MANUEL PROFESSIONNEL · ÉDITION 2026

Module M04 — Qualité, préparation et documentation des données

Phase 3 · Diagnostiquer, nettoyer, documenter : le socle entier comme matière

Analyse de données appliquée & Business Intelligence
8 partie(s) assemblée(s) · document de production

Table des matières

MODULE M04.C01 — LE DIAGNOSTIC EN DOUZE POINTS : SAVOIR CE QUI NE VA PAS AVANT DE TOUCHER À QUOI QUE CE SOIT 7

1. Objectifs du chapitre 7

2. Pourquoi cette notion est importante 7

3. Explication simple 8

4. Vocabulaire essentiel 8

5. Cours approfondi : les douze points 9

5.1 Point 1 — L'identité du fichier : taille, horodatage, encodage 9
5.2 Point 2 — La structure : lignes, colonnes, en-têtes 9
5.3 Point 3 — La complétude : qui manque, sur quelles lignes 9
5.4 Point 4 — L'unicité par la clé technique 10
5.5 Point 5 — L'unicité par la clé métier : là où le doublon se cache 10
5.6 Point 6 — La validité des types, cellule par cellule 10
5.7 Point 7 — La validité des domaines : valeurs autorisées 10
5.8 Point 8 — La cohérence interne : les égalités du fichier 11
5.9 Point 9 — La cohérence externe : les références qui cassent 11
5.10 Point 10 — L'actualité : la période annoncée contre la période trouvée 11
5.11 Point 11 — L'exactitude : la vérité est dehors 11
5.12 Point 12 — La traçabilité : et le rapport qu'on vous a donné, qui l'a mesuré ? 12
5.13 Les douze points, en une grille 12

6. Exemple concret : quarante-cinq minutes sur un fichier qu'on ne connaît pas 13

7. Démonstration pas à pas : produire la fiche sur le fichier de ventes 14

8. Erreurs fréquentes 15

9. Bonnes pratiques professionnelles 15

10. Exercice guidé — le trio qui décide du verdict (30 min, /10) 16

11. Exercices autonomes 16

12. Correction détaillée 17

13. Mini-projet M04.P1 — « La fiche de diagnostic du fichier reçu ce matin » (45 min) 17

14. Résumé du chapitre 18

15. À retenir 18

16. Évaluation formative (auto-correction, 8 min) 18

MODULE M04.C02 — LES ABSENCES : CINQ MÉCANISMES, CINQ TRAITEMENTS, ET LE COÛT DE CHAQUE CHOIX 20

1. Objectifs du chapitre 7

2. Pourquoi cette notion est importante 7

3. Explication simple 8

4. Vocabulaire essentiel 8

5. Cours approfondi : les cinq mécanismes, les cinq traitements 22

5.1 Mécanisme A — Le non-applicable : la valeur n'existe pas à l'état civil 22
5.2 Mécanisme B — Le non-collecté : on n'a jamais posé la question 22
5.3 Mécanisme C — Le non-répondu : la question a été posée, la réponse non 22

5.4 Mécanisme D — La perte à l'import ou à la conversion 22
5.5 Mécanisme E — L'absence déguisée en valeur tranquille 23
5.6 Traitements (1/2) : assumer, joindre 23
5.7 Traitements (2/2) : dériver, imputer, supprimer 23
5.8 Les cinq mécanismes et leurs chiffres, en une grille 24

6. Exemple concret : la phrase qui fait basculer un tableau de bord 25

7. Démonstration pas à pas : de l'œil au compteur, puis à la décision 25

8. Erreurs fréquentes 15

9. Bonnes pratiques professionnelles 15

10. Exercice guidé — trier quatre colonnes en quatre traitements (30 min, /10) 27

11. Exercices autonomes 16

12. Correction détaillée 17

13. Mini-projet M04.P2 — « Politique de valeurs manquantes » (1 h) 29

14. Résumé du chapitre 18

15. À retenir 18

16. Évaluation formative (auto-correction, 8 min) 18

MODULE M04.C03 — DOUBLONS : LA CLÉ QUI MENT, LA CLÉ QUI DIT VRAI, ET LE DICTIONNAIRE DE CORRESPONDANCES 32

1. Objectifs du chapitre 7

2. Pourquoi cette notion est importante 7

3. Explication simple 8

4. Vocabulaire essentiel 8

5. Cours approfondi 34
5.1 Les quatre niveaux où l'on peut chercher un doublon 34
5.2 Le faux doublon : ce que la répétition a de normal 34
5.3 Anatomie des 3 360 : ce que le fichier dit de son accident 34
5.4 La règle de garde : laquelle des deux lignes on conserve 35
5.5 Le registre des doublons, premier livrable de traçabilité 35
5.6 Les correspondances : quand deux écritures ne font qu'un objet 36
5.7 L'appariement des clients : la règle qui ne casse pas les homonymes 36
5.8 Les correspondances manquantes : 486 lignes et 372 clients 37

