

# AUDITEZ UN ENVIRONNEMENT DE DONNÉES

Projet Data Engineer - KELLENI Antoine



# OBJECTIF DE LA MISSION



- Comprendre l'architecture data existante
- Analyser le cycle de vie des données
- Étudier les logs de la base de données
- Identifier les risques techniques
- Formuler des recommandations concrètes



# CE QUE NOUS ALLONS DÉCOUVRIR

Déroulé



**l'architecture de la BDD**



**le dictionnaire des données**



**le schéma relationnel de la BDD**



**le résultat des requêtes**



**la compréhension des logs**



**la conclusion sur le problème qui a été identifié**



**une série de mesures correctives**

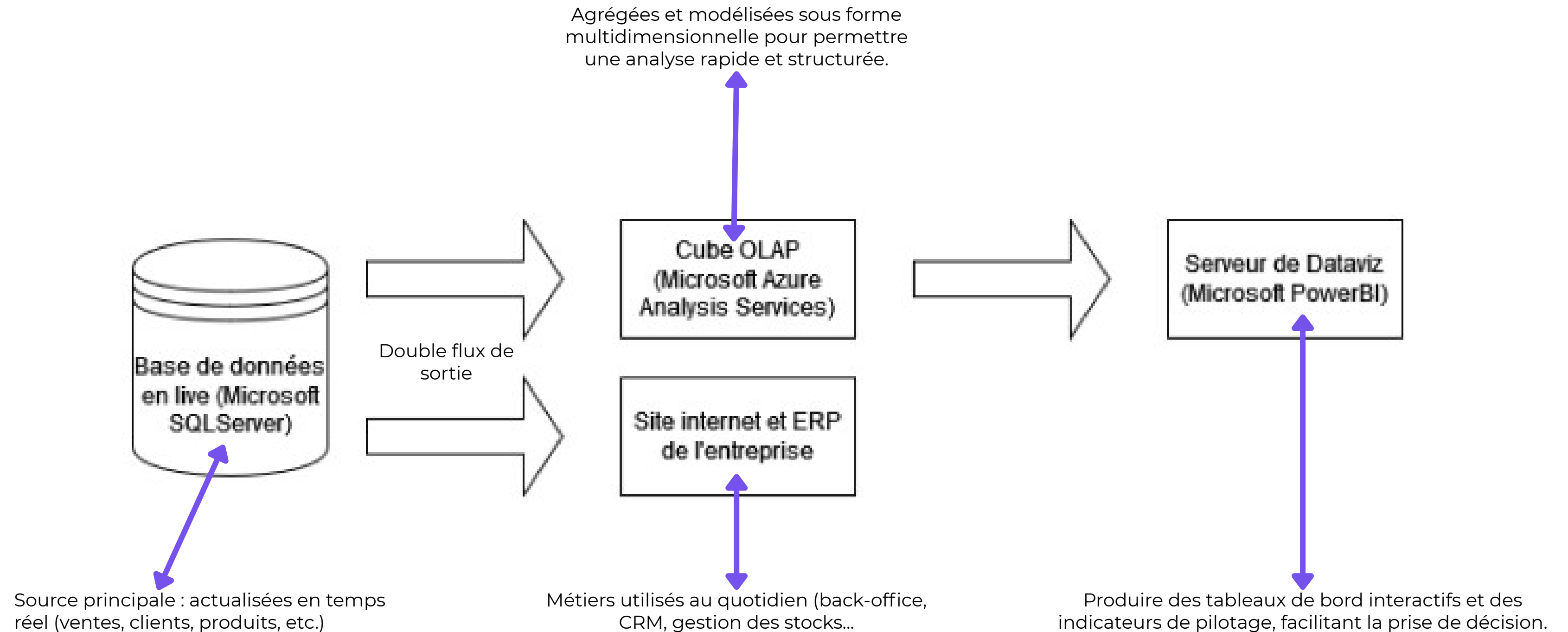


**le prototype de nouvelle BDD**



# L'ARCHITECTURE DE LA DBB

Cette architecture illustre la chaîne de traitement des données de l'entreprise, depuis l'origine jusqu'à la visualisation.



# LE DICTIONNAIRE DES DONNÉES

Table : **sale\_detail**

Code	Signification	Type	Longueur	Nature	Règle de gestion
id_sale	identifiant unique de la vente	VARCHAR	64	Clé primaire	NOT NULL
ean	identifiant du produit	INTEGER	-	Clé étrangère	NOT NULL et référence à product.ean
customer_id	identifiant du client	VARCHAR	25	Clé étrangère	NOT NULL et référence à customer.customer_id
id_employe	identifiant de l'employée	VARCHAR	64	Clé étrangère	NOT NULL et référence à employe.id_employe
date	date de la vente	INTEGER	-	Clé étrangère	NOT NULL et référence à calendar.date
id_ticket	identifiant du ticket de la vente	VARCHAR	16	Attribut	NOT NULL

Table : **product**

Code	Signification	Type	Longueur	Nature	Règle de gestion
ean	identifiant du produit	INTEGER	-	Clé primaire	NOT NULL et référence à sale_detail.ean
name_product	nom complet du produit	VARCHAR	255	Attribut	Référence commune.code_commune ET NOT NULL
price	prix du produit	VARCHAR	(10;2)	Attribut	NOT NULL
id_aisle	identifiant du rayon	INTEGER	-	Clé étrangère	référence à supermarket_aisle.id_aisle
id_category	identifiant de la catégorie du produit	INTEGER	-	Clé étrangère	référence à category.id_category

Table : **category**

Code	Signification	Type	Longueur	Nature	Règle de gestion
id_category	identifiant de la catégorie d'un produit	INTEGER	-	Clé primaire	NOT NULL et référence à product.id_category
category_name	nom de la catégorie	VARCHAR	100	Attribut	NOT NULL

Table : **supermarket\_aisle**

Code	Signification	Type	Longueur	Nature	Règle de gestion
id_aisle	identifiant d'un rayon	INTEGER	-	Clé primaire	NOT NULL et référence à product.id_aisle
aisle_name	nom d'un rayon	VARCHAR(100)	100	Attribut	NOT NULL

# LE DICTIONNAIRE DES DONNÉES

Table : **calendar**

Code	Signification	Type	Longueur	Nature	Règle de gestion
date	date au format d'un nombre à 5 chiffres	INTEGER	-	Clé primaire	NOT NULL et référence à sale_detail.date
year	année	INTEGER	-	Clé étrangère	NOT NULL
month	mois	INTEGER	-	Attribut	NOT NULL
month_name	nom du mois	VARCHAR	20	Attribut	NOT NULL
number_day_of_month	numéro du jour dans le mois	INTEGER	-	Attribut	NOT NULL
day_week_number	numéro du jour dans la semaine	INTEGER	-	Attribut	NOT NULL
quarter	trimestre	VARCHAR	3	Attribut	NOT NULL
date_reelle	date au format YYYY-MM-DD	DATE	-	Attribut	NOT NULL

