Projet 12 – POC Avantages Sportifs

Support de présentation – Choix techniques & architecture

---

1. Contexte & objectifs du projet

Contexte métier

Le projet s'inscrit dans le POC "Avantages Sportifs" pour l'entreprise fictive *Sport Data Solution*.
L'objectif est de tester la faisabilité technique d'un système permettant :

- de collecter et centraliser des données RH et sportives,
- de calculer des avantages financiers (prime sportive, journées bien-être),
- de contrôler la cohérence des déclarations salariés (trajets domicile-travail),
- de diffuser des messages en temps quasi-réel sur Slack.

Ces objectifs sont définis dans la note de cadrage Note+de+cadrage+_+POC+Avantages….

Contraintes principales

- Données RH sensibles → sécurité & intégrité
- Pipeline automatisé, traçable et monitoré
- Tests de qualité de données obligatoires
- Restitution Power BI
- Liberté de choix technologiques

---

2. Vue d'ensemble de l'architecture

Principe général

L'architecture repose sur :

- une base PostgreSQL managée,
- une séparation en couches (RAW / STG / OPS),
- un pipeline batch orchestré,
- un pipeline CDC temps réel pour Strava,
- des outils cloud managés pour le monitoring.

📌 Cette approche est alignée avec les bonnes pratiques présentées dans *Gérez un projet d'infrastructure* Gérez un projet d_infrastructur….

---

3. Choix de la base de données

PostgreSQL sur Aiven Cloud

Pourquoi ce choix :

- Base 100 % managée (sauvegardes, patchs, haute dispo)
- Support natif :
  - SQL avancé
  - contraintes d'intégrité
  - types avancés (`text[]` , `numeric` , `timestamptz`)
- Compatible avec :
  - outils BI (Power BI),
  - CDC (Redpanda / Kafka-like),
  - Soda Cloud.

👉 Ce choix répond directement aux contraintes de robustesse, sécurité et scalabilité.

---

4. Modélisation des données – approche en couches

Principe RAW / STG / OPS

Couche	Rôle
RAW	Données brutes, historisées, proches des sources
STG	Données nettoyées, normalisées, prêtes métier
OPS	Tables métier finales, KPI, contrôles

Cette séparation permet :

- traçabilité complète,
- reprocessing possible,
- contrôle qualité progressif.

4.1 Couche RAW – ingestion

Sources :

- Fichiers RH (`Données RH.xlsx`)
- Déclarations sportives (`Données Sportive.xlsx`)
- Simulation Strava (`strava_simulation_2025_weekly_max.csv`)

Chargement :

- Script Python `load_raw_tables.py` load_raw_tables
- UPSERT par clé métier (`id_salarie` , `id`)

Choix clés :

- Colonnes `source_file` et `ingestion_timestamp`
- Contraintes d'unicité
- FK logiques entre tables

📌 Objectif : fidélité maximale à la source, sans transformation métier.

4.2 Couche STG – normalisation

Scripts principaux :

- `stg_tables.py` stg tables
- `employee_sport_bonus.py` employee_sport_bonus

Transformations réalisées :

- Parsing d'adresses (rue, CP, ville)
- Normalisation des dates
- Conversion des moyens de transport en `text[]`
- Calcul de la prime sportive (5 % du salaire)

📌 Ici, la logique métier commence, mais sans agrégation finale.