6. Exemple concret : le tableau croisé qui accuse un magasin 38

7. Démonstration pas à pas : trouver, décider, journaliser 38

8. Erreurs fréquentes 15

9. Bonnes pratiques professionnelles 15

10. Exercice guidé — reproduire le fichier propre, puis prouver que c'est le bon (35 min, /10) 40

11. Exercices autonomes 16

12. Correction détaillée 17

13. Mini-projet M04.P3 — « Le dossier de dédoublonnage » (1 h 30) 41

14. Résumé du chapitre 18

15. À retenir 18

16. Évaluation formative (auto-correction, 8 min) 18

MODULE M04.C04 — VALEURS ABERRANTES ET RÈGLES DE CONTRÔLE : UNE RÈGLE DOIT RENDRE UNE DÉCISION, PAS UN NOMBRE 44

1. Objectifs du chapitre 7

2. Pourquoi cette notion est importante 7

3. Explication simple 8

4. Vocabulaire essentiel 8

5. Cours approfondi 34

5.1 Les trois natures, comptées sur le fichier 46

5.2 Quatre familles de règles, et leur rendement mesuré 46

5.3 Ce que le fichier dit vraiment des 973 lignes 47

5.4 L'échelle de gravité, appliquée au socle 47

5.5 Une règle rend une décision : la forme à écrire 48

5.6 Calibrer un seuil : deux preuves, pas une habitude 48

5.7 Retourner une règle qui hurle : le cas des stocks 48

5.8 Les règles statistiques, remises à leur place 49

6. Exemple concret : la ligne à 75 662 kg 50

7. Démonstration pas à pas : la grille de contrôles en une heure 50

8. Erreurs fréquentes 15

9. Bonnes pratiques professionnelles 15

10. Exercice guidé — la règle de prix, de la distribution à la décision (35 min, /10) 51

11. Exercices autonomes 16

12. Correction détaillée 17

13. Mini-projet M04.P4 — « Le registre de contrôles du fichier de ventes » (1 h 30) 53

14. Résumé du chapitre 18

15. À retenir 18

16. Évaluation formative (auto-correction, 8 min) 18

MODULE M04.C05 — NORMALISER ET RENDRE COMPARABLE : CHANGER L'ÉCRITURE, JAMAIS LA VALEUR 56

1. Objectifs du chapitre 7

2. Pourquoi cette notion est importante 7

3. Explication simple 8

4. Vocabulaire essentiel 8

5. Cours approfondi 34

5.1 Trois objets, trois traitements — et ne pas les mélanger 58

5.2 Les montants texte : découper, puis prouver par une seconde voie 59

5.3 Mesurer le gain avant d'écrire la formule 60

5.4 Les dates : la norme est respectée, la réalité non 61

5.5 Unités et correspondances : ce qui demande un facteur, pas une formule 61

5.6 Importer sans rien casser : séparateur, décimales, encodage, en-tête 62

5.7 Les clés : normaliser pour que les jointures disent quelque chose 63

5.8 Ce qu'une normalisation ne doit jamais faire 63

6. Exemple concret : la cellule 16 638 FCFA 64

7. Démonstration pas à pas : le protocole de normalisation en dix étapes 65

8. Erreurs fréquentes 15

9. Bonnes pratiques professionnelles 15

10. Exercice guidé — reconvertir une colonne monétaire et le prouver (35 min, /10) 67

11. Exercices autonomes 16

12. Correction détaillée 17

13. Mini-projet M04.P5 — « Le protocole de normalisation » (1 h 30) 69

14. Résumé du chapitre 18

15. À retenir 18

16. Évaluation formative (auto-correction, 8 min) 18

MODULE M04.C06 — DOCUMENTER, NOMMER, PROTÉGER : LA DERNIÈRE ÉTAPE EST UNE ÉCRITURE, PAS UN CALCUL 71

1. Objectifs du chapitre 7

2. Pourquoi cette notion est importante 7

3. Explication simple 8

4. Vocabulaire essentiel 8

5. Cours approfondi 34

5.1 L’arborescence : trois zones, une règle d’écriture chacune 73

5.2 Le nommage : mesurer la convention avant de la proclamer 73

5.3 Le README de données : six sections, pas une de plus 74

5.4 Le journal des transformations : une ligne par opération, un compteur par ligne 75

5.5 Le contrôle final : un vert à 0 n’existe pas, et c’est une bonne nouvelle 75

5.6 Le périmètre des données personnelles : quatre fichiers, un risque mesuré 76

5.7 Anonymiser sans casser : l’ordre compte 77

6. Exemple concret : le journal du socle, complété 78

7. Démonstration pas à pas : la boîte à données en huit étapes 79

8. Erreurs fréquentes 15

9. Bonnes pratiques professionnelles 15

10. Exercice guidé — la boîte à données du socle (40 min, /10) 80

11. Exercices autonomes 16

12. Correction détaillée 17

13. Mini-projet M04.P6 — « La boîte à données » (1 h 30) 82

14. Résumé du chapitre 18

15. À retenir 18

16. Évaluation formative (auto-correction, 8 min) 18

PROJET M04.P — « LE DOSSIER DE DONNÉES DE LA QUINCAILLERIE DE KOUDOU-GOU » 84

1. Énoncé 84

Le matériel 84

Les six livrables 84

2. Ce qui est noté, ce qui est piégé 85

3. Barème et seuil 86

4. Correction des points d’attention 86

ÉVALUATION M04 — « QUALITÉ, PRÉPARATION ET DOCUMENTATION DES DONNÉES » 88

A · Questions de récupération (non notées) 88

B · Quiz (15 questions · 15 points) 89

Bloc 1 — Types et normalisation (Q1 à Q4) 89

Bloc 2 — Doublons et clés (Q5 à Q8) 89

Bloc 3 — Règles et valeurs aberrantes (Q9 à Q12) 89

Bloc 4 — Documentation et protection (Q13 à Q15) 90

Corrigé du quiz 90

C · Exercices pratiques (50 min · /20) 90

D · Étude de cas (60 min · /20, seuil 12) — « Le stock indiqué est-il crédible ? » 91

Lecture des résultats 92

Module M04.C01 — Le diagnostic en douze points : savoir ce qui ne va pas avant de toucher à quoi que ce soit

Outil de ce chapitre : le tableur et un éditeur de texte, sur le socle entier (243 360 lignes). Durée indicative : 6 h. Niveau : N2.

> **L'idée du chapitre.** Un fichier n'est pas « propre » ou « sale » : il est **connu** ou **inconnu**. Le diagnostic est l'heure que personne ne veut payer et que tout analyste fini paie deux fois — une fois pour établir l'état réel, une fois pour réparer un total qu'il croyait bon. Ce chapitre donne une grille de douze points, une durée (45 minutes sur un fichier jamais vu), et un livrable : la fiche de diagnostic. Il se termine sur une démonstration qui dérange : le rapport de défauts livré avec notre propre socle est faux sur quatre de ses huit lignes, parce qu'il a été écrit d'après l'intention et non d'après le fichier.
>
> **Base de travail — le socle entier, pas un extrait.** Quatre fichiers, tous lus sur disque : `01_socle_donnees/data/brut/ventes_brutes.csv` (28 516 920 octets, 243 360 lignes × 20 colonnes), `01_socle_donnees/data/brut/clients.csv` (23 912 lignes), `01_socle_donnees/data/brut/stocks_quotidiens.csv` (60 000 lignes), `01_socle_donnees/data/reference/verites_terrain.csv` (la cible : 240 000 lignes, 15 419 985 157 FCFA) — et, pour le point 12, `01_socle_donnees/data/reference/rapport_defauts.json`. Les chemins sont longs parce qu'ils sont réels : tout nombre de ce chapitre sort de ces fichiers, et le manuel les régénère à l'identique (graine `20260917`).

---

1. Objectifs du chapitre

À la fin de ce chapitre, vous saurez :

- **compter** avant de regarder : lignes, colonnes, clés distinctes, vides — et citer le dénominateur de chaque chiffre ;
- **classer** un défaut en six familles (complétude, unicité, validité, cohérence, exactitude, actualité) sans hésiter ;
- **exécuter** les douze points de la grille sur un fichier inconnu en moins de 45 minutes, au tableur seul ;
- **distinguer** un défaut qui dérange d'un défaut qui change la décision — en le chiffrant sur le total ;
- **rédiger** une fiche de diagnostic qu'un collègue peut rejouer sans vous demander aucune précision.

2. Pourquoi cette notion est importante

- Vous recevrez le fichier d'un autre. Pas le dictionnaire de données complet, pas la notice de fabrication, pas l'intention du collègue : un CSV, un export, un tableau collé dans un courriel. **Le premier acte utile est de le décrire, pas de le corriger.**
- Un nettoyage non diagnostiqué casse des totaux sans message. Sur notre socle, additionner la colonne des montants tels qu'ils arrivent rend `15 417 873 368` FCFA au lieu de `15 419 985 157` : **2 111 789 FCFA envolés, et aucune erreur affichée**. Le tableur a additionné ce qu'il savait additionner.

