

OCR - PROJET 4

Audit d'un environnement de données

Parcours DATA ENGINEER

Mathilde LE SOLLIEC

09/09/2025

SOMMAIRE

1

Contexte

2

Architecture de l'entreprise

3

Compréhension des problèmes

- prototype en local
- et sur base de l'analyse des logs

4

Recommandations

1. Contexte

SuperSmartMarket

- Une chaîne de supermarchés
- Personnalise l'expérience en magasin grâce au suivi des comportements clients
- Une équipe de data analyst suit les ventes (types de produits, prix, employés, clients...)



Problématique : problème de cohérence des données remontées

Les chiffres d'affaires historiques **changent dans le temps.**

Le responsable de tout le pôle Business Intelligence a observé le CA du 14 août était de :

- 275 186,59 € le 14 août
- 284 243,88 € le 16 août sans nouvelle vente



Projet : en tant que Data Engineer



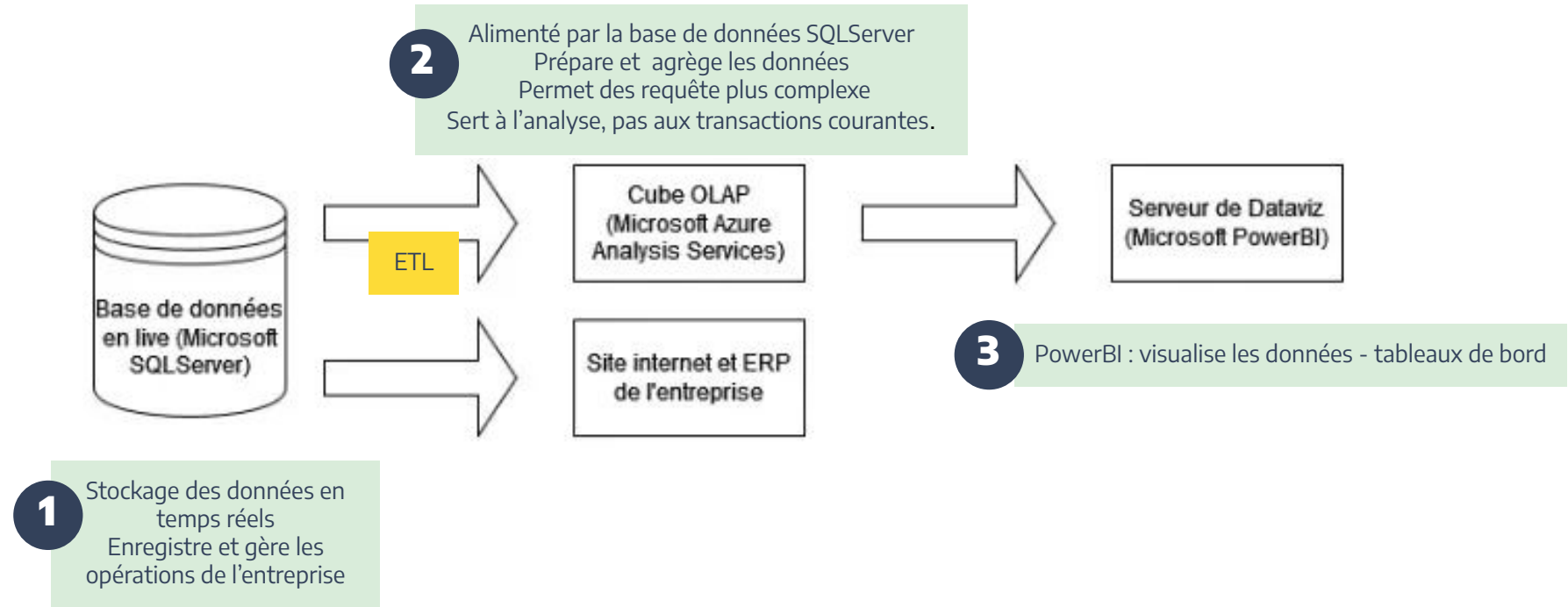
Comprendre et analyser les flux de données de l'entreprises