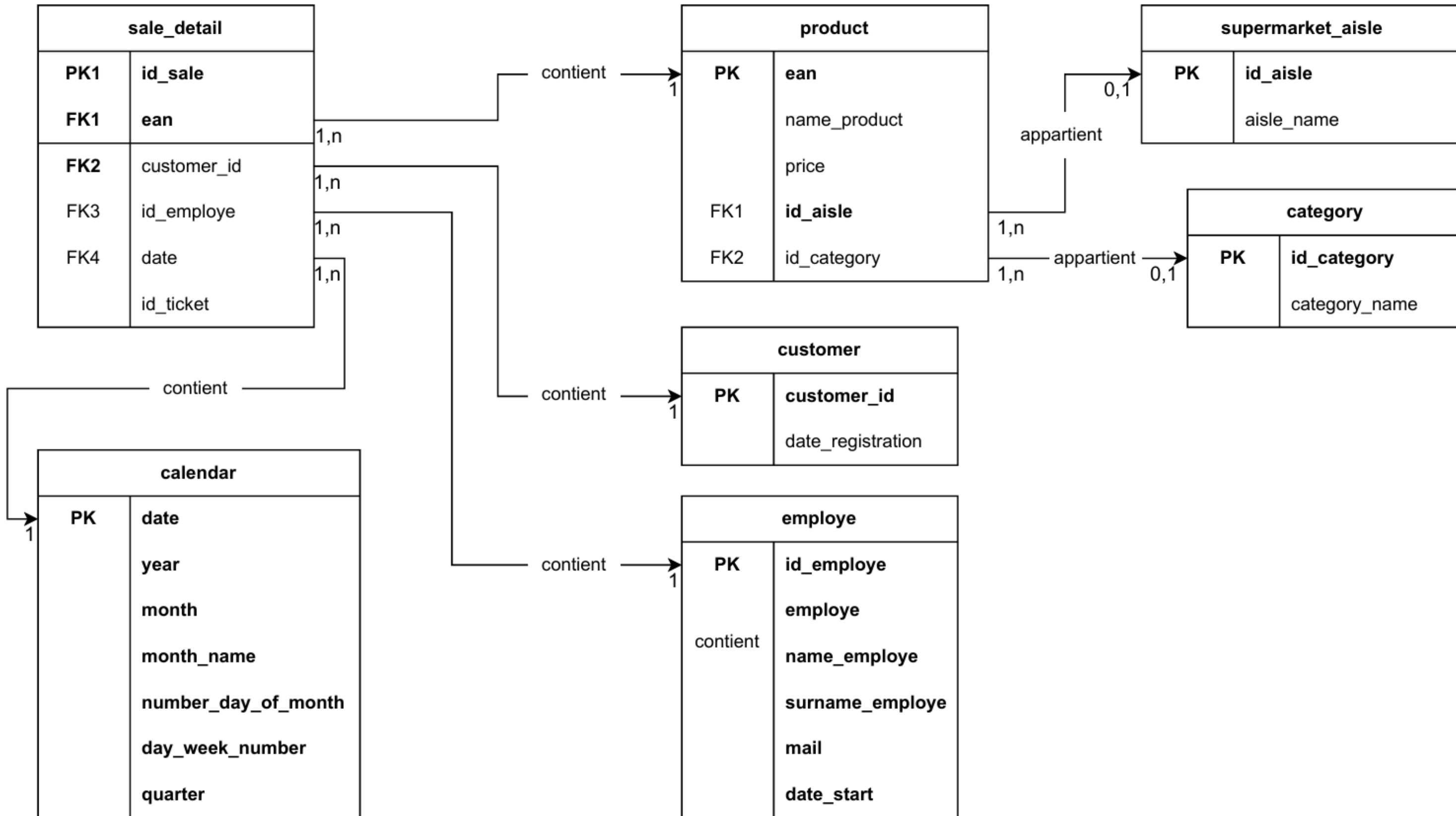
Table : **customer**

Code	Signification	Type	Longueur	Nature	Règle de gestion
customer_id	identifiant d'un client	VARCHAR	25	Clé primaire	NOT NULL et référence à sale_detail.customer_id
date_registration	date de l'inscription d'un client	DATE	-	Attribut	-

