	subscription_id [PK] integer	customer_id integer	plan_id integer	start_date date	status character varying (20)
1	1	87	1	2005-05-27	Active
2	2	184	1	2005-05-26	Active
3	3	477	1	2005-05-30	Active
4	4	273	2	2005-05-25	Active
5	5	550	2	2005-05-30	Active
6	6	394	2	2005-05-26	Active
7	7	51	1	2005-05-25	Active
8	8	272	1	2005-05-25	Active
9	9	70	2	2005-05-31	Active
10	10	190	2	2005-05-27	Active

## TABLE MÉTHODES DE PAIEMENT

## REQUÊTE SQL

```
CREATE TABLE payment_methods (
    payment_method_id SERIAL PRIMARY KEY,
    customer_id INTEGER NOT NULL,
    card_type VARCHAR(20),
    last_four VARCHAR(4),
    expiry_date DATE,
    is_default BOOLEAN DEFAULT TRUE,
    CONSTRAINT fk_payment_customer FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);
```

	payment_method_id [PK] integer	customer_id integer	card_type character varying (20)	last_four character varying (4)	expiry_date date	is_default boolean
1	1	184	Active Card	1095	2027-12-31	true
2	2	87	Active Card	2043	2027-12-31	true
3	3	477	Active Card	7534	2027-12-31	true
4	4	273	Active Card	5222	2027-12-31	true
5	5	550	Active Card	1993	2027-12-31	true
6	6	51	Active Card	9578	2027-12-31	true
7	7	394	Active Card	1952	2027-12-31	true
8	8	272	Active Card	5207	2027-12-31	true
9	9	70	Active Card	9767	2027-12-31	true
10	10	190	Active Card	2072	2027-12-31	true

# TABLE HISTORIQUE DES CHANGEMENTS D'ABONNEMENT

REQUÊTE SQL

```
CREATE TABLE subscription_history (
    history_id SERIAL PRIMARY KEY,
    subscription_id INTEGER NOT NULL,
    change_type VARCHAR(50),
    change_date TIMESTAMP WITHOUT TIME ZONE NOT NULL,
    notes TEXT,
    CONSTRAINT fk_history_subscription FOREIGN KEY (subscription_id) REFERENCES customer_subscriptions(subscription_id)
);
```

The screenshot shows a database interface with a toolbar at the top containing icons for Data Output, Messages, Notifications, and a SQL editor tab. Below the toolbar is a table with the following data:

	history_id [PK] integer	subscription_id integer	change_type character varying (50)	change_date timestamp without time zone	notes text
1	1	150	RENEWAL	2007-02-15 22:25:46.996577	Paiement réel de 7.99€
2	2	150	RENEWAL	2007-02-16 17:23:14.996577	Paiement réel de 1.99€
3	3	150	RENEWAL	2007-02-16 22:41:45.996577	Paiement réel de 7.99€
4	4	150	RENEWAL	2007-02-19 19:39:56.996577	Paiement réel de 2.99€
5	5	150	RENEWAL	2007-02-20 17:31:48.996577	Paiement réel de 7.99€
6	6	150	RENEWAL	2007-02-21 12:33:49.996577	Paiement réel de 5.99€
7	7	256	RENEWAL	2007-02-17 23:58:17.996577	Paiement réel de 5.99€
8	8	256	RENEWAL	2007-02-20 02:11:44.996577	Paiement réel de 5.99€
9	9	256	RENEWAL	2007-02-20 13:57:39.996577	Paiement réel de 2.99€
10	10	402	RENEWAL	2007-02-16 00:10:50.996577	Paiement réel de 4.99€
11	11	402	RENEWAL	2007-02-16 01:15:33.996577	Paiement réel de 6.99€
12	12	402	RENEWAL	2007-02-17 01:26:00.996577	Paiement réel de 0.99€

NETFLIX



# REQUÊTES SQL

# 1- Les films les plus regardés

## DESCRIPTION

- **Objectif** : comprendre les goûts passés de nos clients pour mieux cibler l'offre de lancement du Streaming.
- **Analyse de la popularité historique (DVD)**: cette requête identifie les blockbusters de notre catalogue physique en comptabilisant le nombre total de locations pour chaque film.

## REQUÊTE SQL

```
SELECT
    f.film_id,
    f.title,
    COUNT(r.rental_id) AS total_views
FROM rental r
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN film f ON i.film_id = f.film_id
GROUP BY f.film_id, f.title
ORDER BY total_views DESC;
```

## RESULTATS

	film_id [PK] integer	title character varying (255)	total_views bigint
1	103	Bucket Brotherhood	34
2	738	Rocketeer Mother	33
3	382	Grit Clockwork	32
4	331	Forward Temple	32
5	730	Ridgemont Submarine	32
6	489	Juggler Hardly	32
7	767	Scalawag Duck	32
8	1000	Zorro Ark	31
9	621	Network Peak	31
10	369	Goodfellas Salute	31

# 2 - Statistiques de visionnage par période

## DESCRIPTION

- **Objectif** : utiliser ces tendances historiques pour dimensionner l'infrastructure de nos futurs serveurs et anticiper les périodes de fort trafic.
- **Analyse de la saisonnalité** : cette requête agrège l'historique des locations mois par mois pour identifier les pics d'activité récurrents (vacances, fêtes).