4.3 Couche OPS – logique métier & KPI

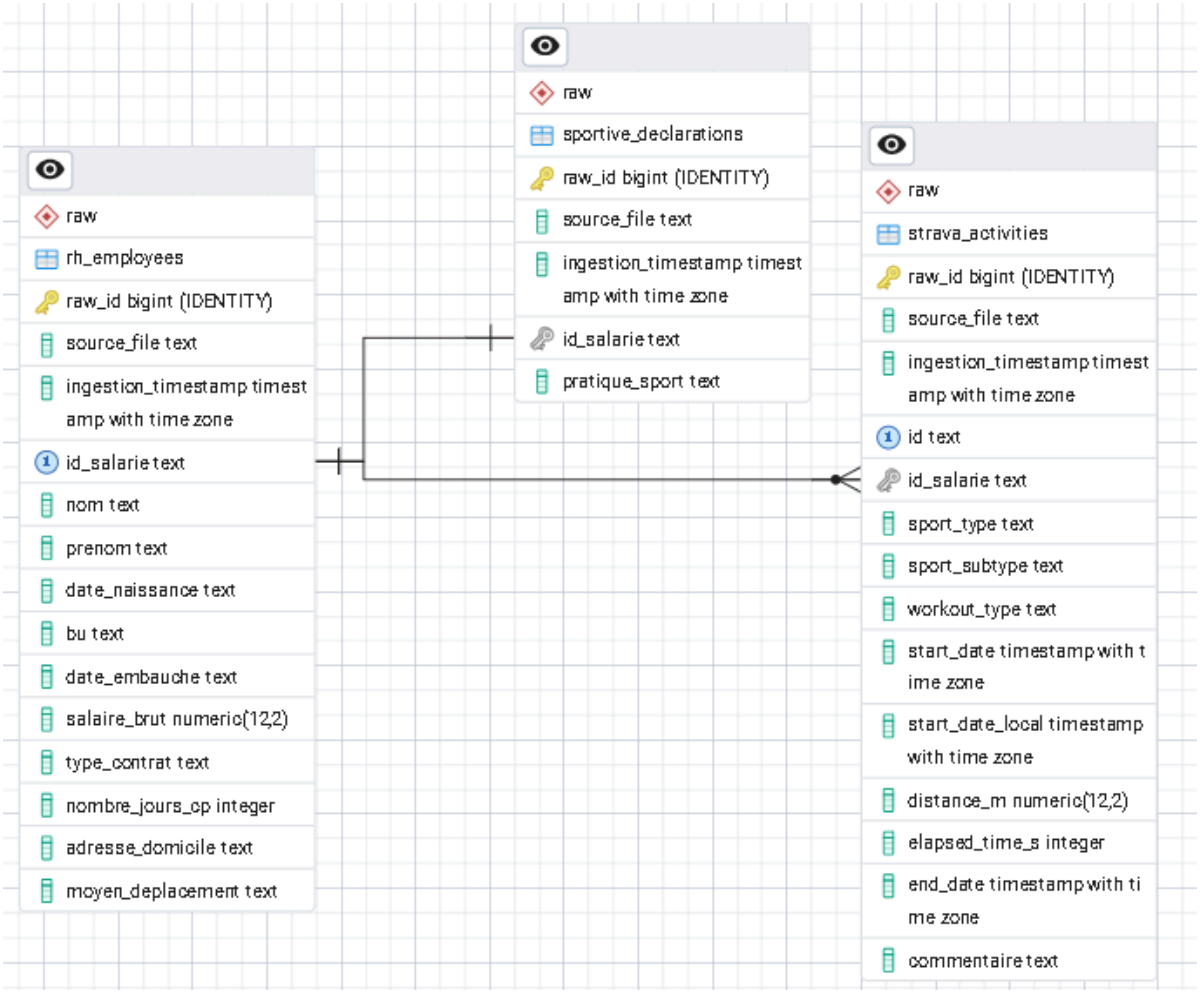
Tables clés :