Garantir la sécurité et la qualité des données pour l'ensemble de l'entreprise

2. Architecture de données

Etape 1 : comprendre l'architecture

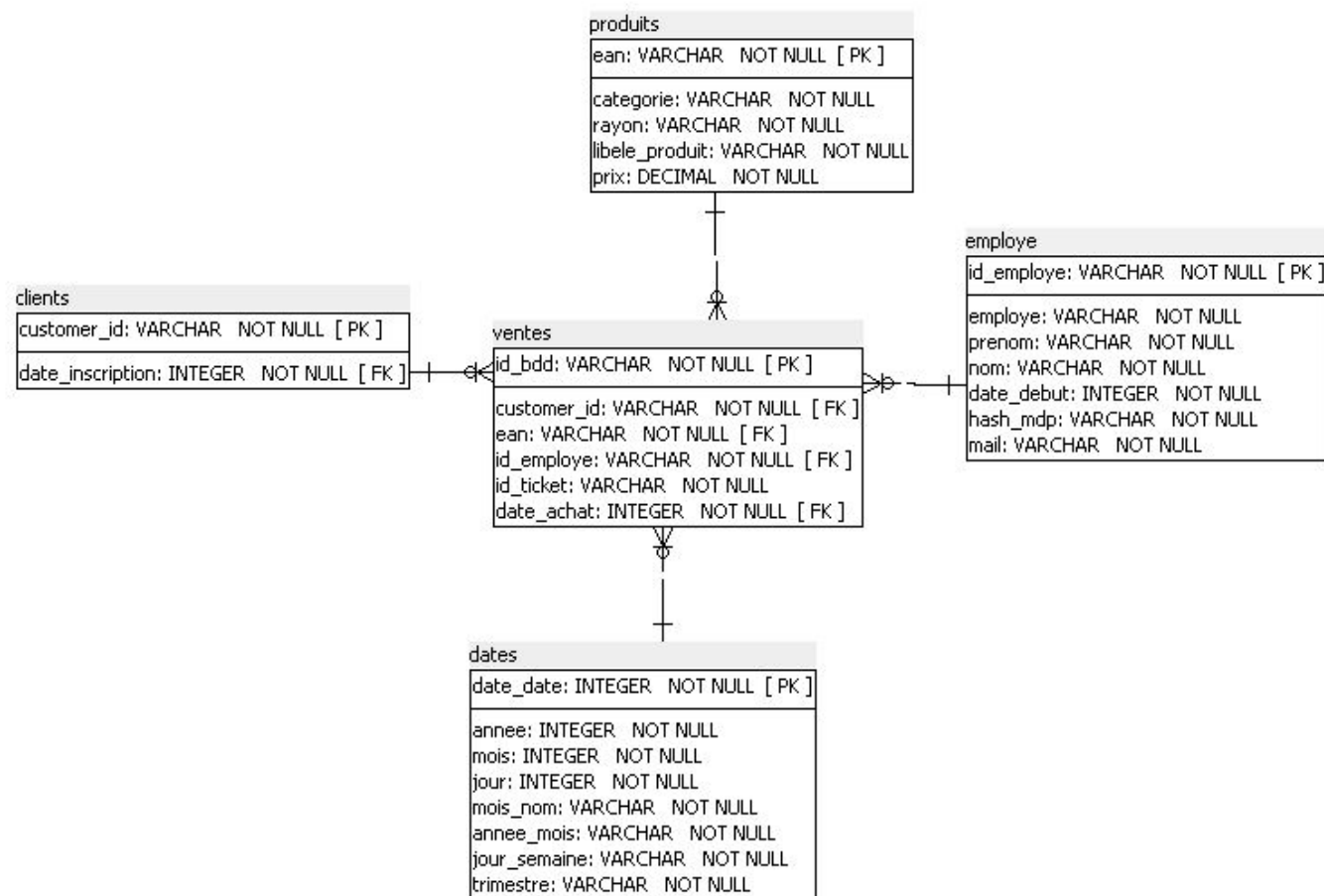


Etape 2 : documenter la base de donnée OLAP

Schéma relationnel

Modèle en étoile

- 1 table de fait
ventes
- 4 tables de dimensions
clients,
employes,
produits,
dates,



3. Compréhension des problèmes

Test des données du cube dans un prototype en local

En local (PostgreSQL)

Vérification du chiffres d'affaire

Pour rappel, le responsable de tout le pôle Business Intelligence a observé le CA du 14 août:

- 275 186,59 € le 14 août
- 284 243,88 € le 16 août sans nouvelle vente

Résultat chiffre d'affaires total
pour le 14 Août

284 243,88 euros

Requête

```
with achat as (  
    select ean,  
    count (ean) as nbr_achat  
    from p4.ventes  
    left join p4.dates  
    on p4.ventes.date_achat = p4.dates.date_date  
    where annee = '2024'  
    and mois = '08'  
    and jour = '14'  
    group by ean  
)  
select  
    sum(achat.nbr_achat * prix) as chiffre_affaire  
from p4.produits  
left join achat using(ean);
```