## REQUÊTE SQL

```
SELECT  
DATE_TRUNC('month', rental_date) AS period,  
COUNT(*) AS total_views  
FROM rental  
GROUP BY period  
ORDER BY period;
```

## RESULTATS

	period timestamp without time zone	total_views bigint
1	2005-05-01 00:00:00	1156
2	2005-06-01 00:00:00	2311
3	2005-07-01 00:00:00	6709
4	2005-08-01 00:00:00	5686
5	2006-02-01 00:00:00	182

# 3- Évolution des clients actifs

## DESCRIPTION

- **Objectif** : surveiller la santé de la base d'abonnés pour détecter les baisses d'engagement avant qu'elles ne se transforment en désabonnements (Churn).
- **Suivi des Utilisateurs Actifs** : cette requête calcule le nombre de clients uniques ayant consommé du contenu chaque mois, permettant de distinguer les utilisateurs actifs des comptes inscrits mais "dormants".

## REQUÊTE SQL

```
CREATE OR REPLACE VIEW vw_active_customers_month AS
SELECT
    DATE_TRUNC('month', rental_date) AS period,
    COUNT(DISTINCT customer_id) AS active_customers
FROM rental
GROUP BY period;

SELECT *
FROM vw_active_customers_month
ORDER BY period;
```

## RESULTATS

	period timestamp without time zone	active_customers bigint
1	2005-05-01 00:00:00	520
2	2005-06-01 00:00:00	590
3	2005-07-01 00:00:00	599
4	2005-08-01 00:00:00	599
5	2005-09-01 00:00:00	0
6	2005-10-01 00:00:00	0
7	2005-11-01 00:00:00	0
8	2005-12-01 00:00:00	0
9	2006-01-01 00:00:00	0
10	2006-02-01 00:00:00	158

# 4 - Évolution du Chiffre d'Affaires mensuel

## DESCRIPTION

- **Objectif** : monitorer la santé financière de l'entreprise en identifiant les tendances de croissance du chiffre d'affaires.
- **Analyse du Chiffre d'Affaires** : calcul du revenu total généré par période pour détecter la saisonnalité des ventes et valider la stabilité économique avant le pivot stratégique.

## REQUÊTE SQL

```
SELECT  
DATE_TRUNC('month', payment_date) AS period,  
SUM(amount) AS revenue  
FROM payment  
GROUP BY period  
ORDER BY period;  
;
```

## RESULTATS

	period	revenue
1	2007-02-01 00:00:00	8351.84
2	2007-03-01 00:00:00	23886.56
3	2007-04-01 00:00:00	28559.46
4	2007-05-01 00:00:00	514.18

# 5- Analyse de la valeur client

## DESCRIPTION

- **Objectif** : savoir combien rapporte un client moyen et identifier les plus fidèles.
- **Analyse de la valeur client** : le revenu moyen généré par un utilisateur et le classement des clients ayant le plus dépensé, cibles prioritaires pour nos programmes de rétention.

## REQUÊTE SQL

```
SELECT
    ROUND(
        SUM(amount) / COUNT(DISTINCT customer_id),
        2
    ) AS arpu
FROM payment;
```

## RESULTATS

	arpu
	numeric
1	102.36

## Top clients (revenu généré)

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    SUM(p.amount) AS total_spent
FROM payment p
JOIN customer c ON p.customer_id = c.customer_id
GROUP BY c.customer_id, c.first_name, c.last_name
ORDER BY total_spent DESC;
```

	customer_id [PK] integer	first_name character varying (45)	last_name character varying (45)	total_spent numeric
1	148	Eleanor	Hunt	211.55
2	526	Karl	Seal	208.58
3	178	Marion	Snyder	194.61
4	137	Rhonda	Kennedy	191.62
5	144	Clara	Shaw	189.60
6	459	Tommy	Collazo	183.63
7	181	Ana	Bradley	167.67
8	410	Curtis	Irby	167.62
9	236	Marcia	Dean	166.61
10	403	Mike	Way	162.67

# 6 – Revenu généré par catégorie de film

## DESCRIPTION

- **Objectif** : Affiner les algorithmes de recommandation en identifiant les thématiques qui déclenchent le plus souvent l'acte d'achat.
- **Préférences thématiques** : classement décroissant des catégories selon le volume de revenus générés. Cela permet de définir les "genres" à mettre en avant sur la page d'accueil de la plateforme de streaming.

## REQUÊTE SQL