- L'inverse est vrai : « réparer » les montants texte sans traiter les lignes en trop rend `15 639 751 286` FCFA, soit **219 766 129 FCFA de trop**. Un défaut non vu dans le bon sens produit une erreur deux fois plus grosse que le défaut qu'on croyait corriger.
- Le métier appelle ça la due diligence sur la donnée : dans un audit, un appel d'offres, une reprise de dossier, c'est la pièce qui protège l'analyste. Un chiffre d'affaires contesté se défend avec une fiche de diagnostic, pas avec une capture d'écran.
- Enfin, le diagnostic est ce qui rend un travail **transmissible** : la fiche que vous laissez vaut plus que le fichier réparé que vous envoyez.

3. Explication simple

On ne répare pas une maison sans l'avoir inspectée, et l'inspection a un ordre : la charpente d'abord, la peinture ensuite. Le diagnostic d'un fichier, c'est la même chose, avec une règle unique : **à chaque question, un nombre et un dénominateur**.

Comparez deux façons de dire la même chose sur notre fichier de ventes :

Façon de dire	Ce que le lecteur peut en faire
« la colonne des montants a des soucis de texte »	rien — il faut vous rappeler pour savoir combien
« 3 421 montants sur 243 360 ne sont pas des nombres (1,4 %), et ils valent 2 111 789 FCFA »	tout — il sait si c'est grave, et peut revérifier

La première phrase est une impression. La seconde est une mesure. Douze points de mesure, exécutés dans l'ordre, et vous savez en trois quarts d'heure ce que le fichier contient — pas ce qu'il devrait contenir.

Le dernier des douze points est le plus facile à sauter, et le plus coûteux : **le rapport de qualité que vous avez reçu est-il mesuré, lui aussi ?** Si quelqu'un l'a écrit à la main depuis un tableau de spécifications, il décrit une intention. Nous allons le vérifier sur notre propre socle — et il est faux sur quatre lignes.

4. Vocabulaire essentiel

Français — English	Définition simple	Piège à éviter
Diagnostic de qualité — Data profiling	Décrire l'état réel d'un fichier par des compteurs, avant toute correction.	Confondre profiler et nettoyer : profiler ne change aucune cellule.
Complétude — Completeness	Une colonne est complète si chaque ligne concernée a une valeur.	Compter les vides sans dire quelles lignes sont concernées (une facture annulée n'a pas de date de paiement).
Unicité — Uniqueness	Une clé est unique si aucune valeur ne revient deux fois.	Croire que la clé primaire du fichier est la clé du réel : ici `id_vente` est impeccable et le doublon existe.
Clé métier — Business key	Combinaison de colonnes qui, dans le monde, identifie une ligne (ticket + produit + quantité).	L'oublier : 3 440 lignes passent entre les mailles de la clé technique.
Validité — Validity	Une valeur est valide si elle a le bon type et appartient au domaine attendu.	Valider le format et non le contenu : une date au bon format mais en 2026-12 alors que l'historique s'arrête en août.
Cohérence — Consistency	Deux expressions d'une même vérité ne se contredisent pas (`montant_ttc` = `montant_ht` + `montant_tva`).	Contrôler chaque colonne isolément et jamais les égalités entre elles.
Exactitude — Accuracy	La valeur correspond au fait réel — vérifié contre une source indépendante.	Prendre la somme du fichier pour sa propre exactitude : la vérité est dehors (ici, `verites_terrain.csv`).

Français — English	Définition simple	Piège à éviter
Actualité — Timeliness / Currency	La donnée est à jour par rapport à la période qu'elle prétend couvrir.	Confondre « date présente » et « date plausible ».
Conformité — Conformity	La donnée respecte la règle de gestion écrite (remise en proportion, retour à montant négatif).	Ne pas avoir écrit la règle : sans texte, aucune violation n'est démontrable.
Fiche de diagnostic — Data quality report	Le document qui consigne les douze points, leurs nombres, et la décision prise pour chacun.	La rédiger après coup, de mémoire : elle ne rejoue plus.

5. Cours approfondi : les douze points

Les points vont du plus structurel au plus sémantique. Pour chacun : **la question**, **la procédure**, et **la réponse mesurée sur le socle** — parce qu'une grille sans exemple chiffré ne s'apprend pas.

5.1 Point 1 — L'identité du fichier : taille, horodatage, encodage

Avant d'ouvrir : combien d'octets, quelle date de modification, quel encodage. Un fichier de 28 516 920 octets ne se diagnostique pas à l'œil ; un fichier dont l'horodatage est antérieur à la période annoncée est suspect avant même la première ligne. L'encodage se teste, il ne se devine pas : notre fichier tarif fournisseur (`01_socle_donnees/data/brut/tarif_fournisseur_peinture.csv`) **ne se décode pas** en UTF-8 — il échoue à l'octet 73 — et contient 95 octets hors ASCII : c'est un cp1252.

> **Définition.** Un **encodage** est la table qui associe un octet à un signe. `UTF-8` en utilise un à quatre par signe, `cp1252` exactement un. Lire un cp1252 comme de l'UTF-8 ne casse pas le fichier : ça remplace les accents par des caractères de substitution, et ça transforme « évacuation » en `Ã©vacuation` — un défaut introduit par la lecture, pas par le fichier.

5.2 Point 2 — La structure : lignes, colonnes, en-têtes

Le compte doit sortir du fichier, pas de son nom. Ici : **243 360 lignes de données, 20 colonnes**, dont les vingt noms sont `id_vente, id_ticket, date_vente, heure, id_magasin, id_vendeur, id_client, id_produit, quantite, prix_unitaire_ht, taux_remise, montant_ht, montant_tva, montant_ttc, mode_paiement, canal, est_retour, poids_kg, mois, annee` — et trois colonnes (`mois`, `annee`, `heure`) qui ne sont que des dérivées : leur présence doit être justifiée, sinon elles sont une source de divergence silencieuse entre l'original et le calcul.

5.3 Point 3 — La complétude : qui manque, sur quelles lignes

Un vide n'est un défaut que si la ligne est concernée. Sur le fichier de ventes, aucune colonne n'est vide au sens strict. Sur le fichier clients, **1 911 lignes sur 23 912 n'ont pas de ville**. La phrase correcte n'est pas « il manque la ville à 1 911 clients » mais « la ville est absente pour 1 911 clients sur 23 912, soit 8,0 % ; 0 ligne de `clients.csv` n'a de téléphone absent ». Deux colonnes, deux verdicts : ici le téléphone est systématiquement présent (23 912 lignes sur 23 912), l'e-mail presque jamais (absent sur 16 514 lignes).

> **Attention.** Compter les vides après l'import, c'est compter ce que l'outil a fabriqué. Un `0` recopié par une règle de remplissage automatique, une chaîne vide lue comme `NaN`, une cellule texte « N/A » : trois absences différentes, trois compteurs différents, un seul qui corresponde à la réalité du fichier.