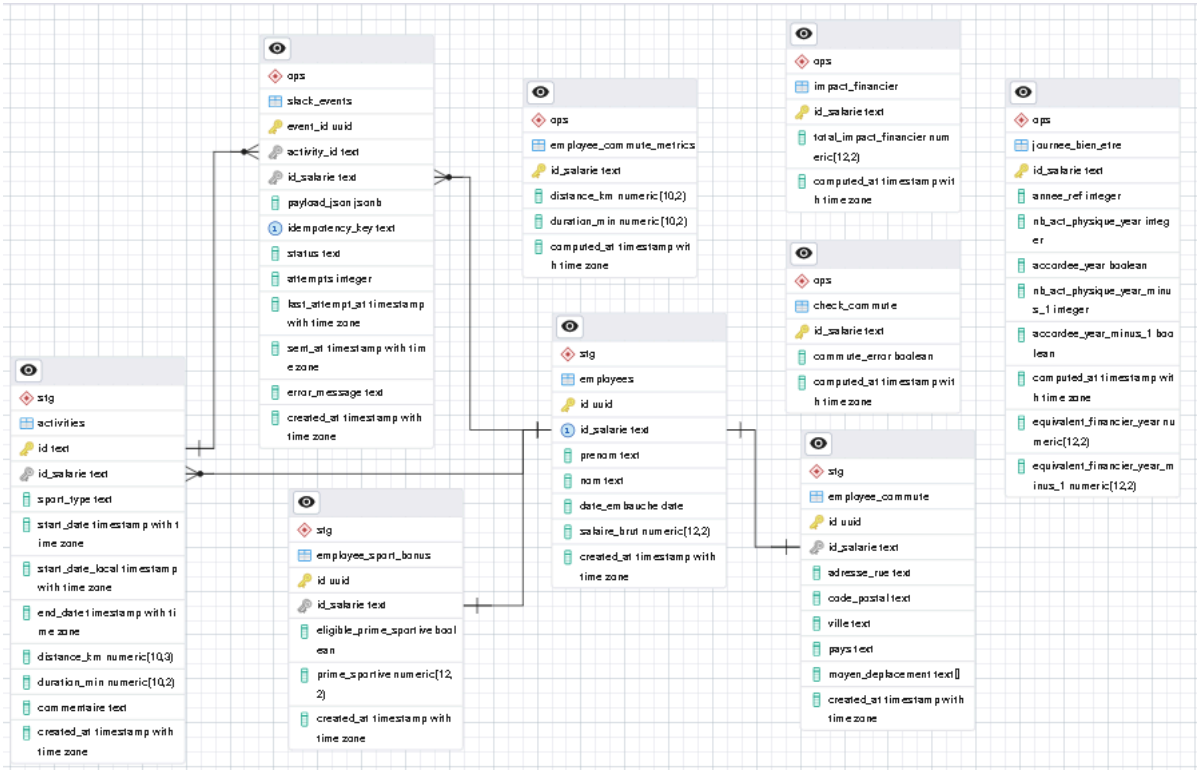
- `ops.employee_commute_metrics` → distance & durée calculées
- `ops.check_commute` → détection d'anomalies
- `ops.journee_bien_etre` → attribution des journées
- `ops.impact_financier` → impact global par salarié

Scripts associés :

- `commute.py` (API Google Routes) commute
- `check_commute.py` check_commute
- `journees_bien_etre.py` journees_bien_etre
- `impact_financier.py` impact_financier

📌 Tous les KPI Power BI sont calculés exclusivement dans cette couche.


raw
rh_employees
raw_id bigint (IDENTITY)
source_file text
ingestion_timestamp timestamp with time zone
id_salarie text
nom text
prenom text
date_naissance text
bu text
date_embauche text
salaire_brut numeric(12,2)
type_contrat text
nombre_jours_cp integer
adresse_domicile text
moyen_deplacement text
raw
sportive_declarations
raw_id bigint (IDENTITY)
source_file text
ingestion_timestamp timestamp with time zone
id_salarie text
pratique_sport text
raw
strava_activities
raw_id bigint (IDENTITY)
source_file text
ingestion_timestamp timestamp with time zone
id text
id_salarie text
sport_type text
sport_subtype text
workout_type text
start_date timestamp with time zone
start_date_local timestamp with time zone
distance_m numeric(12,2)
elapsed_time_s integer
end_date timestamp with time zone
commentaire text


ops
slack_events
event_id uuid
activity_id text
id_salarie text
payload_json jsonb
idempotency_key text
status text
attempts integer
last_attempt_at timestamp with time zone
sent_at timestamp with time zone
error_message text
created_at timestamp with time zone
ops
employee_commute_metrics
id_salarie text
distance_km numeric(10,2)
duration_min numeric(10,2)
computed_at timestamp with time zone
ops
impact_financier
id_salarie text
total_impact_financier numeric(12,2)
computed_at timestamp with time zone
ops
journee_bien_etre
id_salarie text
annee_ref integer
nb_act_physique_year integer
accordee_year boolean
nb_act_physique_year_minus_1 integer
accordee_year_minus_1 boolean
computed_at timestamp with time zone
equivalent_financier_year numeric(12,2)
equivalent_financier_year_minus_1 numeric(12,2)
ops
check_commute
id_salarie text
commute_error boolean
computed_at timestamp with time zone
stg
activities
id text
id_salarie text
sport_type text
start_date timestamp with time zone
start_date_local timestamp with time zone
end_date timestamp with time zone
distance_km numeric(10,3)
duration_min numeric(10,2)
commentaire text
created_at timestamp with time zone
stg
employees
id uuid
id_salarie text
prenom text
nom text
date_embauche date
salaire_brut numeric(12,2)
created_at timestamp with time zone
stg
employee_commute
id uuid
id_salarie text
adresse_rue text
code_postal text
ville text
pays text
moyen_deplacement text[]
created_at timestamp with time zone
stg
employee_sport_bonus
id uuid
id_salarie text
eligible_prime_sportive boolean
prime_sportive numeric(12,2)
created_at timestamp with time zone

5. Simulation Strava & données temps réel

Génération des données

- Notebook `strava_activities.ipynb`
- Historique simulé sur 12 mois
- Plusieurs milliers d'activités

Cette simulation répond explicitement à la demande de la note de cadrage Note+de+cadrage+_+POC+Avantages....