Table : **employe**

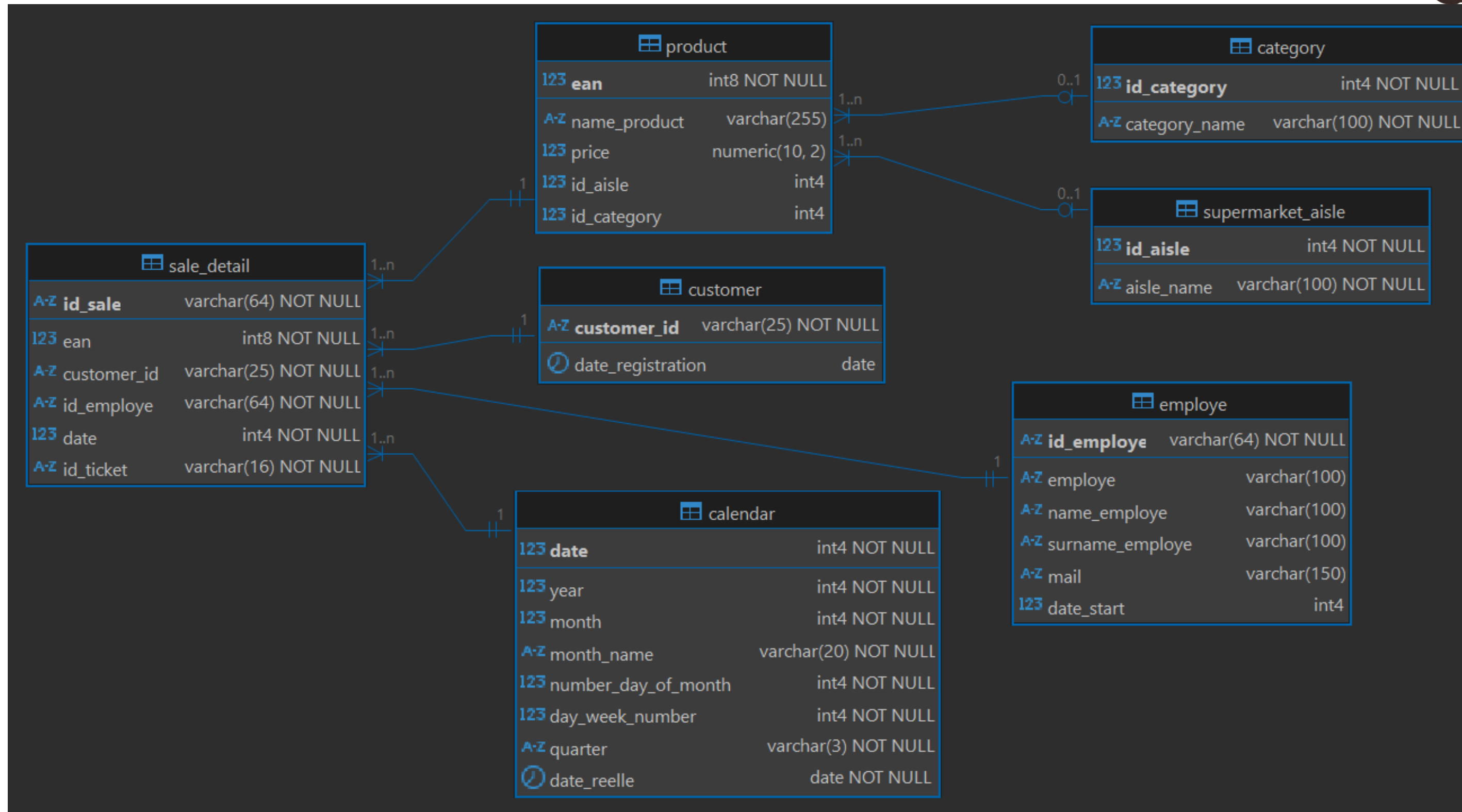
Code	Signification	Type	Longueur	Nature	Règle de gestion
id_employe	identifiant de l'employé	VARCHAR	64	Clé primaire	NOT NULL et référence à sale_detail.id_employe
employe	1er lettre du prénom avec le nom collé	VARCHAR	100	Attribut	-
name_employe	prénom de l'employé	VARCHAR	100	Attribut	-
surname_employe	nom de l'employé	VARCHAR	100	Attribut	-
mail	mail de l'employé	VARCHAR	150	Attribut	-
date_start	date de commencement (nombre à 5 chiffres)	INTEGER	-	Attribut	-

# LE SCHÉMA RELATIONNEL





# DIAGRAMME RELATIONNEL ERD





# RÉSULTAT DES REQUÊTES SQL

Le chiffre d'affaires pour le 14 août 2024 (date 45518)

Chiffre d'affaires : 284243,88 €  
Nombre total de ventes : 41377

SELECT

SUM(p.price) AS total\_ca,

COUNT(\*) AS total\_ventes

FROM sale\_detail sd

JOIN product p ON sd.ean = p.ean

JOIN calendar c ON sd.date = c.date

WHERE c.date = 45518;

Résultats 1

SELECT SUM(p.price) AS total\_ca, COUNT(\*) AS total\_ven

	123 total_ca	123 total_ventes
1	284243,88	41377

# RÉSULTAT DES REQUÊTES SQL

## Le chiffre d'affaires par client pour le top 10 des clients

- Le client CUST-JNSOZSFORR88 arrive en tête avec 846,86€ répartis sur 11 achats (produits).
- Tous les clients du top 10 ont généré entre 463€ et 846€ de chiffre d'affaires.

Cette analyse permet d'identifier les clients les plus rentables.

```

SELECT
    s.customer_id,
    COUNT(*) AS nb_achats,
    ROUND(SUM(p.price), 2) AS chiffre_affaires
FROM sale_detail s
JOIN product p ON s.ean = p.ean
GROUP BY s.customer_id
ORDER BY chiffre_affaires DESC
LIMIT 10;

```

sale\_detail 1

Tableau

Texte

SELECT s.customer\_id, COUNT(\*) AS nb\_achats, ROUND(SUM(p.price), 2) AS chiffre\_affaires

FROM sale\_detail s

JOIN product p ON s.ean = p.ean

GROUP BY s.customer\_id

ORDER BY chiffre\_affaires DESC

LIMIT 10;

Entrez une expression

	A-Z customer_id	123 nb_achats	123 chiffre_affaires
1	CUST-JNSOZSFORR8	11	846,86
2	CUST-GM6VBAYAB8SF	33	666,86
3	CUST-L2ST2JHI7K9O	38	644,18
4	CUST-WU7ZKQJE4L17	24	608,93
5	CUST-9WM83101QDTI	19	582,03
6	CUST-ZMAOVX8XYGJY	24	576,39
7	CUST-3K66CV0OHH7Q	35	571,44
8	CUST-CG23SXJDRNYR	39	531,09
9	CUST-D8IOFHVUF3Y	39	477,35
10	CUST-IHN1HQR17PYJ	7	463,73

# RÉSULTAT DES REQUÊTES SQL

Le chiffre d'affaire encaissé par employé

```
SELECT
  e.id_employe,
  e.name_employe,
  e.surname_employe,
  e.mail,
  ROUND(SUM(p.price), 2) AS chiffre_affaires_employe,
  ROUND(SUM(p.price) * 100.0 / total.total_ca, 2) AS part_pourcent
FROM sale_detail s
JOIN product p ON s.ean = p.ean
JOIN employe e ON s.id_employe = e.id_employe
CROSS JOIN (
  SELECT SUM(p2.price) AS total_ca
  FROM sale_detail s2
  JOIN product p2 ON s2.ean = p2.ean
) AS total
GROUP BY e.id_employe, e.name_employe, e.surname_employe, total.total_ca
ORDER BY part_pourcent DESC;
```

- Répartition du CA par employé
- Classement par performance
- Part en pourcentage : poids de chaque employé dans le CA global