5.4 Point 4 — L'unicité par la clé technique

`id_vente` compte 243 360 valeurs distinctes pour 243 360 lignes : **zéro doublon**. Cette ligne de la grille doit être écrite avec son résultat, même quand il est bon — c'est elle qui innocente la clé, et sans elle le lecteur ne sait pas si vous l'avez regardée.

5.5 Point 5 — L'unicité par la clé métier : là où le doublon se cache

Et pourtant, le fichier propre contient 240 000 lignes : **243 360 − 240 000 = 3 360 lignes** ont été retirées. Aucune n'est un copié-collé intégral (0 ligne entièrement dupliquée), aucune ne porte un `id_vente` déjà vu. Elles se voient seulement sur la clé du monde réel — un même ticket, un même produit, une même quantité : **3 440 lignes** sont en collision sur ce triplet, et 8 453 lignes participent à une collision ticket + produit, dont 7 640 avec la même quantité. C'est le point le plus instructif du diagnostic : **la clé primaire d'un fichier n'est pas la clé de ce que le fichier décrit.**

Ce qu'on a testé	Test	Résultat sur le socle
clé technique	`id_vente` distincts ÷ lignes	243 360 ÷ 243 360 → 0 doublon
ligne intégralement identique	égalité colonne à colonne	0 ligne
clé métier (ticket + produit + quantité)	collision sur le triplet	3 440 lignes
écart à corriger	brut − propre	3 360 lignes

> **Définition.** Une **clé candidate** est un jeu de colonnes dont les valeurs identifient une ligne et une seule. `id_vente` en est une ; le triplet ticket + produit + quantité en est une autre, **incompatible** avec la première : quand deux clés candidates se contredisent, l'une des deux n'est pas une clé, et c'est presque toujours celle que le fichier fournit déjà.
>
> **Dans les faits.** Sur un poste de saisie, la même colonne a trois visages selon la machine : `56658` en nombre, `56 658` en texte à cause du séparateur de milliers, `56658` avec une espace venue d'un copier-coller. Le point 6 les compte ensemble ; le C05 les traite séparément, parce qu'ils ne se réparent pas de la même main.

5.6 Point 6 — La validité des types, cellule par cellule

La question n'est pas « de quel type est la colonne » mais « combien de cellules tiennent leur type ». Réponse : **3 421 montants `montant_ttc` sur 243 360 ne sont pas des nombres** ; `montant_ht`, `quantite`, `prix_unitaire_ht` et `poids_kg` sont numériques sur 100 % des lignes. Un seul défaut de type, sur une seule colonne, sur 1,4 % des lignes : c'est exactement le profil d'un accident d'import, pas d'une saisie désordonnée — et cela se traite donc en amont (C05), pas ligne à ligne.

5.7 Point 7 — La validité des domaines : valeurs autorisées

Un domaine est une liste, et une liste se compare. `mode_paiement` a 5 modalités (Chèque, Créance 30 j, Espèces, Mobile Money, Virement), `canal` en a 3 (Livraison, Magasin, Téléphone) : dans les deux cas, **aucune valeur hors liste**. Le domaine est aussi numérique : `remise` est définie comme une proportion entre 0 et 1 ; or **973 lignes portent une valeur supérieure à 1**, avec un maximum lu à 30 — ce sont des pourcentages saisis en points (30 = 30 %). Mais attention à la conclusion hâtive : divisées par 100, ces valeurs ne réparent rien — **258 lignes** sortiraient encore du domaine contractuel (remise maximale admise : 0,25), et les montants, eux, restent cohérents. La colonne `taux_remise` n'est pas une proportion mal échelonnée : sur ces lignes, elle ne décrit plus le prix payé.

5.8 Point 8 — La cohérence interne : les égalités du fichier

Trois égalités découlent des règles de gestion publiées dans la notice du socle : `montant_ht = quantite × prix × (1 − remise)`, `montant_ttc = montant_ht + montant_tva`, et « une ligne marquée `est_retour = 1` a un montant négatif ». Mesure : **973 lignes** violent la première — exactement les 973 lignes à remise aberrante —, **0** la deuxième (la TVA rend 18 % du montant HT sur les 243 360 lignes, au franc près), et la troisième : sur 2 848 retours signalés, 2 801 ont un montant négatif et **47** n'ont pas de montant du tout (c'est le point 6 qui les a happés). Le détail important est au § 5.7 : les 973 lignes incohérentes ont un `montant_ht` juste et une `taux_remise` faux — le défaut ne casse donc **aucun** total, il casse uniquement les analyses de remise.

> **Définition.** Un **contrôle croisé** est une vérification qui porte sur deux colonnes au lieu d'une. Il attrape les défauts qu'aucun contrôle unitaire ne voit : une remise à 30 est « une valeur plausible » si on la regarde seule, et une impossibilité dès qu'on la multiplie par le prix.

5.9 Point 9 — La cohérence externe : les références qui cassent

Une clé étrangère doit pointer quelque part : **486 lignes de vente** portent un `id_client` absent du fichier clients, sur **372 clients distincts** inconnus. C'est le point dont le coût est le plus mal estimé : 486 lignes ne disparaissent pas, elles disparaissent du rapprochement. Toute jointure avec les clients perd ces lignes sans message, et tout tableau croisé construit sur la jointure affiche un chiffre d'affaires minoré — sans que la moitié des lignes concernées aient le moindre défaut apparent. (Le nombre de tickets distincts — 146 161 — sert ici de contrôle d'ordre de grandeur, pas de verdict.)

> **Attention.** Un orphelin n'est pas une ligne à supprimer. Les 486 lignes concernées sont des ventes réelles ; ce qui manque, c'est la référence. Effacer la ligne pour réparer la jointure, c'est effacer un chiffre d'affaires. On crée un client « à créer », ou on rattache à un compte générique explicitement nommé — et on l'écrit dans le journal des transformations (C06).

5.10 Point 10 — L'actualité : la période annoncée contre la période trouvée

La règle du socle est écrite : l'historique court de 2023-01-01 à 2026-08-31, et l'année 2026 est partielle. Mesure : la date maximale du fichier est **2026-12-06** et **51 lignes** dépassent le 31/08/2026. Cinquante lignes sur 243 360 sont invisibles à l'œil, mais suffisent à faire dire à une comparaison annuelle ce qu'elle ne doit pas dire : « 2026 dépasse 2025 ». Le contrôle d'actualité n'est pas une question de propreté, c'est une question de **date de coupure déclarée** — et cette date doit figurer sur la fiche.