2

Question et résultat

Le chiffre d'affaires par client
pour le top 10

	id_customer character varying 	chiffre_affaire numeric 
1	CUST-JNSOZSFORR...	846.86
2	CUST-GM6VBAYAB...	666.86
3	CUST-L2ST2JHI7K90	644.18
4	CUST-WU7ZKQJE4L...	608.93
5	CUST-9WM83101Q...	582.03
6	CUST-ZMAOVX8XY...	576.39
7	CUST-3K66CV00H...	571.44
8	CUST-CG23SXJDRN...	531.09
9	CUST-D8IOFHVUFX...	477.35
10	CUST-IHN1HQRI7PYJ	463.73

Requête

```
with achat as (  
    select id_customer,  
           ean,  
           count(ean) as nbr_achat  
    from p4.ventes  
    group by ean, id_customer  
)  
select id_customer,  
       sum(achat.nbr_achat * prix) as chiffre_affaire  
from p4.produits  
inner join achat using(ean)  
group by id_customer  
order by chiffre_affaire desc  
limit 10;
```

3

Question et résultat

Le chiffre d'affaires encaissé par employé

id_employe character varying	chiffre_affaire numeric	part_pourcent numeric
f491076a1ff2d873ebea809c11144542	7818.82	2.75
e01e752175e05f00c8314ccb8da4c4...	7736.16	2.72
8d1001fbad3d2a60ff7530600ed5d55e	6995.14	2.46
6c1c3292c852c6c593b95cc146b00c...	6616.46	2.33
a7ada0770091e838e3dcd45265282...	6483.84	2.28
2477db17c02f512ebc4b20f01a7edb...	6361.22	2.24
528c733809cb51a3634befb260b5d2...	6133.34	2.16
dd595f0f0b3400df2908f0be7723dad4	6111.18	2.15

Requête

-- 3 - la part de chiffre d'affaires encaissé par employé.

```
with achat as (
    select id_employe,
           ean,
           count(ean) as nbr_achat
    from p4.ventes
    group by ean, id_employe
),

ca_par_employe as (

    select id_employe,
           sum(achat.nbr_achat * prix) as chiffre_affaire
    from p4.produits
    inner join achat using(ean)
    group by id_employe
)

select
    id_employe,
    chiffre_affaire,
    ROUND(chiffre_affaire / sum(chiffre_affaire) over () * 100, 2) as part_pourcent
from ca_par_employe
order by part_pourcent desc;
```

Compréhension des problèmes sur la base des logs

Les logs sauvegardes l'historique :

- de différentes types actions
(insertion, modification, suppressions)
- sur les tables du cube OLAP
(Ventes, Produits, Clients, Employés)

Nombre de log par action et par table

type_action character varying 🔒	table_insert character varying 🔒	nrb_log bigint 🔒
INSERT	Ventes	206885
UPDATE	Produits	575
INSERT	Client	20
UPDATE	Employé	7
DELETE	Employé	2

- 207 489 logs
- Principalement des INSERT dans la table ventes

Compréhension des problèmes sur la base des logs

Les champs modifiés

type_action	table_insert	champs	nrb_log
character varying	character varying	character varying	bigint
INSERT	Ventes	ID ticket	41377
INSERT	Ventes	id_employe	41377
INSERT	Ventes	CUSTOMER_ID	41377
INSERT	Ventes	Date	41377
INSERT	Ventes	EAN	41377
UPDATE	Produits	prix	575
INSERT	Client	date_inscription	20
UPDATE	Employé	hash_mdp	7
DELETE	Employé	[null]	2

Chaque ventes provoquent 5 logs, qui modifie chaque colonne de la table vente

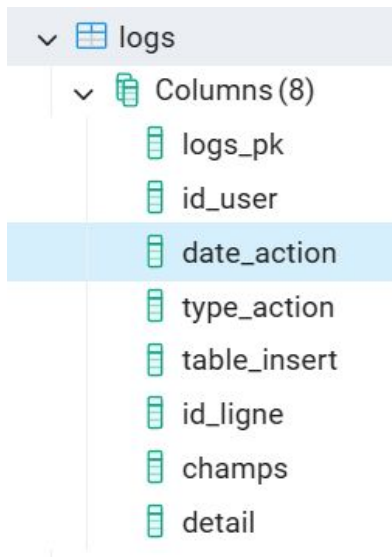
Requête associée

```
select
  type_action
  ,table_insert
  ,champs
  ,count(logs_pk) as nrb_log
from p4.logs
group by type_action,table_insert, champs
order by count(logs_pk) desc;
```