```
SELECT c.name, SUM(p.amount) AS total_sales
FROM category c
JOIN film_category fc ON c.category_id = fc.category_id
JOIN inventory i ON fc.film_id = i.film_id
JOIN rental r ON i.inventory_id = r.inventory_id
JOIN payment p ON r.rental_id = p.rental_id
GROUP BY c.name
ORDER BY total_sales DESC;
```

## RESULTATS

	<b>name</b> character varying (25)	<b>total_sales</b> numeric
1	Sports	4892.19
2	Sci-Fi	4336.01
3	Animation	4245.31
4	Drama	4118.46
5	Comedy	4002.48
6	New	3966.38
7	Action	3951.84
8	Foreign	3934.47
9	Games	3922.18
10	Family	3830.15

# NETFLIX 7 – Répartition géographique du portefeuille actif

## DESCRIPTION

- **Objectif** : cartographie de la densité client pour évaluer la performance commerciale par territoire.
- **Analyse de pénétration locale** : ce classement géographique révèle les marchés locaux les plus forts. Il permet de confirmer l'adéquation entre le catalogue proposé et la demande réelle de chaque territoire.

## REQUÊTE SQL

```
SELECT
    ci.city,
    COUNT(cs.customer_id) AS nombre_clients
FROM customer_subscriptions cs
JOIN customer c ON cs.customer_id = c.customer_id
JOIN address a ON c.address_id = a.address_id
JOIN city ci ON a.city_id = ci.city_id
WHERE cs.status = 'Active' -- Changé de Inactive à Active
pour voir tes données réelles
GROUP BY ci.city
ORDER BY nombre_clients DESC;
```

## RESULTATS

	city character varying (50)	nombre_clients bigint
1	London	2
2	Aurora	2
3	Tokat	1
4	Atlixco	1
5	Mukateve	1
6	Gatineau	1
7	Pontianak	1
8	Saint-Denis	1
9	Molodetno	1
10	Yingkou	1

# 8 - abonnements et moyens de paiement

## DESCRIPTION

- **Objectif**: diminuer les désabonnements causés par des échecs de paiement
- **Sécurisation des revenus**: lier le forfait au type de carte permet de repérer les moyens de paiement fragiles (ex: cartes à expiration proche). Cette analyse sert à anticiper les rejets de transaction pour garantir la continuité des abonnements.

## REQUÊTE SQL

```
SELECT
    c.first_name,
    c.last_name,
    sp.plan_name,
    pm.card_type,
    pm.last_four
FROM customer c
JOIN customer_subscriptions cs ON c.customer_id =
cs.customer_id
JOIN subscription_plans sp ON cs.plan_id = sp.plan_id
JOIN payment_methods pm ON c.customer_id =
pm.customer_id;
```

## RESULTATS

	first_name character varying (45) 	last_name character varying (45) 	plan_name character varying (50) 	card_type character varying (20) 	last_four character varying (4) 
1	Vivian	Ruiz	Découverte	Active Card	1095
2	Wanda	Patterson	Découverte	Active Card	2043
3	Dan	Paine	Découverte	Active Card	7534
4	Priscilla	Lowe	Standard	Active Card	5222
5	Guy	Brownlee	Standard	Active Card	1993
6	Alice	Stewart	Découverte	Active Card	9578
7	Chris	Brothers	Standard	Active Card	1952
8	Kay	Caldwell	Découverte	Active Card	5207
9	Christina	Ramirez	Standard	Active Card	9767
10	Yolanda	Weaver	Standard	Active Card	2072

# 9 – Diversité du catalogue

## DESCRIPTION

- **Objectif** : vérifier si notre catalogue est assez riche pour se lancer dans de nouveaux pays.
- **Audit linguistique** : On compte combien de films nous avons pour chaque langue. Cela permet de voir immédiatement nos lacunes, nous indiquant s'il faut en acheter ou en produire.

## REQUÊTE SQL

```
SELECT
    l.name AS langue,
    COUNT(f.film_id) AS nb_films
FROM language l
LEFT JOIN film f ON l.language_id = f.language_id -- LEFT
JOIN pour voir même les langues sans films
GROUP BY l.name
ORDER BY nb_films DESC;
```

## RESULTATS

	langue character (20)	nb_films bigint
1	English	1000
2	French	0
3	Mandarin	0
4	German	0
5	Japanese	0
6	Italian	0

# NETFLIX 10 - Classement des formules les plus vendues

## DESCRIPTION

- **Objectif** : vérifier si nos offres correspondent bien aux attentes du marché.
- **Analyse des ventes** : ce classement nous montre quel forfait est le "best-seller". Cela peut indiquer si nos offres sont perçues comme trop chères ou pas assez attractives.

## REQUÊTE SQL