5.11 Point 11 — L'exactitude : la vérité est dehors

Les dix premiers points interrogent le fichier sur lui-même. Le onzième le confronte à une source indépendante. Ici, `verites_terrain.csv` affirme 240 000 lignes et 15 419 985 157 FCFA pour l'ensemble des ventes. Vérification, dans l'ordre où le nettoyage la rend vraie :

Contrôle	Valeur	Écart à la cible
somme naïve des `montant_ttc` du brut	15 417 873 368	− 2 111 789
somme après réparation des textes, sans dédoublonner	15 639 751 286	+ 219 766 129
cible (`verites_terrain.csv`)	15 419 985 157	—

Retenez la forme du résultat, pas les nombres : **le total n'est pas un indicateur de propreté, c'est un indicateur d'écart**, et il ne se rétablit que si les deux défauts sont traités. Additionner sans réparer sous-évalue ; réparer sans dédoublonner sur-évalue de cent fois plus. C'est la démonstration que « ça a l'air bon » ne suffit pas.

5.12 Point 12 — La traçabilité : et le rapport qu'on vous a donné, qui l'a mesuré ?

Le socle est livré avec un `rapport_defauts.json` — la fiche que l'équipe de production a écrite pour le corrigé. Confrontons-la à la mesure :

Défaut annoncé	Annoncé	Mesuré sur le fichier	Verdict
montants en texte	3 360	3 421	sous-estimé de 61
dates au format mixte	2 640	0	inexistant dans ce fichier
lignes en double	3 360	3 360 (clé métier)	juste de nombre, faux de mécanisme
ville manquante aux clients	1 880	1 911	sous-estimé de 31
catégories de produits hétéroclites	61	44	sur-estimé de 17

Cinq lignes, trois erreurs, et un nombre qui tombe juste pour la mauvaise raison. La plus instructive est le 61 : il ne compte pas des lignes à corriger, il compte le **recouvrement** de deux défauts — 61 lignes sur les 3 421 sont aussi des lignes en trop. Un chiffre vrai, sorti d'un tableau de plan, rangé sous un mauvais libellé : exactement le défaut qu'aucun contrôle automatique ne signale. Ce n'est pas de la négligence : c'est ce qui arrive quand on recopie un **plan d'injection** au lieu de mesurer le fichier obtenu. Un défaut injecté au générateur peut se trouver compensé, annulé ou étendu par le reste de la chaîne.

> **Définition.** On parle d'**intégrité référentielle** quand toute valeur d'une colonne de référence pointe une ligne existante de la table référencée. 486 lignes de notre fichier la violent ; comme une jointure perd silencieusement ce qu'elle ne retrouve pas, la violation ne se voit qu'au total, jamais à la cellule.
>
> **Conseil professionnel.** Notez sur la fiche de diagnostic le nom du fichier, son empreinte de taille, la date de lecture, et la commande ou la formule qui a produit chaque nombre. Un diagnostic sans mode d'emploi est une opinion datée ; avec, c'est une preuve rejouable — et la différence se voit le jour où l'on vous conteste un chiffre.
>
> **À retenir.** Les douze points ne se valent pas, et leur ordre non plus. Compter les lignes avant les vides évite de diviser par le mauvais dénominateur ; tester la clé technique avant la clé métier évite de conclure « propre » trop vite ; garder l'exactitude en avant-dernier évite de réparer dix défauts pour un total qui reste faux. Un ordre, c'est ce qui rend un diagnostic reproductible par quelqu'un d'autre — et rejouable six mois plus tard.

5.13 Les douze points, en une grille

#	Question	Ce qu'on compte	Ici
1	Identité	octets, date, encodage	28 516 920 o, cp1252 testé
2	Structure	lignes × colonnes	243 360 × 20
3	Complétude	vides par colonne	0 ici ; 1 911 villes absentes (clients)
4	Unicité (technique)	distincts ÷ lignes	243 360 ÷ 243 360
5	Unicité (métier)	collisions sur la clé réelle	3 440 lignes, 3 360 à retirer
6	Types	cellules hors type	3 421 montants texte

#	Question	Ce qu’on compte	Ici
7	Domaines	valeurs hors liste	0 hors liste ; 973 remises > 1
8	Cohérence interne	égalités entre colonnes	973 violations
9	Cohérence externe	clés orphelines	486 lignes, 372 clients
10	Actualité	bornes vs période annoncée	51 dates hors période, max 2026-12-06
11	Exactitude	total vs source indépendante	écart − 2 111 789 / + 219 766 129
12	Traçabilité	le rapport reçu est-il mesuré ?	3 lignes fausses sur 5 testées

Diagnostic de qualité — les douze points, dans l’ordre d’exécution

Chaque case : la question, le nombre, le dénominateur. Filet rouge = le point change le verdict.

1 identité
structure
28 516 920 o · cp1252 testé

2 structure
structure
243 360 lignes · 20 colonnes

3 complétude
structure
0 vide au brut · 1 911 villes absentes

4 unicité (technique)
clé
243 360 clés distinctes · 0 doublon

5 unicité (métier)
clé
3 440 collisions · 3 360 à retirer

6 types
valeur
3 421 montants en texte

7 domaines
valeur
973 remises > 1 · max lu 30

8 cohérence interne
valeur
973 lignes · les mêmes qu’au point 7

9 cohérence externe
valeur
486 lignes · 372 clients inconnus

10 actualité
temps
51 dates après le 31/08/2026

11 exactitude
décision
écart -2 111 789 puis +219 766 129

12 traçabilité
décision
rapport fourni faux sur 3 lignes sur 5

Trois points décident du nettoyage :
5 · la clé métier (le doublon invisible à id_vente) · 6 · les types (les montants texte)
11 · l’exactitude (le total contre source extérieure). Sur le socle : 3 421 + 3 360 - 61 = 6 720 lignes à traiter, soit 2,8 % du fichier. Les neuf autres points changent la confiance.

6. Exemple concret : quarante-cinq minutes sur un fichier qu’on ne connaît pas

Le contexte est celui d’un lundi matin : la direction commerciale envoie `ventes_brutes.csv`, trois ans d’historique, sans autre commentaire que « tout est dedans ». Minutage réel d’un diagnostic complet, tableur ouvert :

Minutes	Acte	Sortie
0-3	taille, horodatage, ouverture de la première ligne	cp1252, 28 516 920 octets, séparateur `;` ou `,` confirmé
3-8	import en forçant les types de chaque colonne	243 360 lignes × 20 colonnes dans la feuille
8-14	comptes : lignes, clés distinctes, vides	points 2, 3, 4
14-22	types et domaines (`NB` contre `NBVAL`, listes de valeurs)	points 6, 7
22-30	collisions de clé métier (tableau croisé sur le triplet)	point 5
30-36	égalités internes et jointures externes	points 8, 9
36-40	bornes de dates, total contre source indépendante	points 10, 11

Minutes	Acte	Sortie
40-45	fiche de diagnostic : douze lignes, un nombre et une décision par ligne	livrable

Trente-six minutes dans M01-M03 s'appelaient « nettoyer ». Ici, quarante-cinq minutes ne changent **aucune cellule** : à la fin, le fichier est identique et vous êtes capable de dire ce qui ne va pas, de combien, et ce que ça coûte sur le total. C'est le renversement que ce module attend de vous.