Compréhension des problèmes sur la base des logs

Autres informations de la table logs

Colonnes de la table des logs



▼	logs
▼	Columns (8)
	logs_pk
	id_user
	date_action
	type_action
	table_insert
	id_ligne
	champs
	detail

id_user : qui a fait l'action
Toujours le même : compte de service utilisé par l'outil d'intégration

Compréhension des problèmes sur la base des logs

Rappel de la problématique

Les chiffres d'affaires historiques changent dans le temps

L'erreur détecté est en lien avec le chiffre d'affaire : **on se concentre en priorité sur les logs des ventes & produits (update des prix)**

type_action character varying 🔒	table_insert character varying 🔒	champs character varying 🔒	nrb_log bigint 🔒
INSERT	Ventes	ID ticket	41377
INSERT	Ventes	id_employe	41377
INSERT	Ventes	CUSTOMER_ID	41377
INSERT	Ventes	Date	41377
INSERT	Ventes	EAN	41377
UPDATE	Produits	prix	575
INSERT	Client	date_inscription	20
UPDATE	Employé	hash_mdp	7
DELETE	Employé	[null]	2

Hypothèse 1

problème au niveau des updates des prix

Vérification que l'UPDATE des prix a bien mis
été à jours dans le cube

Résultat



- UPDATE des prix bien intégré à la table ventes

Requête

```
-- ERREUR : calcul du CA du 15/08

with ventes_15_08 as (
    select
        detail,
        count(detail) as nbr_ventes
    from p4.logs
    where date_action = DATE '2024-08-15'
        and champs = 'EAN'
    group by detail)

select
    sum(nbr_ventes*prix) as chiffre_affaire
from p4.produits p
left join ventes_15_08 v
on v.detail = p.ean
;
```

Extrait des résultats

categorie	prix_update_log	prix_produit
character varying	text	text
Produits Secs & Conserves	02.08	2.08
Produits Secs & Conserves	03.02	3.02
Produits Secs & Conserves	04.07	4.07

Hypothèse 2

les ventes ne sont pas remontés dans le cube OLAP

Exploration :

- comparaison du nombre de log de la
table vente et du nombre de vente



Toutes les ventes sont
remontées

Il a bien été vérifié que :

- 41 377 logs ont été enregistrées
concernant un update de vente
- la table vente également
comptabilise 41 377 ventes.

```
select
    count(*) as nrb_log_ventes
from p4.logs
where champs = 'ID ticket'
;

select
    count(*) as nbr_ventes
from p4.ventes
```

Hypothèse 3

Les ventes ne sont pas remontées le jour même dans le cube OLAP

La table LOGS montre :

- des ventes ont été insérées dans le cube OLAP le 15-08-2024
- alors que les ventes ont été réalisées le 14-08-2024

- Cela concerne une partie des ventes

Extrait de la table logs

date	action	table_insert	champs	detail
15/08/2024	INSERT	Ventes	CUSTOMER_ID	CUST-B8W3YSU9HIUX
15/08/2024	INSERT	Ventes	id_employe	951e47dfa4c5298382d1b7d75f
15/08/2024	INSERT	Ventes	EAN	6 386 228 298 857
15/08/2024	INSERT	Ventes	Date	14/08/2024

La date d'insertion dans le cube OLAP (colonne 1) est différente de la date de la vente (détail)

Nombre de logs concernés

nbr_log_ventes_15
bigint
6895

```
SELECT count(*) as nbr_log_ventes_15
FROM p4.logs l
WHERE table_insert = 'Ventes'
AND date_action = DATE '2024-08-15';
```

Vérification

Cela correspondait bien au **9057,29 Euros de chiffre d'affaire de différence** entre les deux chiffres d'affaire observés par le responsable de service.