CDC avec Redpanda Cloud

Outil : Redpanda Cloud + Redpanda Connect
Configuration : `redpanda.yml`

Fonctionnement :

- CDC sur `raw.strava_activities`
- Transformation :
  - `distance_m → distance_km`
  - `elapsed_time_s → duration_min`
- Écriture :
  - table `stg.activities`
  - topic Redpanda (audit/debug)
  - message Slack

📌 Ce pipeline permet de tester un flux "quasi-temps réel" sans impacter le batch.

---

6. Orchestration du pipeline

Prefect Cloud

Fichier principal :

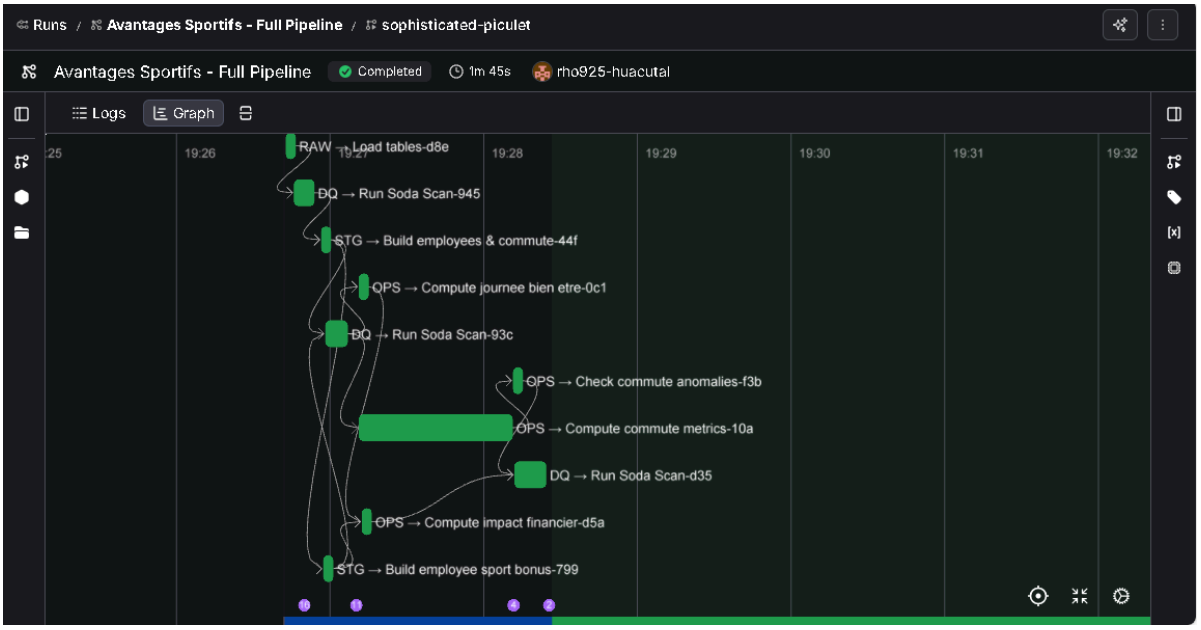
- `Prefect/flows/main_flow.py` main_flow

Ordonnancement logique :

1. RAW ingestion
2. Checks qualité RAW
3. STG transformations
4. Checks qualité STG
5. OPS calculs métier
6. Checks qualité OPS

Chaque étape dépend explicitement de la précédente.

📌 Avantage : lisibilité, reprise sur erreur, observabilité.



7. Qualité des données – Soda Cloud

Pourquoi Soda ?