7. Démonstration pas à pas : produire la fiche sur le fichier de ventes

On écrit les douze contrôles comme formules de tableur, sur la feuille `Ventes` (243 360 lignes, sous la limite de `1 048 576` lignes d'une feuille — si votre fichier dépasse la limite, voir § 5.1 et le point 1 : c'est déjà une information de diagnostic, et M05 est là pour ça).

1. **Feuille de contrôle.** Une feuille `DIAGNOSTIC`, trois colonnes : point, nombre, décision. Douze lignes.
2. **Structure.** `=NBVAL(Ventes!A:A)-1` → 243 360 lignes de données ; `=NBVAL(Ventes!$A$1:$T$1)` → 20 colonnes.
3. **Complétude, par colonne.** `=NBVAL(Ventes!N2:N243361)-NB(Ventes!N2:N243361)` : cellules remplies moins cellules comptées comme nombres ; sur `montant_ttc` (colonne N), la soustraction rend 3 421. Consignez le dénominateur à côté : sans lui, le nombre ne dit rien à un repreneur.
4. **Unicité technique.** `=SOMMEPROD(1/NB.SI(Ventes!A:A;Ventes!A2:A243361))` rend le nombre de valeurs distinctes d'`id_vente` : 243 360. Si `NBVAL` de la colonne égale ce nombre, la clé tient.
5. **Unicité métier.** Colonne utilitaire `=Ventes!B2&"|"&Ventes!H2&"|"&Ventes!I2` (ticket | produit | quantité), puis `=NB.SI(utilitaire;utilitaire)` compté en colonne 2 : les lignes où le compte dépasse 1 sont les 3 440 collisions. Un tableau croisé sur cette colonne (valeur : nombre d'articles) donne la même réponse en dix secondes, et c'est la voie que je recommande — le `SOMMEPROD` est là pour qu'on ne dépende pas du croisé.
6. **Types.** `=NB(Ventes!N2:N243361)` contre `=NBVAL(…)` : `239 939` contre 243 360. L'écart est le nombre de textes.
7. **Domaines.** Sur `taux_remise` : `=NB.SI.ENS(Ventes!K2:K243361;">1")` → 973. Sur les colonnes déclaratives, un tri à valeurs uniques confronte la liste obtenue à la liste attendue (5, puis 3 modalités).
8. **Cohérence interne.** Colonne utilitaire `=ARRONDI(Ventes!I2*Ventes!J2*(1-Ventes!K2);0)` et `=SI(ABS(…-Ventes!L2)>1;1;0)` sommée → 973. La même population que le point 7 : c'est la preuve que les deux défauts n'en font qu'un.
9. **Cohérence externe.** `=ESTNUM(RECHERCHEX(Ventes!G2;clients!A:A;clients!A:A;"absent"))` recopiée, puis comptée : 486 lignes, 372 `id_client` distincts en erreur.
10. **Actualité.** `=MAX(Ventes!C2:C243361)` et `=NB.SI(Ventes!C2:C243361;">"&DATE(2026;8;31))` → 2026-12-06, 51.
11. **Exactitude.** `=SOMME(Ventes!N2:N243361)` → 15 417 873 368, contre la cible 15 419 985 157 : l'écart se reporte en clair dans la colonne décision, avec le signe.
12. **Traçabilité.** Une cellule, une phrase : « valeurs mesurées le [date] sur [fichier], par les formules de cette feuille ; source de comparaison : [fichier ou personne] ».

> **Boîte à outils du chapitre.** Les douze points tiennent dans une feuille-modèle : point, formule, nombre, dénominateur, décision, date de lecture. Cette dernière colonne est celle qui fait la différence entre un document et une preuve : elle permet de dire, un mois plus tard, si la mesure a été faite avant ou après le fichier que vous avez reçu.

8. Erreurs fréquentes

Symptôme visible	Cause réelle	Correction
« Le fichier est propre, `id_vente` n'a aucun doublon »	seule la clé technique a été testée	ajouter le point 5 : collisions sur la clé métier (3 440 lignes ici)
Le total obtenu après nettoyage dépasse la cible	un défaut traité en a révélé un autre (les textes réparés sans retirer les doublons)	ne jamais valider un nettoyage sur un seul écart : contrôler les deux signes, et la cible
Le rapport de défauts fourni ne correspond pas au fichier	il a été écrit depuis un plan, pas depuis une mesure	le traiter comme une hypothèse : le point 12 teste le rapport, pas seulement le fichier
Les pourcentages de vide varient d'une lecture à l'autre	les vides dépendent de l'import (0 remplacé, `NaN` fabriqué)	compter sur le fichier texte, avant import, puis comparer avec le compte après import
Un tri « à la main » sur la colonne des dates fait disparaître 51 lignes	elles étaient invalides pour l'outil mais lisibles	la date se contrôle par `NB` sur la colonne, pas par l'œil : point 6
Le diagnostic est fini « en dix minutes »	six des douze points ont été faits d'intuition	la fiche impose une ligne par point, même vide : un point non mesuré se voit

9. Bonnes pratiques professionnelles

- **Une ligne de fiche par point, sans exception.** Un point qui ne trouve rien s'écrit « 0 sur 243 360 » — c'est un résultat, et c'est le seul moyen de prouver que vous l'avez cherché.
- **Toujours le dénominateur.** « 973 » ne veut rien dire ; « 973 sur 243 360 » veut dire tout.
- **Compter avant d'importer, revérifier après.** Deux compteurs, une différence : c'est l'import qui a menti, pas vous.
- **Ne jamais réparer pendant le diagnostic.** La fiche est signée sur le fichier tel quel ; la réparation s'écrit dans un journal séparé (C06), avec l'avant et l'après.
- **Écrire la règle de gestion à côté du contrôle.** « remise ∈ [0 ; 1] » doit être une phrase d'un document, sinon personne ne peut vous reprocher d'avoir signalé 973 lignes — ni vous défendre de les avoir ignorées.

> **Attention.** Un contrôle croisé est la première victime d'un changement de règle de gestion : la TVA passe de 18 % à 20 %, l'égalité est restée écrite avec l'ancien taux quelque part, et le contrôle se met à hurler sur des lignes parfaitement saines. La règle vit dans le dictionnaire (C06), la formule l'applique — jamais l'inverse. - **Conserver le fichier de départ, en lecture seule.** Un diagnostic rejouable suppose que la source n'a pas bougé. - **Faire relire la fiche par quelqu'un qui n'a pas le fichier.** S'il ne peut pas retrouver ne serait-ce qu'un nombre, la fiche est mauvaise — c'est le seul test d'utilité qui compte.

10. Exercice guidé — le trio qui décide du verdict (30 min, /10)

On traite ensemble les trois points qui, sur ce fichier, changent la conclusion. Vous faites, je corrige au fil de l'eau ; chaque étape finit par un nombre à reporter sur la fiche.