Certaines lignes ont été chargées dans le cube après les autres : **Problème d'atomicité**

Requête vérifiant le chiffre d'affaire correspondant au décalage observé

```
-- ERREUR : calcul du CA du 15/08

with ventes_15_08 as (
    select
        detail,
        count(detail) as nbr_ventes
    from p4.logs
    where date_action = DATE '2024-08-15'
        and champs = 'EAN'
    group by detail)

select
    sum(nbr_ventes*prix) as chiffre_affaire
from p4.produits p
left join ventes_15_08 v
on v.detail = p.ean
;
```

chiffre_affaire
numeric

9057.29

Compréhension des problèmes sur la base des logs

Vérification : les ventes dont l'intégration a été décalée **ont-elles une caractéristique en commun ?**

Caractéristiques	Résultats
Type de produit	1 322 produits concernés - pas un produit en particulier
Employé	56 employés - pas un en particulier
id_user	id_user 08c8b678f8e6f0caz05880ef4ebba10az -pas spécifique aux logs erronés : ne permet pas de retracer qui a fait une erreur

Aucune caractéristique commune n'a pu être observée

Requête associée

```
- exploration du type de insert qui ont eu des erreurs
SELECT detail,
       count(logs_pk)
FROM p4.logs
WHERE date_action = DATE '2024-08-15'
      and champs = 'id_employe'
GROUP BY detail
order by count(logs_pk) desc
```

Plusieurs raisons pourraient expliquer le décalage d'intégration de certaines données dans le cube :

Ressources insuffisantes

Manque de mémoire de la base sql server pour ingérer un gros volume de données.

Conséquence : certaines insertions échouent ou prennent beaucoup de temps → décalage dans le cube

Taille excessive des données

Si un lot est très volumineux, il peut dépasser les limites du serveur ou provoquer un timeout.

Une contrainte non respectée

ex : mauvais format de date

La **table LOG ne donne pas suffisamment d'information et de suivi pour s'assurer que ce type d'erreur ne se reproduise pas.**

Plusieurs recommandations peuvent être évoquées pour résoudre cela et avoir une meilleure prise en charge des erreurs.

4. Recommendations

Recommandation - Plan d'action

Étape	Délai	Action	Objectif	Où
1	2 sem.	Améliorer Table de log	Avoir une trace des erreurs de chargement des lignes qui ne passent pas vers le cube OLAP	SQL Server
		Transaction atomique	Si une seule étape échoue, tout le bloc est annulé, et la base reste cohérente	SQL Server
		Vérifier - ajouter des contraintes d'intégrité	Bloquer données incohérentes	SQL Server
		Gestion des droits	Gérer les accès et suivre les modifications	SQL Server Management Studio
2	1 mois	Monitoring ETL	Visualisation pour suivre échecs et écarts	Azure Data Factory
		Triggers	Alerter sur erreurs	SQL Server + Agent Jobs

1- Amélioration table log

Objectif

Avoir une trace des erreurs de chargement des lignes qui ne passent pas vers le cube OLAP

Centraliser les anomalies

Historisation, suivi des erreurs, audit → Facilite le monitoring

- Ajouter dans la table logs les colonnes nécessaires pour diagnostiquer les anomalies
- Statut (Rejetée, Réussie, Bloquée...)
- heures exactes de l'action - du job
- valeur avant/après

Dans notre cas, cela aurait permis de retracer **les ventes qui n'ont pas été insérées à temps** et comprendre le décalage - chargement à échoué

2- Transaction atomique

Objectif

Préserve l'intégrité de données → s'il y a anomalie un lot est complètement validé ou annulé.

- Si **une seule étape échoue**, tout le bloc est annulé, et la base reste cohérente
- Éviter des **incohérences partielles** qui créent des écarts de chiffre d'affaires jour par jour.

Dans notre cas, tout le lot vente auraient été bloqué en cas d'erreur , ce qui aurait évité écarts de chiffre d'affaires jour par jour.

3- Vérifier les contraintes d'intégrité

Empêchent qu'une donnée incohérente entre dans la base ou passe le flux.

Les contraintes communes :

- Unique PK, FK
- NOT NULL

Vérification de la bonne configuration des cascades

Les contraintes personnalisées :

- Limites sur les valeurs (parmi une liste autorisée..) (contrainte de domaine)
- CHECK : règles personnalisées (ex. une remise \leq 100%).

Peut être joint à un triggers pour être notifié de l'erreur.

3- Sécuriser la gestion des droits d'administration

Limiter les risques liés aux modifications non contrôlées.

- o Gérer l'accès des droit avec l'authentification , limiter les permissions attribuée, définir des niveaux (administrateur –lecteur – writer – admin...), tracer les actions via un audit log

Outil : SQL Server Management Studio.

4 - Triggers

Programme qui se déclenche automatiquement une action après qu'une erreur survient.