Entrez une expression SQL pour filtrer les résultats (utilisez Ctrl+Espace)						
	A-Z id_employe	A-Z name_employe	A-Z surname_employe	A-Z mail	123 chiffre_affaires_employe	123 part_pourcent
1	f491076a1ff2d873e8ea809c11144542	Adélie	Boulet	aboulet@supersmartmarket.fr	7 818,82	2,75
2	e01e752175e05f00c8314ccb8da4c418	Eugène	Jacquier	ejacquier@supersmartmarket.fr	7 736,16	2,72
3	8d1001fbad3d2a60ff7530600ed5d55e	Charlène	Delisle	cdelisle@supersmartmarket.fr	6 995,14	2,46
4	6c1c3292c852c6c593b95cc146b00c0e	Pierre	Manoury	pmanoury@supersmartmarket.fr	6 616,46	2,33
5	a7ada0770091e838e3dcd45265282820	Tristan	Arsenault	tarsenault@supersmartmarket.fr	6 483,84	2,28
6	2477db17c02f512ebc4b20f01a7edb55	Auriane	Dufresne	adufresne@supersmartmarket.fr	6 361,22	2,24
7	528c733809cb51a3634befb260b5d243	Abelin	Dutertre	adutertre@supersmartmarket.fr	6 133,34	2,16
8	dd595f0f0b3400df2908f0be7723dad4	Arnaud	Lièvremon	alièvremon@supersmartmarket.fr	6 111,18	2,15
9	1c1d83678cc463b52366dae07cd14c8a	Pierre	Ange	pange@supersmartmarket.fr	6 112,37	2,15
10	c4f0909120cfde7086a3c1a56e96a015	Patricia	Rodier	prodier@supersmartmarket.fr	6 094,62	2,14
11	4853b03deab973a1a8e466025bce5b52	Solène	Deslys	sdeslys@supersmartmarket.fr	5 989,84	2,11
12	3084187cfabdb2a7de260f4387241ccc	Claudine	Gachet	cgachet@supersmartmarket.fr	5 936,36	2,09
13	f6cd8ba3485769b3ad9bab5b7725858e	Victoria	Baume	vbaume@supersmartmarket.fr	5 909,73	2,08
14	658e6925c1774feb2781a2888c9d4202	Bruno	Cazal	bcasal@supersmartmarket.fr	5 827,75	2,05
15	4efc10ff0f86c772d9949b490eb99ad3	Isaïe	Escoffier	iescoffier@supersmartmarket.fr	5 734,83	2,02
16	c495d16c3108a11ac8318269a85e9706	Émilienne	Blanchard	éblanchard@supersmartmarket.fr	5 705,28	2,01
17	342281771d02d2096972e38e78e7d6bd	Sabine	Giraud	sgiraud@supersmartmarket.fr	5 557,84	1,96
18	af969b090723911e6fd12c8da3b714ad	Ugo	Chevalier	uchevalier@supersmartmarket.fr	5 567,78	1,96
19	c12bfad8c0dae5bc1055c814069fe7	Anne	Donnet	adonnet@supersmartmarket.fr	5 513,15	1,94
20	951e47dfa4c5298382d1b7d750b0b6a2	Solange	Jacquet	sjacquet@supersmartmarket.fr	5 490,3	1,93
21	6d0cd9c04db768e471c92be478c88a32	Sacha	Pélessier	spélessier@supersmartmarket.fr	5 490,26	1,93
22	23e50fd96a8129e057a79ba0d5575c93	Joseph	Courbet	jcourbet@supersmartmarket.fr	5 450,27	1,92
23	fa836e3f5af72d24e079235332169ce	Emmanuel	Grosjean	egrosjean@supersmartmarket.fr	5 363,58	1,89
24	43710c31fab3c974c906d48477bd4a83	Victoria	Genet	vgenet@supersmartmarket.fr	5 345,16	1,88
25	ccebb4d1c9ccf593c4308cef110237	Yves	Grinda	ygrinda@supersmartmarket.fr	5 315,51	1,87
26	ddd71713c762283afb67295522774546	Aimée	Marchal	amarchal@supersmartmarket.fr	5 220,68	1,84
27	e2107857114936ec4800f52684dee59f	Maité	Rocheffort	mrocheffort@supersmartmarket.fr	5 187,41	1,82
28	89a9efaded9823c38bfd5f70d24e364f	Gabrielle	Granet	ggranet@supersmartmarket.fr	5 187,01	1,82
29	9f29007d63c46217deeb64efc81eba9e	Jean-Jacques	Auch	jauch@supersmartmarket.fr	5 146,52	1,81
30	afb0ad3a096ceea09662ee8692cd5711	Ambre	Besson	abesson@supersmartmarket.fr	5 128,45	1,8
31	e44eabec860498a79e675200bc8fa455	Amand	Coquelin	acoquelin@supersmartmarket.fr	5 013,37	1,76
32	6fa61d0ecae0b563fef18d36b2039c8e	Laure	Maret	lmaret@supersmartmarket.fr	4 941,48	1,74
33	38680e129f45c49322cb12303b7fb655	Vincent	Jacquier	vjacquier@supersmartmarket.fr	4 775,66	1,68
34	578b43483938010803dcd457753d923	Radegonde	Rémy	rrémy@supersmartmarket.fr	4 658,91	1,64
35	faa8bfc560cae9451761b38ff37e6ee9	Marie-Claire	Poincaré	mpoincaré@supersmartmarket.fr	4 638,77	1,63
36	0cb5ad89e8ae7afefa9c17277062db16	Gwendoline	Bonhomme	gbonhomme@supersmartmarket.fr	4 621,61	1,63
37	0859b70a9ca905a870bfccc5f8440ba1	Pierre	Beaumont	pbeaumont@supersmartmarket.fr	4 597,93	1,62
38	37c6a856b2e14217855d808fbc8e56a7	Christian	Vérany	cvérany@supersmartmarket.fr	4 613,51	1,62
39	d0921ae8a5b09f5d33e7c405943625d7	Liliane	Boissonade	lboissonade@supersmartmarket.fr	4 577,33	1,61
40	b5a8777821fdab1a20a4bc0e5f98dfa0	Nassima	Picard	npicard@supersmartmarket.fr	4 512,28	1,59
41	f93c7cff0b2eeb5ff91eba806f78297	Benjamin	Boutroux	bboutroux@supersmartmarket.fr	4 426,2	1,56
42	07c8b678f8e6f0cb04480ef9ebba10ce	Jean-Marc	Beauvau	jbeauvau@supersmartmarket.fr	4 410,01	1,55
43	6bf2742d580924ae05128018a3026aa4	Clara	Morin	cmorin@supersmartmarket.fr	4 395,59	1,55
44	803d880af14752773f6225d55f3bc5b4	Solange	Valluy	svalluy@supersmartmarket.fr	4 333,48	1,52
45	c36c4a024d43bba720a2482a25023378	Léon	Léon	lleon@supersmartmarket.fr	4 223,81	1,40