```
SELECT sp.plan_name, COUNT(cs.subscription_id) AS  
nb_abonnes  
FROM subscription_plans sp  
JOIN customer_subscriptions cs ON sp.plan_id =  
cs.plan_id  
GROUP BY sp.plan_name  
ORDER BY nb_abonnes DESC  
LIMIT 5;
```

## RESULTATS

	plan_name character varying (50)	nb_abonnes bigint	revenu_mensuel_estime numeric
1	Standard	273	5432.70
2	Découverte	326	3227.40

# VALEUR AJOUTÉE & ROI

1



## Optimisation du churn management

- **Avant** : réaction après le départ du client
- **Maintenant** : identification proactive des clients "à risque"
- **Gain** : réduction potentielle du taux d'attrition

2



## Rationalisation des coûts de contenu

- **Avant** : achat de films basé sur l'intuition
- **Maintenant** : investissement ciblé sur les genres et formats qui performent réellement
- **Gain** : maximisation du ROI investi dans les droits de diffusion

3



## Pilotage financier approfondi

- **Avant** : vision comptable globale mensuelle
- **Maintenant** : calcul précis de la "Lifetime Value" (LTV) par abonné.
- **Gain** : capacité d'ajuster le pricing des abonnements en fonction de la consommation réelle.

## 1 - Data Visualization

- Connecter notre Data Warehouse SQL à des outils de visualisation.
- Objectif : Offrir des tableaux de bord dynamiques à la direction sans nécessiter de requêtes manuelles.

## 2- IA & Analyse Prédictive

- Utiliser l'historique de nos tables pour entraîner des modèles de Machine Learning.
- Objectif : Passer de l'analyse descriptive à l'analyse prédictive.

## 3- Optimisation & Scalabilité

- Optimisation majeure du stockage des données historiques.