Prévoir un trigger à chaque erreur :

Ex:

- écrire la ligne rejetée dans une table de logs avec le motif de l'erreur,
- envoyer un mail à l'équipe,

Dans notre cas : associer un triggers aux contraintes SQL (filet de sécurité)

5 - Monitoring des flux

Surveiller et suivre les
erreur après coup

Visualisation global sur :

Echec du job ETL
Lot incomplet,
Temps de traitement des jobs,
Écart entre source et cube.

Mise en place d'un dashboard pour visualiser les
volumes de données, délais de chargements :

- à travers des outils (ex; Azure Data Factory (ADF))
- ou créer soit-même un dashboard sur (Power BI) -> a voir

Implémentation des recommandations

```
--** 1- Améliorer la table de log **

-- Ajouter la colonne statut
ALTER TABLE p4.logs
ADD COLUMN statut VARCHAR(20) CHECK (statut IN ('réussie','rejetée','bloquée'));

-- Ajouter la colonne details
ALTER TABLE p4.logs
ADD COLUMN details TEXT;

-- Ajouter la colonne job_at en timestamps
ALTER TABLE p4.logs
ADD COLUMN job_at TIMESTAMP DEFAULT NOW();
```

Implémentation des recommandations

```
** 3 - Verifier les contraintes d'intégrités **
```

```
-- PK FK : déjà intégrés lors de la création
```

```
ALTER TABLE p4.produits  
ADD CONSTRAINT prix CHECK (prix >= 0);
```

```
ALTER TABLE p4.produits  
ALTER COLUMN prix SET NOT NULL;
```

```
-- pour que la suppression des employés ne supprime pas les ventes rattachés,  
-- au lieu de supprimer l'employé, j'ajoute une colonne 'is_active'
```

```
ALTER TABLE p4.employes  
ADD COLUMN is_active BOOLEAN DEFAULT TRUE;
```

```
-- j'empêche sa suppression
```

```
ALTER TABLE p4.ventes  
ADD CONSTRAINT ventes_id_employe_fk  
FOREIGN KEY (id_employe)  
REFERENCES p4.employes(id_employe)  
ON DELETE RESTRICT;
```

Implémentation des recommandations

--5 . Mise en place de triggers

```
CREATE FUNCTION p4.check_price()
RETURNS TRIGGER AS
$$ BEGIN
    IF NEW.prix <= 0 THEN
        RAISE EXCEPTION 'Prix négatif interdit';
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_check_price
BEFORE INSERT OR UPDATE ON p4.produits
FOR EACH ROW
EXECUTE FUNCTION p4.check_price();
```

-- test

```
INSERT INTO p4.produits (ean, categorie, rayon, libele_produit, prix)
VALUES (577933, 'Produits Secs & Conserves', 'pates', 'pastabox', -3);
```


Sources

1. Documentation officielle

SQL Server Docs – transaction log <https://learn.microsoft.com/fr-fr/sql/relational-databases/logs/the-transaction-log-sql-server?view=sql-server-ver17>

SQL Server Docs – Triggers : <https://learn.microsoft.com/sql/t-sql/statements/create-trigger-transact-sql>

SQL Server Docs – SQL Server Agent Jobs : <https://learn.microsoft.com/fr-fr/ssms/agent/sql-server-agent>

SQL Server Docs – Integrity Constraints : <https://learn.microsoft.com/sql/relational-databases/tables/primary-and-foreign-key-constraints>

Azure Data Factory – Monitoring and Alerts : <https://learn.microsoft.com/azure/data-factory/monitor-visually>

Microsoft Learn – Surveiller les actualisations des flux de données :

<https://learn.microsoft.com/fr-fr/power-query/dataflows/monitor-dataflow-refreshes>

2. Ressources pédagogiques

DataCamp – Introduction aux transactions ACID : <https://www.datacamp.com/blog/acid-transactions>

DataCamp – Guide des triggers SQL pour débutants : <https://www.datacamp.com/tutorial/sql-triggers>

DataCamp – Architecture d'un Data Warehouse : <https://www.datacamp.com/blog/data-warehouse-architecture>

Decivision – Analysis Services : Présentation, avantages et inconvénients de l'outil:

<https://www.decivision.com/blog/microsoft-bi/presentation-analysis-services>

Suites

Au delà de l'aspect technique : enjeux de bien communiquer sur les risques avec l'équipe métier, faire de la prévention.

Travail sur un prototype en local : en réalité, souvent en environnement test directement dans les outils.