- Tests déclaratifs (YAML)
- Interface Cloud pour suivi
- Intégration simple avec PostgreSQL

Fichiers :

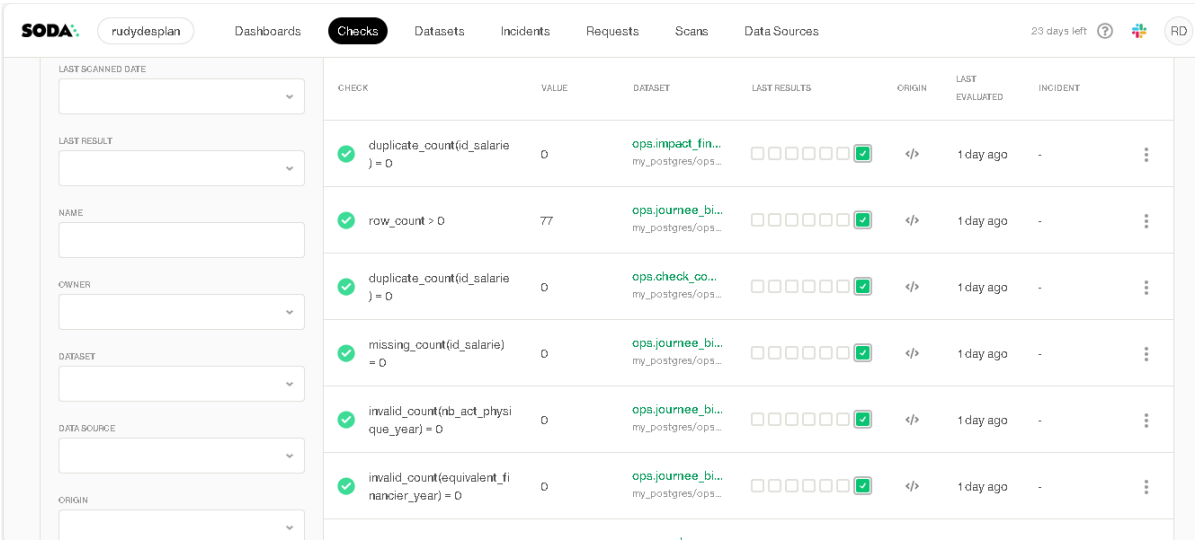
- `checks_raw.yml`
- `checks_stg.yml`
- `checks_ops.yml`
- `configuration.yml`

Documentation détaillée : `Data_quality/Readme.md` Readme

Types de contrôles :

- valeurs négatives
- duplicats
- cohérence dates
- intégrité métier (impact financier)

📌 En cas d'échec : le pipeline est bloqué (mode strict).



8. Monitoring & observabilité

Outil	Rôle
Aiven Console	DB, performance, connexions
Prefect Cloud	exécutions, logs, retries
Redpanda Cloud	CDC, messages, lag
Soda Cloud	qualité des données

👉 Tous les outils sont cloud managés, aucun monitoring “fait maison”.

9. Restitution Power BI & choix BI

Fichier `Report.pbix`

9.1 Choix du mode de connexion : DirectQuery

Mode utilisé : DirectQuery sur PostgreSQL (Aiven)

Justification du choix :

- Le projet intègre :
  - des flux quasi temps réel (CDC Strava via Redpanda),
  - des recalculs batch fréquents (OPS via Prefect).
- Le mode DirectQuery permet :
  - d’avoir une vue toujours à jour des indicateurs,
  - d’éviter toute duplication de données dans Power BI,
  - de garantir que les KPI affichés reflètent exactement l’état de la base OPS.

📌 Ce choix est cohérent avec :

- l’objectif “live / monitoring” du POC,

- l'utilisation de solutions cloud managées,
- une volumétrie encore maîtrisée (POC).

9.2 Organisation du rapport Power BI

Le rapport est structuré en 5 pages thématiques, chacune correspondant à une brique métier clairement identifiée dans la base OPS/STG.

👉 Principe clé :
Aucun KPI n'est recalculé côté BI à partir de données brutes RAW.
Power BI consomme uniquement :

- `stg.activities`
- `stg.employee_commute`
- `stg.employee_sport_bonus`
- `ops.*`

9.3 KPIs – Pratique sportive & activité

Tables sources

- `stg.activities`

Mesures DAX principales

Distance totale parcourue (km)

```
Distance totale (km) =
SUM ( stg_activities[distance_km] )
```

Distance moyenne par activité (km)

```
Distance moyenne par activité (km) =
AVERAGE ( stg_activities[distance_km] )
```

Durée totale (min)

```
Durée totale (min) =
SUM ( stg_activities[duration_min] )
```

Durée moyenne par activité (min)

```
Durée moyenne par activité (min) =
AVERAGE ( stg_activities[duration_min] )
```

Visualisation

- Histogramme / bar chart :
  - Axe X : `sport_type`
  - Valeur : `COUNT(*)`

📌 Objectif métier :
Visualiser l'intensité et la répartition des pratiques sportives sur l'année simulée.