Étape 1. Combien de lignes de vente portent un montant texte ? `NBVAL(N2:N243361) − NB(N2:N243361)` → **3 421**. Si vous trouvez 3 360, vous avez lu le rapport fourni au lieu du fichier — retournez au § 5.12, c'est exactement l'erreur visée.

Étape 2. Combien de lignes doit-on retirer pour rejoindre le fichier propre ? 243 360 − 240 000 → **3 360**. Notez que ce n'est pas le même nombre, et qu'il n'y a aucun paradoxe : un défaut de type, un défaut de doublon.

Étape 3. Les deux défauts se recouvrent-ils ? Comptez les lignes qui sont à la fois en texte sur le montant et absentes du fichier propre → **61 lignes**. C'est peu, et c'est le chiffre qui décide de l'ordre des opérations : ces 61 lignes doivent passer par la réparation de type avant le dédoublonnage, sinon on retire l'une des deux et on laisse l'autre avec un montant vide — le total ne se réconcilie plus, et plus rien ne se prouve ligne à ligne.

Étape 4. Vérifiez l'addition : 3 421 + 3 360 − 61 = **6 720** lignes concernées par au moins un des deux défauts, soit 2,8 % du fichier. Écrivez cette ligne dans la fiche : c'est le périmètre du nettoyage, et il vous évitera d'annoncer « 3 421 lignes à corriger » à un lecteur qui en trouvera 6 720.

Étape 5. Décision. Les 6 720 lignes valent 2 111 789 FCFA côté texte et 219 766 129 FCFA côté doublons. Reportez la décision : réparer les textes par `montant_ht + montant_tva`, puis dédoublonner sur la clé métier, puis recontrôler contre `verites_terrain.csv`. **Note /10** : 2 points par nombre juste et situé (point, formule, dénominateur), 2 points pour le recouvrement des 61 lignes, 4 points pour la décision. Un nombre sans dénominateur vaut zéro.

11. Exercices autonomes

Exercice 1.1 (★) — Le trio structurel (8 min). Refaites les points 2, 4 et 6 sur `clients.csv` (23 912 lignes) et écrivez les trois lignes de fiche. Quel est le seul point de la grille qui, ici, n'a pas de réponse nulle ?

Exercice 1.2 (★★) — La clé de dédoublonnage (12 min). Le point 5, appliqué aux clients : comptez les lignes en trop si la clé candidate est le nom normalisé (lettres minuscules, sans accent ni espace) ; comptez-les si elle est nom normalisé + seuls chiffres du téléphone. Donnez les deux nombres et expliquez lequel est utilisable.

Exercice 1.3 (★★) — Le seuil qui ne contrôle rien (15 min). Les stocks : sur les 60 000 lignes de `stocks_quotidiens.csv`, comptez les quantités physiques négatives, les lignes sous le seuil d'alerte, et l'écart théorique − physique maximal. Les deux premiers nombres sont identiques — que faut-il en conclure sur la qualité de la colonne `seuil_alerte` ?

Exercice 1.4 (★★) — Trois nombres pour un fichier (12 min). Le tarif fournisseur : donnez les trois nombres qui, à eux seuls, décrivent ce fichier en un paragraphe (lignes de bruit avant l'en-tête, lignes de données, prix dont le séparateur décimal n'est pas un point) et indiquez lequel se corrige à la lecture, lequel à l'écriture.

Exercice 1.5 (★★★) — Rendre un contrôle rejouable (20 min). Le point 12, retourné : écrivez la ligne de fiche qui aurait attrapé l'erreur des dates au format mixte (2 640 annoncées) sans ouvrir le fichier d'à côté — c'est-à-dire en rendant la formule rejouable. Vous ajouterez aussi la ligne qui aurait évité l'erreur sur les catégories de produits (61 annoncées, 44 mesurées) : celle qui oblige à citer **la liste attendue** avec son nombre de modalités.

12. Correction détaillée

Exercice 1.1. Point 2 : 23 912 lignes × 12 colonnes. Point 4 : 23 912 `id_client` distincts → 0 doublon technique. Point 6 : le téléphone est numérique-comme-texte partout, la ville absente sur **1 911** lignes, l'e-mail sur **16 514**. Le seul point non nul est le point 3 (complétude) — et c'est aussi le seul qui ne concerne pas le fichier de ventes. Résultat intermédiaire à vérifier : 1 911 + 22 001 = 23 912.

Exercice 1.2. Sur le nom normalisé seul, 23 912 − 12 342 = **11 570** lignes « en trop » : la règle est inutilisable, elle fusionnerait des clients réellement distincts (le jeu contient des noms génériques comme `Menage 00002`). Sur nom normalisé + chiffres du téléphone, l'excédent tombe à **412** — exactement le nombre de lignes retirées pour passer de 23 912 à 23 500 dans le fichier propre. Conclusion à écrire : **une clé de dédoublonnage se valide contre le résultat attendu, pas contre son élégance** ; et le téléphone est ce qui rend la clé utilisable, parce que son écriture varie d'une ligne à l'autre (le jumeau a perdu ses espaces) alors que ses chiffres restent ceux du client — le nom `SARL K`, lui, est porté par 75 lignes du fichier.

Exercice 1.3. 1 800 lignes à quantité physique négative, 1 800 lignes sous le seuil d'alerte, écart maximal 532 unités. Les deux compteurs coïncident : la règle croisée est un identique — toute ligne négative est sous le seuil. On en conclut que `seuil_alerte` ne discrimine pas (il ne peut pas, si le signe suffit à déclencher) : le contrôle utile n'est donc pas « quantité < seuil » mais « quantité < 0 » d'une part, « écart |théorique − physique| > seuil » d'autre part. C'est l'exemple type du point 8 : une formule juste sur une donnée fausse produit un alarme permanent, et une alarme permanente ne contrôle rien.

Exercice 1.4. Trois nombres : **3** lignes de bruit avant l'en-tête (« export genere depuis… », la date d'édition, le nom du fournisseur), **38** lignes de données, **38** prix sur 38 écrits avec une virgule décimale. Le bruit se corrige à la lecture (forcer la ligne d'en-tête, ou supprimer les trois lignes dans le fichier d'origine — pas dans la feuille de travail) ; la virgule se corrige à l'écriture, c'est-à-dire dans le paramètre régional de l'import, sinon le nombre lu est 2 900 au lieu de 2 900,57 et la colonne entière devient du texte. Ajoutez le quatrième, qui n'était pas demandé et qui gêne davantage : **1** seule valeur de `conditionnement` pour 38 lignes — une colonne déclarative qui n'a qu'une modalité est une colonne qui ne dit rien.