# COMPRÉHENSION DES LOGS



# LE DICTIONNAIRE DES DONNÉES - LOGS

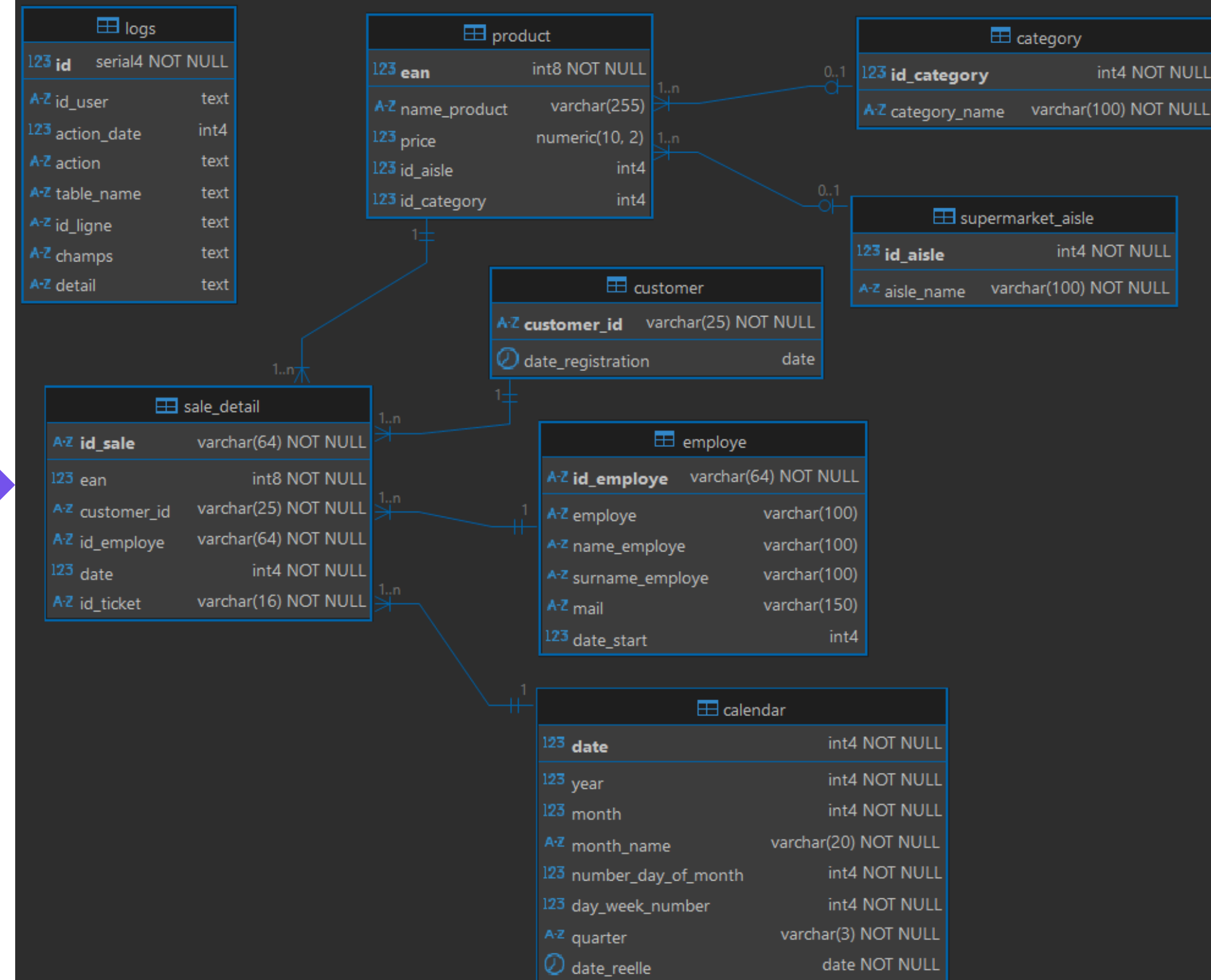
## Dictionnaire des données - Logs

Table : logs

Code	Signification	Type	Longueur	Nature	Règle de gestion
id	identifiant du log	SERIAL	-	Clé primaire	NOT NULL
id_user	identifiant de l'employé	TEXT	-	Attribut	-
action_date	date au format d'un nombre à 5 chiffres	INTEGER	-	Attribut	-
action	action (insert, update, delete) sur la table	TEXT	-	Attribut	-
table_name	nom de la table ou l'action a été effectué	TEXT	-	Attribut	-
id_ligne	identifiant employé, client ou produit	TEXT	-	Attribut	-
champs	nom de la colonne ou l'action a été effectué	TEXT	-	Attribut	-
detail	information sur la valeur de l'action	TEXT	-	Attribut	-

# LOGS

```
CREATE TABLE logs (
  id SERIAL PRIMARY KEY,
  id_user TEXT,
  action_date int4,
  action TEXT,
  table_name TEXT,
  id_ligne TEXT,
  champs TEXT,
  detail TEXT
);
```



# LOGS

## Lister les tables impactées par les logs

The screenshot shows a SQL IDE interface. At the top, a query editor contains the SQL command: `SELECT DISTINCT table_name FROM logs;`. Below the editor, a tab labeled 'logs 1' is active. The results are displayed in a table view, sorted by 'A-Z table\_name'. The table has two columns: an index and the table name. The results are as follows:

	A-Z table_name
1	Client
2	Employé
3	Produits
4	Ventes

La commande nous permet d'identifier rapidement les tables ayant été impactées par des **actions utilisateurs** :

Client  
Employé  
Produits  
Ventes





# LOGS

## Voir tous les types d'actions globalement

Aucune action **DELETE** ou **UPDATE** sur la table Ventes :  
Le C.A. n'a pas été diminué ou modifié par des suppressions ou mises à jour de ventes existantes.

En revanche, il y a **206885 INSERT** sur **Ventes**, ce qui peut artificiellement faire varier le C.A. d'un jour à l'autre.

**575 UPDATE de prix** ont été enregistrées dans la table **Produits**, ce qui pourrait fausser le chiffre d'affaires si les calculs ne tiennent pas compte des prix au moment de la vente.