- Objectif : Assurer que l'application reste rapide et ne ralentisse jamais, peu importe le nombre d'abonnés connectés

# ROADMAP & PERSPECTIVES

Vers une architecture Big Data et prédictive

NETFLIX



Power BI

# DEMONSTRATION

# Dashboard Power BI – Analyse des ventes (DVD Rental / Sakila)

\$61.31K

Total ventes

\$30.66K

Moyenne ventes

\$3.83K

Moy ventes catégories

\$61.31K

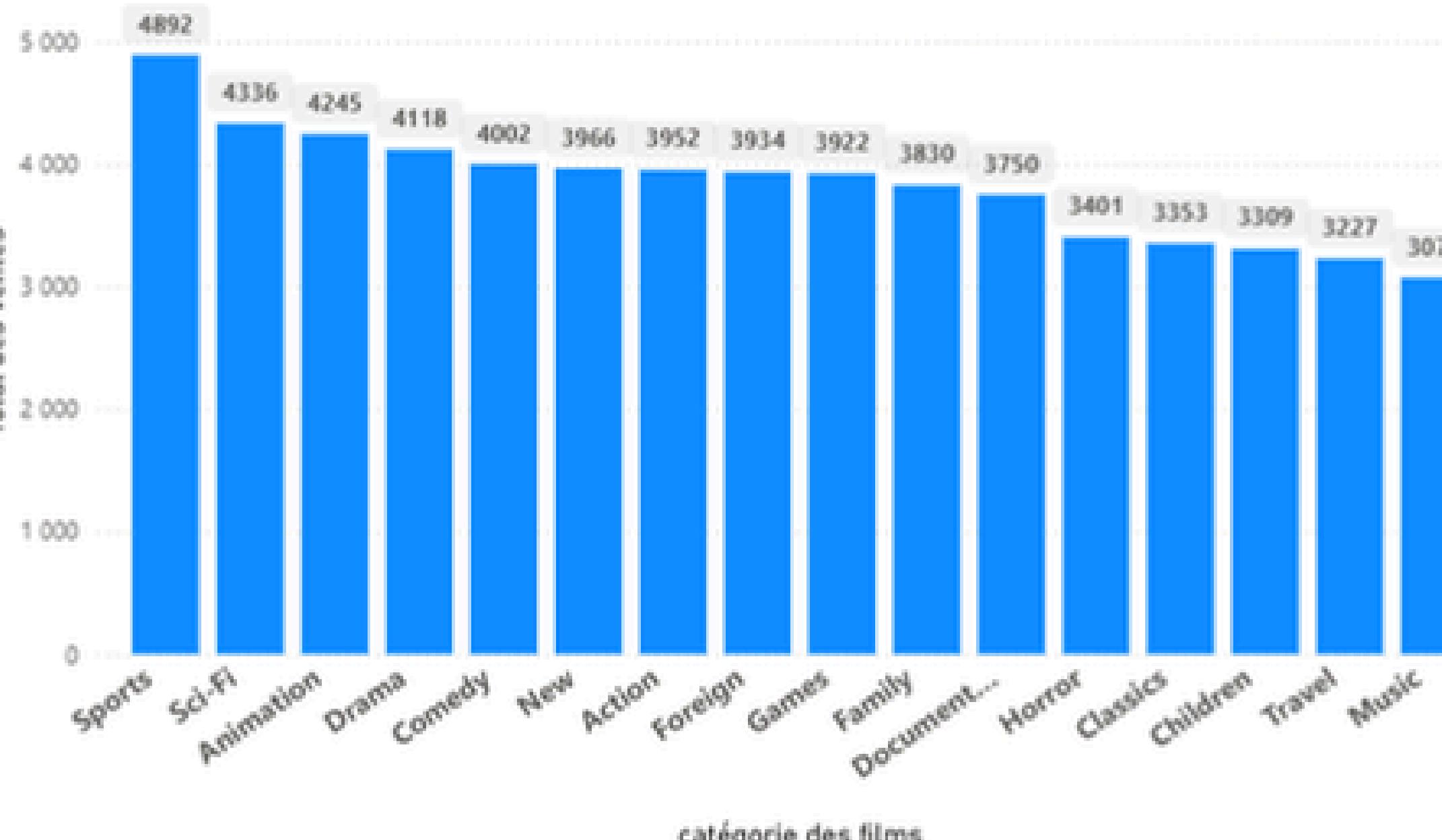
Total des revenus

16.044K

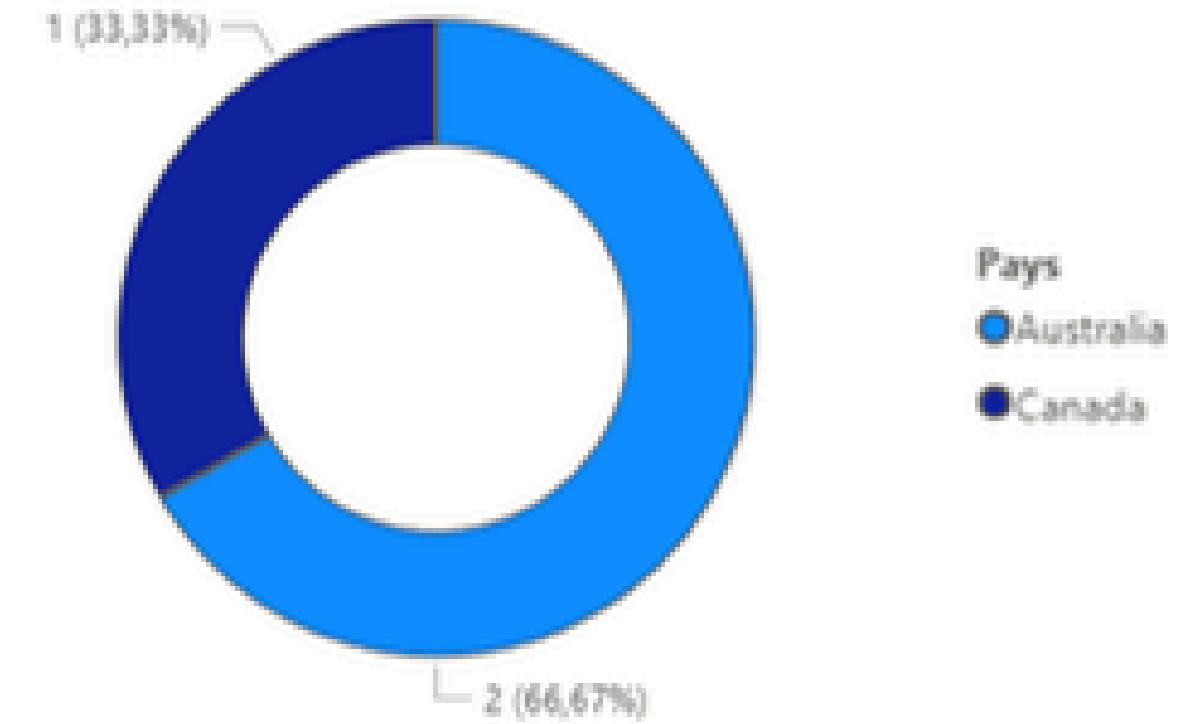
Total du nb de locations