9.4 KPIs – Mobilité domicile-travail

Tables sources

- `ops.employee_commute_metrics`
- `ops.check_commute`

Mesures DAX

Distance totale des trajets (km)

```
Distance totale trajets (km) =
SUM ( ops_employee_commute_metrics[distance_km] )
```

Durée totale des trajets (min)

```
Durée totale trajets (min) =
SUM ( ops_employee_commute_metrics[duration_min] )
```

Distance moyenne par salarié (km)

```
Distance moyenne par salarié (km) =
AVERAGE ( ops_employee_commute_metrics[distance_km] )
```

Durée moyenne par salarié (min)

```
Durée moyenne par salarié (min) =
AVERAGE ( ops_employee_commute_metrics[duration_min] )
```

Nombre d'anomalies de trajet

```
Nb anomalies trajets =
COUNTROWS (
    FILTER (
        ops_check_commute,
        ops_check_commute[commute_error] = TRUE()
    )
)
```

Taux d'anomalies de trajet

```
Taux anomalies trajets =
DIVIDE (
    [Nb anomalies trajets],
    COUNT ( ops_check_commute[id_salarie] )
)
```

📌 Objectif métier :

- Identifier les déclarations incohérentes,
- Mesurer la fiabilité des données déclaratives RH.

9.5 KPIs – Journées bien-être

Table source

- `ops.journee_bien_etre`

Mesures DAX

Nombre de salariés avec journée accordée

```
Nb salariés journée accordée =
CALCULATE (
    COUNT ( ops_journee_bien_etre[id_salarie] ),
    ops_journee_bien_etre[accordee_year] = TRUE()
)
```

Taux d'accord

```
Taux accord journée =
DIVIDE (
    [Nb salariés journée accordée],
    COUNT ( ops_journee_bien_etre[id_salarie] )
)
```

Total de journées accordées

```
Total journées accordées =
SUMX (
    ops_journee_bien_etre,
    IF ( ops_journee_bien_etre[accordee_year], 1, 0 )
)
```

Équivalent financier total journées bien-être

```
Equivalent financier total JBE =
SUM ( ops_journee_bien_etre[equivalent_financier_year] )
```

📌 Objectif métier :
Mesurer l'impact RH non monétaire et monétisé des pratiques sportives hors travail.

9.6 KPIs – Prime sportive

Table source

- `stg.employee_sport_bonus`

Mesures DAX

Nb de salariés éligibles prime sportive

```
Nb salariés éligibles prime =
CALCULATE (
    COUNT ( stg_employee_sport_bonus[id_salarie] ),
    stg_employee_sport_bonus[eligible_prime_sportive] = TRUE()
)
```

Montant total prime sportive

```
Montant total prime sportive =
SUM ( stg_employee_sport_bonus[prime_sportive] )
```

Prime moyenne (sur éligibles)

```
Prime moyenne éligibles =
CALCULATE (
    AVERAGE ( stg_employee_sport_bonus[prime_sportive] ),
    stg_employee_sport_bonus[eligible_prime_sportive] = TRUE()
)
```

📌 Objectif métier :
Évaluer le coût direct de l'avantage "mobilité active".

9.7 KPI – Impact financier global

Table source

- `ops.impact_financier`

Mesures DAX

Impact financier total (global)

```
Impact financier total =
SUM ( ops_impact_financier[total_impact_financier] )
```

Impact financier moyen par salarié

```
Impact financier moyen par salarié =
AVERAGE ( ops_impact_financier[total_impact_financier] )
```

Visualisation – Répartition par bins

- Création d'une colonne calculée :

```
Impact Bin =
FLOOR ( ops_impact_financier[total_impact_financier], 500 )
```

- Histogramme :
  - Axe X : `Impact Bin`
  - Valeur : `COUNT(id_salarie)`

📌 Objectif métier :
Visualiser la distribution des coûts et identifier les profils à fort impact.

9.8 Cohérence avec l'architecture globale

✔️ Les KPI Power BI :

- reposent uniquement sur des tables OPS ou STG validées,
- sont protégés par des checks Soda Cloud,
- sont recalculés automatiquement via Prefect,
- reflètent les flux CDC Strava en quasi temps réel.

👉 Power BI devient donc :

> une couche de visualisation fiable, temps réel, et alignée avec le pipeline data.