SELECT table\_name, action, COUNT(\*) AS nb

FROM logs

GROUP BY table\_name, action

ORDER BY table\_name, action;

logs 1

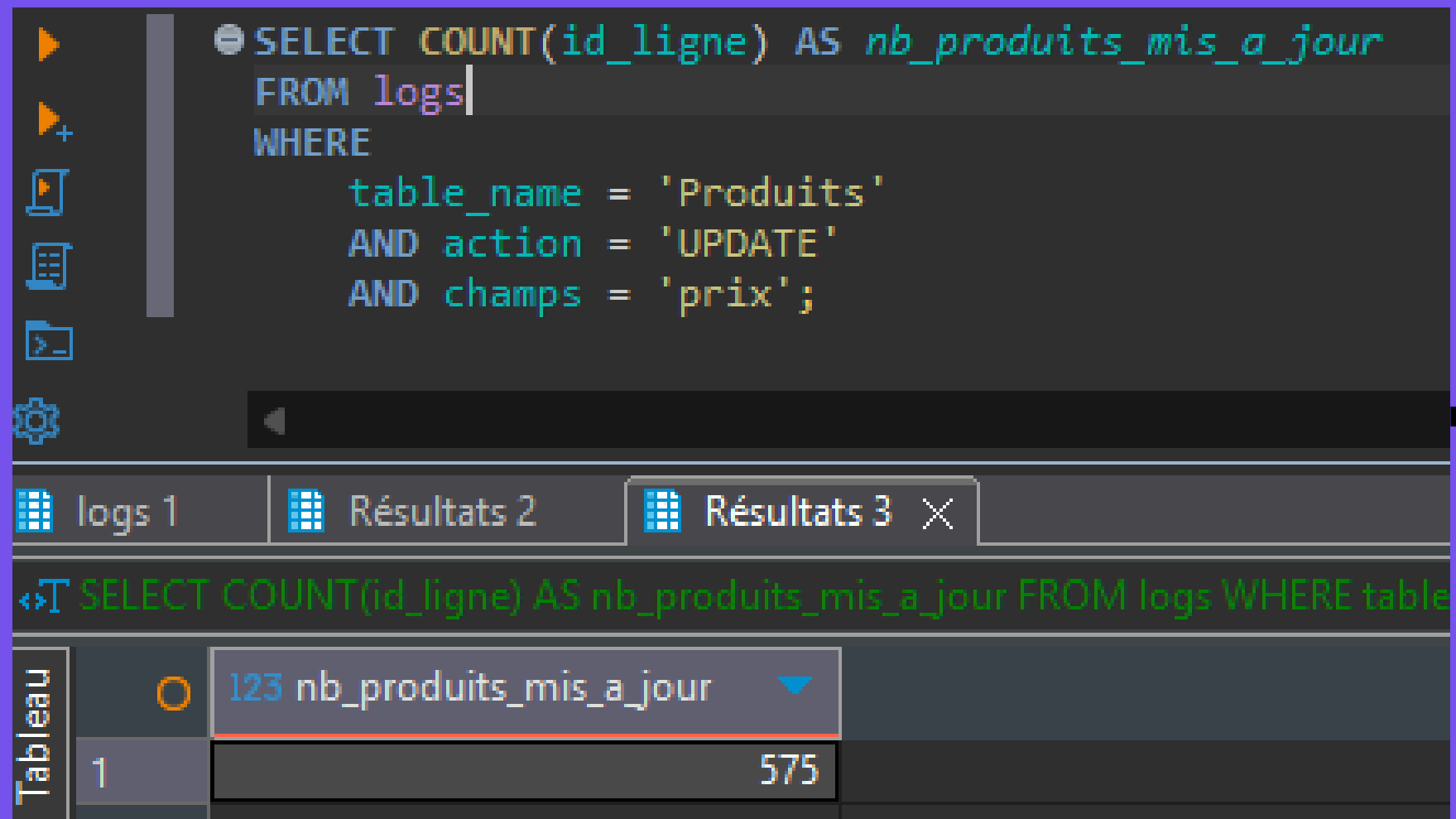
SELECT table\_name, action, COUNT(\*) AS nb FROM logs GROUP BY table\_name, action ORDER BY table\_name, action;

Tableau

Texte

	A-Z table_name	A-Z action	123 nb
1	Client	INSERT	20
2	Employé	DELETE	2
3	Employé	UPDATE	7
4	Produits	UPDATE	575
5	Ventes	INSERT	206 885

## Hypothèse n°1 les 575 UPDATES produits

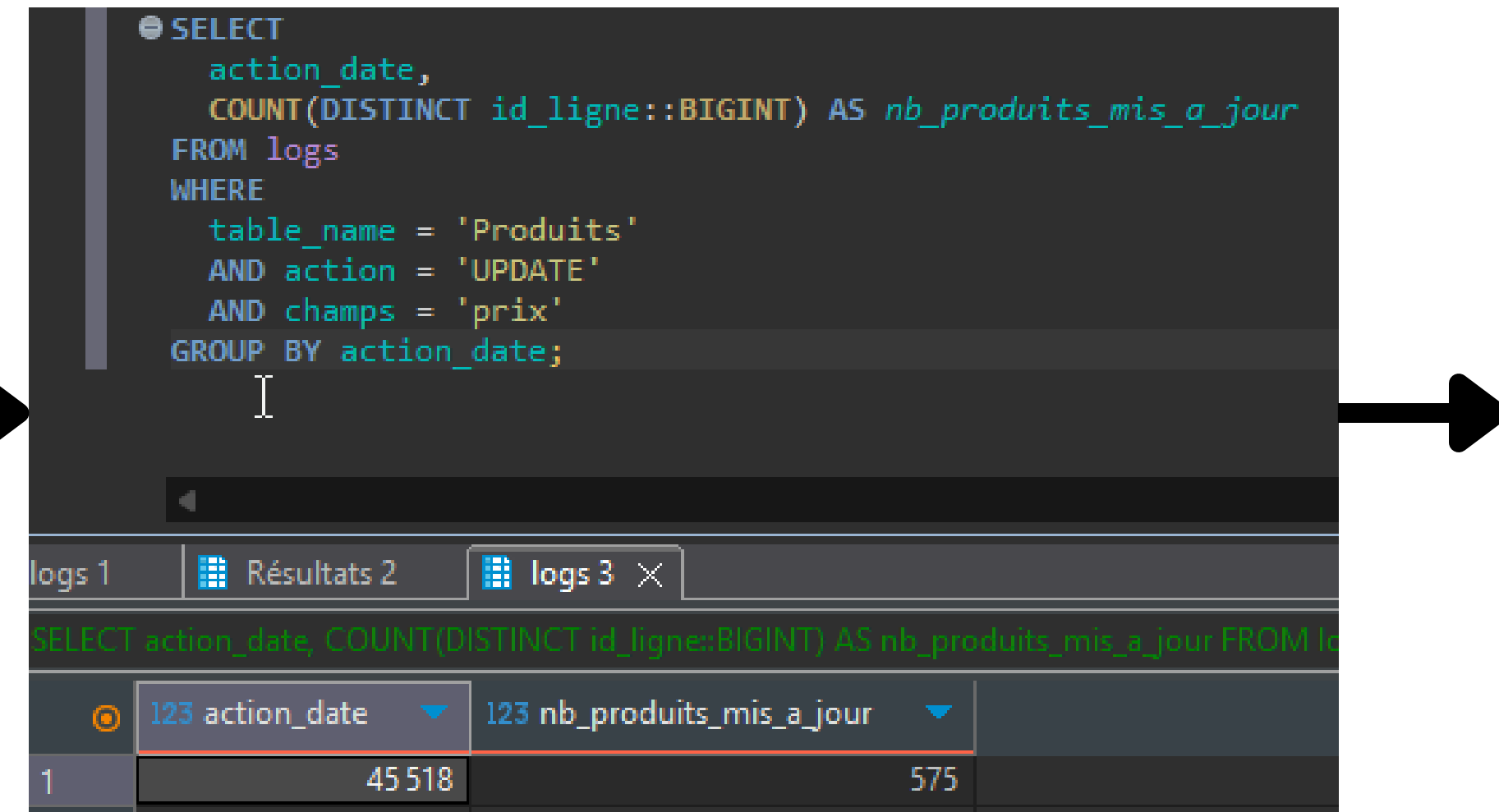


```
SELECT COUNT(id_ligne) AS nb_produits_mis_a_jour
FROM logs
WHERE
    table_name = 'Produits'
    AND action = 'UPDATE'
    AND champs = 'prix';
```