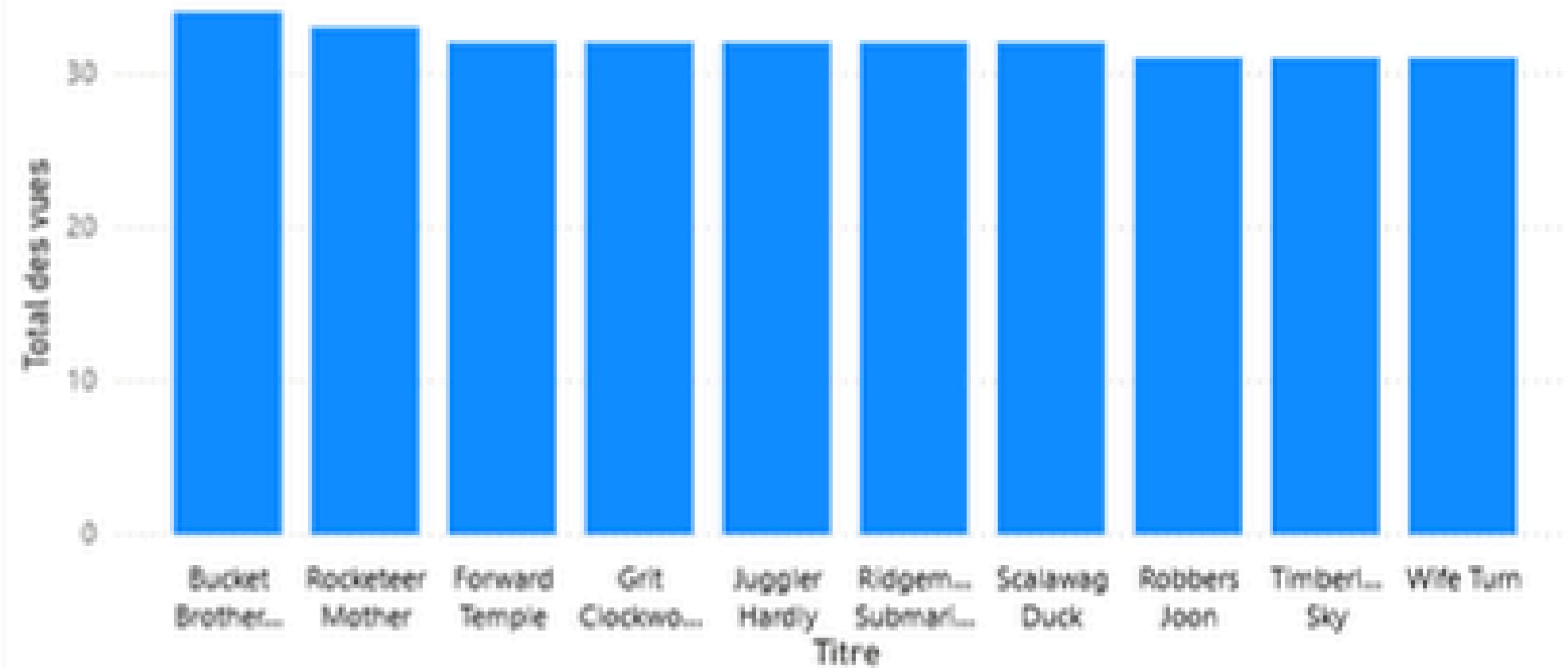
Total des ventes par catégorie des films



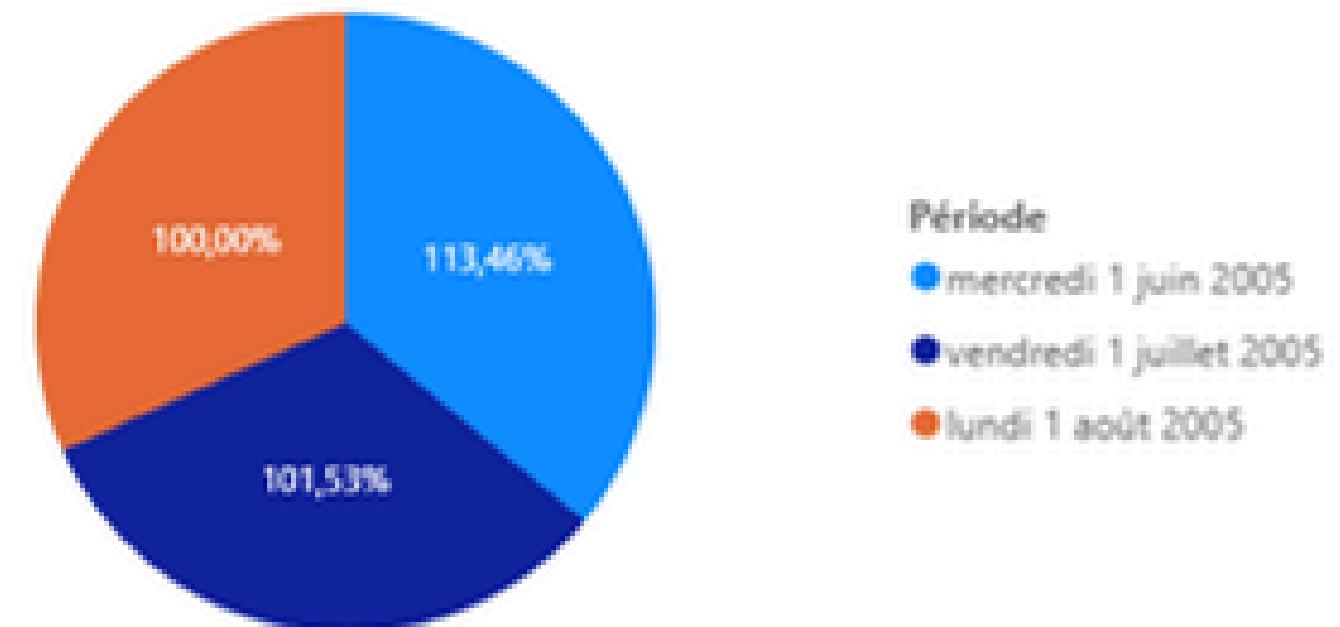
Répartition par pays du staff



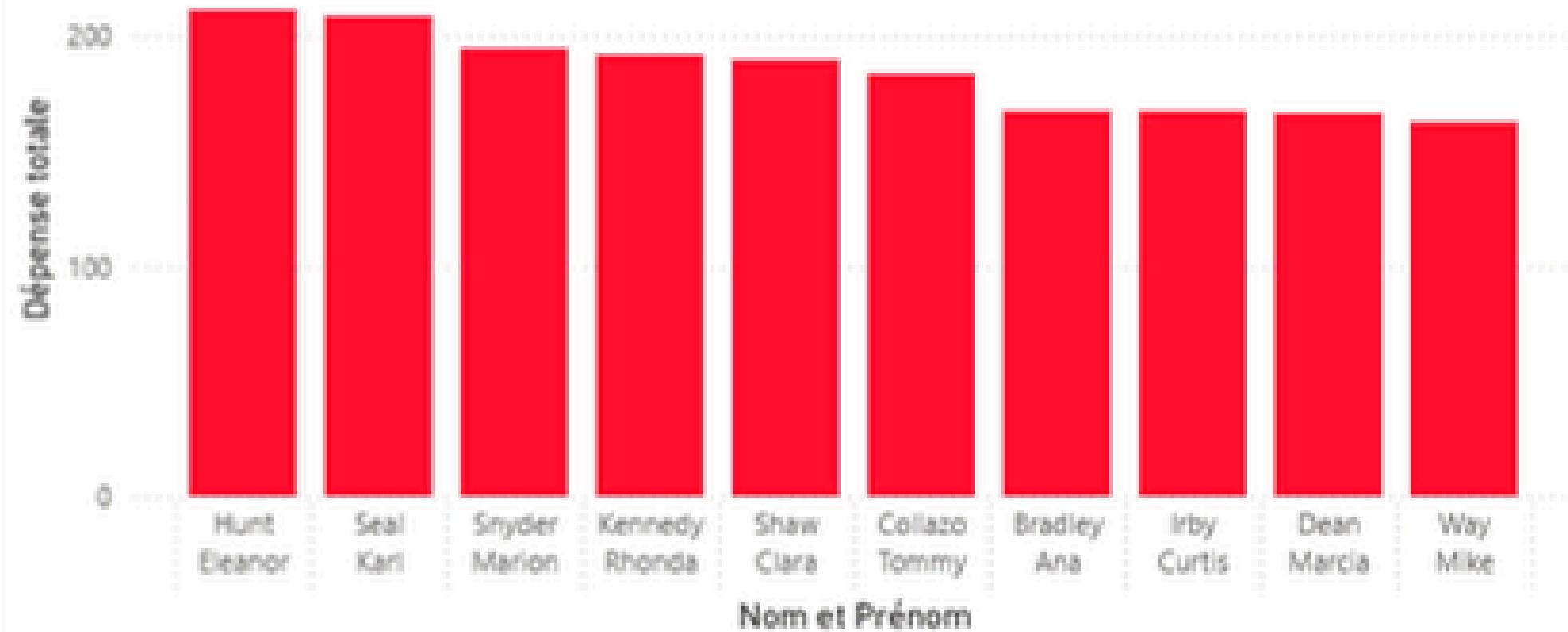
### Les 10 films les plus regardés



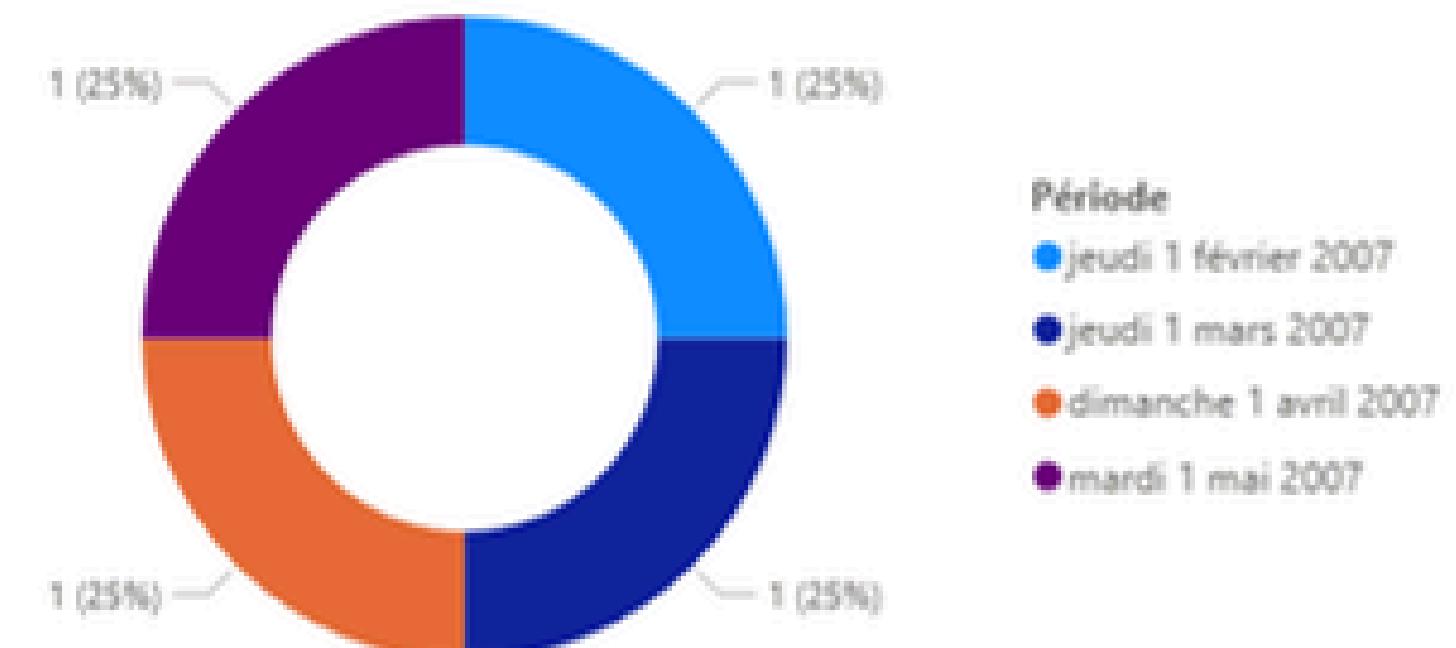
### Taux de croissance des clients par périodes



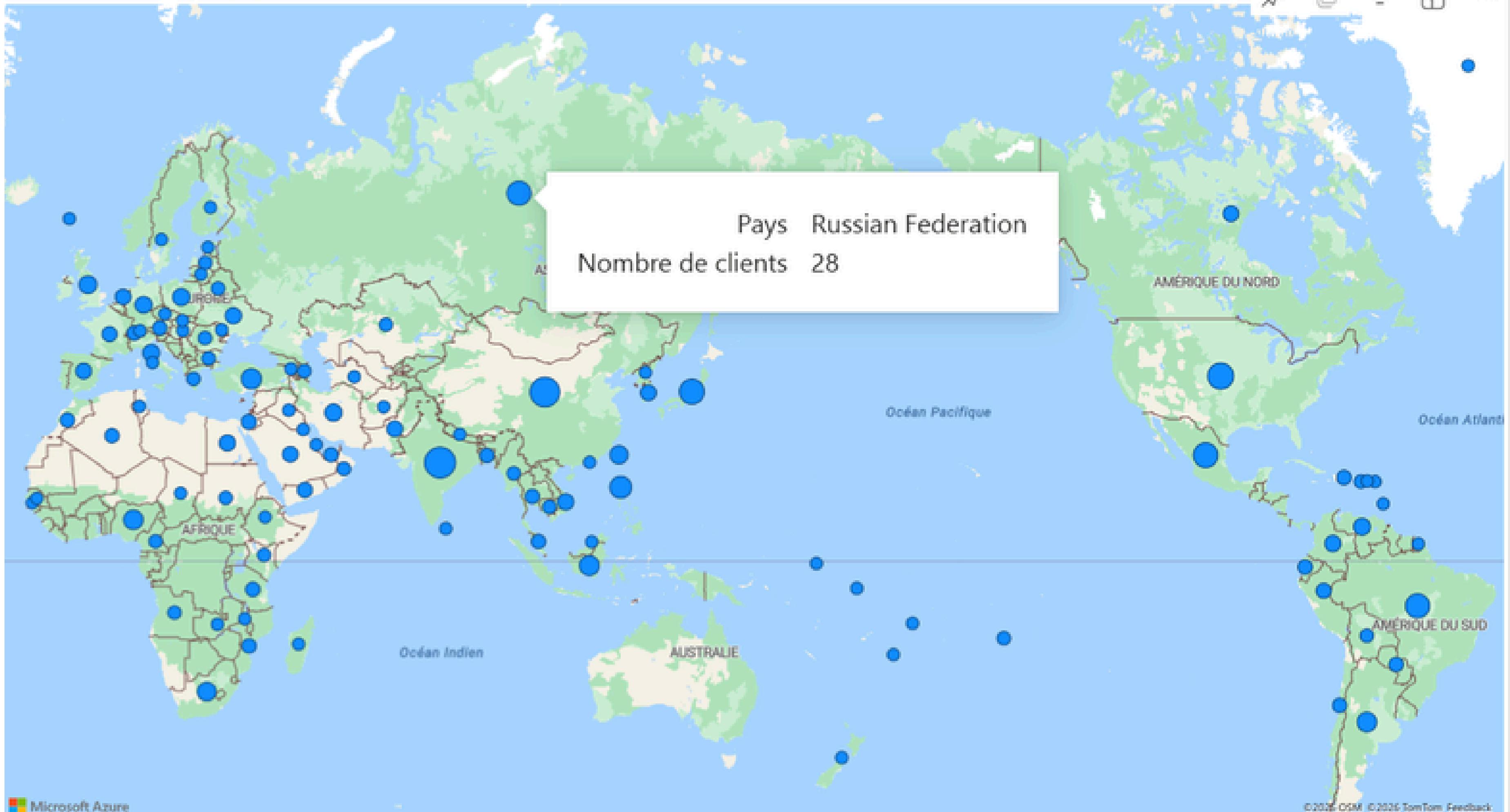
### Les 10 meilleurs clients



### Revenue par Période



## Nombre de clients par pays





**Transformation Data-Driven**



**Merci de votre attention !**