logs 1 | Résultats 2 | Résultats 3 X

SELECT COUNT(id\_ligne) AS nb\_produits\_mis\_a\_jour FROM logs WHERE table

Tableau	123 nb_produits_mis_a_jour
1	575



```
SELECT
    action_date,
    COUNT(DISTINCT id_ligne::BIGINT) AS nb_produits_mis_a_jour
FROM logs
WHERE
    table_name = 'Produits'
    AND action = 'UPDATE'
    AND champs = 'prix'
GROUP BY action_date;
```

logs 1 | Résultats 2 | logs 3 X

SELECT action\_date, COUNT(DISTINCT id\_ligne::BIGINT) AS nb\_produits\_mis\_a\_jour FROM logs

	123 action_date	123 nb_produits_mis_a_jour
1	45518	575

# Hypothèse n°1

## les 575 UPDATES produits

L'hypothèse selon laquelle les **575 updates** de prix ont eu lieu **après** le 14 août 2024 (**45518**) est fausse, car tous les UPDATE de prix observés dans les logs sont **datés précisément du 45518**, ce qui signifie que les modifications ont été effectuées le jour-même des ventes concernées, et non après.

On écarte cette hypothèse :

	A-Z table_name	A-Z action	123 nb
1	Client	INSERT	20
2	Employé	DELETE	2
3	Employé	UPDATE	7
4	Produits	UPDATE	575
5	Ventes	INSERT	206 885



## Hypothèse n°2

### les INSERT dans la table “ventes”

## Volume de ventes insérées dans les logs pour les 14 et 15 août 2024 (45 518 et 45 519)

L'analyse des logs a permis d'identifier le nombre de ventes enregistrées par **action (INSERT)** dans la table Ventes pour les deux dates clés :

- **39 998 ventes** ont été insérées le 14 août (**45518**)
- **1 379 ventes** supplémentaires ont été insérées le 15 août (**45519**), mais datées du 14 août

Ces **1 379 ventes** expliquent l'augmentation tardive du chiffre d'affaires du 14 août dans les tableaux de bord BI.

The screenshot shows a SQL query in a code editor and its results in Tableau. The SQL query is as follows:

```
SELECT
    l.action_date,
    c.date_reelle,
    COUNT(DISTINCT l.id_ligne) AS nb_ventes_log
FROM logs l
JOIN calendar c ON c.date = l.action_date
WHERE
    l.table_name = 'Ventes'
    AND l.action = 'INSERT'
    AND l.action_date IN (45518, 45519)
GROUP BY l.action_date, c.date_reelle
ORDER BY l.action_date;
```

The results are displayed in Tableau, showing a table with the following columns: action\_date, date\_reelle, and nb\_ventes\_log. The data is filtered for action\_date values 45518 and 45519.

	123 action_date	date_reelle	123 nb_ventes_log
1	45 518	2024-08-14	39 998
2	45 519	2024-08-15	1 379

# Hypothèse n°2

## les INSERT dans la table “ventes”

Ventes insérées rétroactivement pour le  
14 août 2024 (45 518)

Nombre de ventes concernées : 1379

Le chiffre d’affaires du 14 août a évolué car 1379 ventes ont été insérées le 15 août avec une date de vente fixée au 14 août. Cela explique la variation observée dans PowerBI.

Hypothèse n°2  
les INSERT dans la table “ventes”

Impact des insertions tardives sur le chiffre d'affaires  
du 14 août 2024 (45 518)

- 1 379 ventes ont été insérées le 15 août 2024, mais datées du 14 août 2024
- Ces ventes n'étaient pas présentes dans les exports ou tableaux de bord avant le 15 août
- Elles ont généré un chiffre d'affaires additionnel de 9 057,29 € pour le 14 août

La variation du chiffre d'affaires observée pour le 14 août s'explique par l'ajout rétroactif de ventes.

```
SELECT
    SUM(p.price) AS ca_retard_14_aout,
    COUNT(*) AS nb_ventes_retardees
FROM (
    SELECT DISTINCT l.id_ligne
    FROM logs l
    JOIN sale_detail sd ON l.id_ligne = sd.id_sale
    JOIN calendar c ON sd.date = c.date
    WHERE l.table_name = 'Ventes'
        AND l.action = 'INSERT'
        AND l.action_date = 45519      -- Insertion faite le 15 août
        AND c.date = 45518            -- Vente datée du 14 août
        AND l.action_date <> c.date
    ) AS ventes_retardees
JOIN sale_detail sd ON ventes_retardees.id_ligne = sd.id_sale
JOIN product p ON sd.ean = p.ean;
```

s 1

SUM(p.price) AS ca\_retard\_14\_aout, (

Entrez une expression SQL pour filtrer les résultats (

123 ca_retard_14_aout	123 nb_ventes_retardees
9057,29	1379



## Hypothèse n°2

### les INSERT dans la table “ventes”

### Impact des insertions tardives sur le chiffre d'affaires du 14 août 2024 (45 518)

- 45518 (14 août 2024) : 275186,59 €
- 45519 (15 août 2024) : 9 057,29 €

**Total cumulé : 284 243,88 €**

Ces montants confirment l'origine du chiffre d'affaires affiché dans PowerBI, dont une partie a été générée par des ventes insérées a posteriori.

```
SELECT
  ventes_uniques.action_date,
  c.date_reelle,
  COUNT(*) AS nb_ventes,
  SUM(p.price) AS ca_total
FROM (
  SELECT DISTINCT l.action_date, l.id_ligne
  FROM logs l
  WHERE l.table_name = 'Ventes'
  AND l.action = 'INSERT'
) AS ventes_uniques
JOIN sale_detail sd ON ventes_uniques.id_ligne = sd.id_sale
JOIN product p ON sd.ean = p.ean
JOIN calendar c ON ventes_uniques.action_date = c.date
GROUP BY ventes_uniques.action_date, c.date_reelle
ORDER BY ventes_uniques.action_date;
```

logs 1

Résultats 2

logs(+) 3

SELECT ventes\_uniques.action\_date, c.date\_reelle, COUNT(\*) AS nb\_ventes, SUM(p.price) AS ca\_to

	123 action_date	date_reelle	123 nb_ventes	123 ca_total
1	45 518	2024-08-14	39 998	275 186,59
2	45 519	2024-08-15	1 379	9 057,29



# EXPLICATIONS DES AJOUTS TARDIF DES VENTES

## Pistes possibles

- Un **manque de ressources** au niveau de la base de données (comme une **saturation** de la mémoire, du CPU ou des connexions disponibles) peut **ralentir, voire bloquer** temporairement, l'intégration des enregistrements, entraînant un **traitement différé** des ventes.
- un **conflit** entre certains champs (comme des **clés ou des contraintes** mal définies) aurait pu provoquer un bug au moment du chargement.
- une **erreur** dans le **pipeline** ou le **flux de données** pourrait avoir empêché l'injection en temps réel des ventes du 14 août, repoussant leur enregistrement au 15 août.

# SÉRIE DE MESURES CORRECTIVES

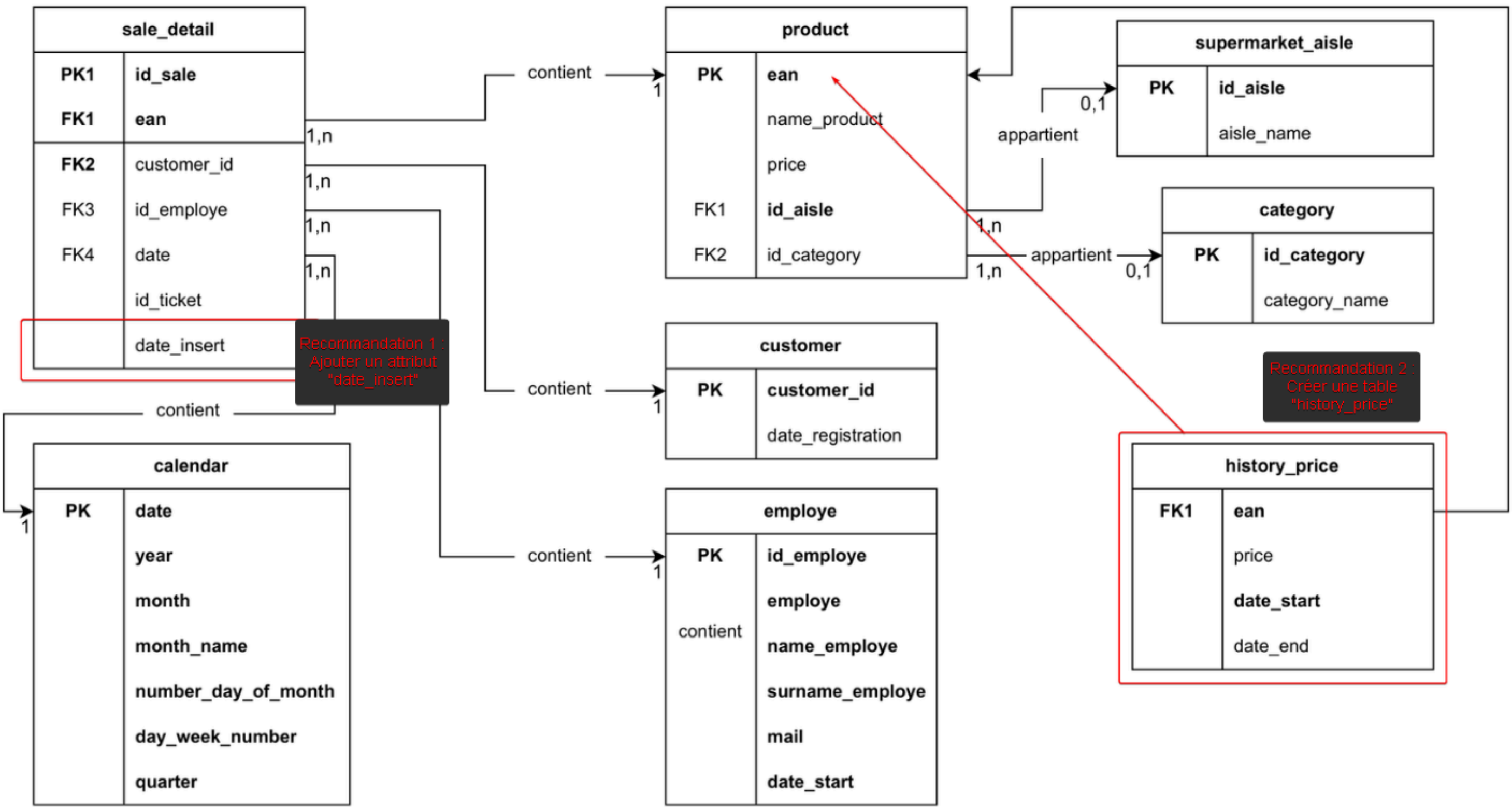
**Objectif** : Sécuriser la fiabilité des données et limiter les erreurs d'interprétation dans les tableaux de bord.

## Actions à mettre en place :

- Ajouter un champ "date\_insert" dans la table "sale\_detail"
- Mettre en place des alertes automatiques sur les variations anormales du chiffre d'affaires. Paramétrer une alerte dans Power BI (>10% de variation)
- Planifier un script hebdomadaire des logs systématiquement pour détecter les ajouts rétroactifs
- Historiser les prix des produits dans une table dédiée : "history\_price" (avec date de début et de fin)



# PROTOTYPE DE NOUVELLE BASE DE DONNÉES



### Points clés

Créer la table **“history\_price”** → prix au moment de la vente

Créer un attribut **“date\_insertion”** dans la table sale\_detail → traçabilité réelle

Script d'analyse hebdomadaire des logs systématiques → détection proactive des anomalies

Alertes automatiques sur les variations anormales du chiffre d'affaires dans Power BI

# MERCI