Exercice 1.5. Pour les dates : la ligne de fiche doit contenir la formule, la colonne et le fichier — « `=NB.SI(date_vente;">"&DATE(2026;8;31))` = 51, mesuré le [date] sur `ventes_brutes.csv` » — de sorte que le relecteur qui refait le calcul trouve 51 et puisse écrire 0 s'il les trouve. Un contrôle non rejouable ne peut pas être contredit, donc ne prouve rien. Pour les catégories : la ligne attendue est « catégories hors des 7 de la liste de référence : 44 sur 154 références, liste attendue jointe en `DIAGNOSTIC!A20:A26` » — c'est-à-dire que **le domaine doit être dans la fiche**, sinon le chiffre de 61 (celui du plan) et celui de 44 (celui du fichier) se valent aux yeux du lecteur, et le seul qui compte disparaît.

13. Mini-projet M04.P1 — « La fiche de diagnostic du fichier reçu ce matin » (45 min)

Livrable unique : `diagnostic_ventes_brutes.md`, une fiche de douze lignes plus un entête. Contraintes, qui sont celles du métier :

- la fiche tient sur **une page** et commence par l'identité du fichier (nom, octets, date de lecture) ;
- chaque ligne des douze points porte : le nombre, le dénominateur, la formule ou l'outil, et **une décision** (ignorer / surveiller / corriger avant tout calcul) ;

- deux lignes au moins doivent contredire le `rapport_defauts.json` fourni, avec la preuve par la mesure, et le périmètre du nettoyage est écrit en **union** (6 720 lignes), jamais en addition des deux défauts ;
- la dernière ligne est le verdict : peut-on calculer un chiffre d'affaires sur ce fichier en l'état ? — répondue par le nombre, pas par l'adjectif.

Barème (10 points). Douze lignes complètes et datées (3) · dénominateurs systématiques (2) · recouvrement des 61 lignes chiffré et exploité (2) · deux contradictions du rapport fourni, preuves à l'appui (2) · verdict exprimé en FCFA et non en adjectif (1). Seuil de validation : 7. Le corrigé type est la fiche que le § 7 a produite, complétée des points 9 à 12 ; la mienne se rejoue en 45 minutes sur le fichier, et c'est le critère.

14. Résumé du chapitre

```
   IDENTITÉ          STRUCTURE         COMPLÉTUDE        UNICITÉ (2 tests)
octets · encodage   lignes × colonnes  vides / colonne   clé technique, puis clé métier
       │                   │                  │                   │
       └───────────────────┴──────────────────┴───────────────────┘
                                  ↓
          VALIDITÉ (types, domaines) → COHÉRENCE (internes, externes)
                                  ↓
                   ACTUALITÉ (bornes annoncées vs trouvées)
                                  ↓
              EXACTITUDE : le total contre une source extérieure
                                  ↓
              TRAÇABILITÉ : le rapport qu'on vous a donné est-il mesuré ?
```

Douze questions, douze nombres, douze décisions. Trois d'entre elles décident du verdict sur le socle — les types (3 421), la clé métier (3 440 collisions, 3 360 à retirer), l'exactitude (l'écart de 2 111 789 FCFA) — et leur recoupement fait le périmètre réel du nettoyage : 6 720 lignes — et une dernière vous protège : celle qui vous fait douter du document censé vous rassurer. Un diagnostic utile ne dit pas que le fichier est sale ; il dit ce qui est cassé, de combien, et ce que ça coûte au chiffre que vous allez publier.

15. À retenir

> **À retenir. ★ La clé du fichier n'est pas la clé du réel.** 243 360 `id_vente` tous distincts, et 3 440 lignes en collision sur ticket + produit + quantité : le doublon ne se voit que si on le cherche sur la bonne clé. ★ **Un défaut non traité peut s'annuler avec un autre, un défaut mal traité s'aggrave.** Réparer les textes sans dédoublonner ajoute 219 766 129 FCFA ; ne rien faire en perd 2 111 789. Le total est un contrôle, pas un verdict. ★ **Tout nombre avec son dénominateur.** 973 ne veut rien dire ; 973 sur 243 360 veut dire : une ligne mille, sauf là où ça compte. ★ **Le diagnostic ne change aucune cellule.** Il se signe sur le fichier tel quel ; la réparation s'écrit ailleurs, avec l'avant et l'après (C06). ★ **Le document de qualité qu'on vous remet est une hypothèse.** Trois de ses cinq lignes testées ne collaient pas au fichier — et personne n'avait tort : personne n'avait mesuré.

16. Évaluation formative (auto-correction, 8 min)

Q1. Un fichier a 243 360 valeurs distinctes sur sa colonne d'identifiant. Peut-on écrire « aucune ligne en trop » ? (a) oui ; (b) non, on ne le sait pas ; (c) oui si le total des montants est cohérent. → **(b).** L'unicité de la clé technique ne dit rien de la clé métier : c'est le § 5.5, 3 440 collisions avec une clé impeccable sur 243 360 lignes.

Q2. Le total d'une colonne de montants rend moins que la source indépendante. Combien de causes au minimum ? (a) 1 ; (b) 2 ; (c) autant qu'il y a de colonnes. → **(b) au minimum** — des valeurs que l'outil n'a pas su additionner, ou des lignes en trop ailleurs dans le fichier. Les deux agissent en sens contraire, et c'est ce qui rend l'écart seul peu informatif.

Q3. Quelle formule distingue une colonne entièrement numérique d'une colonne contenant du texte ? → `NB(colonne)` contre `NBVAL(colonne)` : l'égalité dit « tout est nombre », la différence donne le nombre de textes. Ici `239 939` contre 243 360.

Q4. Vrai ou faux : « un contrôle de date au format `AAAA-MM-JJ` suffit à valider l'actualité d'un fichier ». → **Faux.** 0 date mal formatée ici, et 51 lignes postérieures à la borne annoncée (max 2026-12-06). Le format est un contrôle de validité (point 6), la borne est un contrôle d'actualité (point 10).

Q5. Vous devez prouver qu'un nettoyage n'a pas cassé le chiffre d'affaires. Quelle est la moindre des preuves ? → un **rapprochement de totaux** avec le détail des écarts : avant, après, cible, et le nom de la source de la cible ; plus la liste des lignes retirées, comptée. Le § 5.11 montre pourquoi le « après » seul ne prouve rien.

Question ouverte (la plus importante). Reprenez les trois lignes du § 5.12 où l'annoncé et le mesuré diffèrent, et écrivez, pour chacune, la phrase que vous auriez mise dans la fiche pour que l'écart soit visible sans vous relire. C'est exactement le travail du chapitre : rendre l'écart lisible par quelqu'un d'autre. Corrigé attendu : une phrase par défaut, avec le nombre des deux côtés, la source du nombre annoncé, et la formule qui a produit celui du fichier — trois éléments, pas un adjectif.

Module M04.C02 — Les absences : cinq mécanismes, cinq traitements, et le coût de chaque choix

Outil de ce chapitre : le tableur, sur `ventes_brutes.csv` et `clients.csv`. Durée indicative : 5 h. Niveau : N2.

> **L'idée du chapitre.** Vous allez ouvrir un fichier de 243 360 lignes et 20 colonnes et y chercher des cellules vides. Vous n'en trouverez **aucune** : zéro, sur `4 867 200` cellules. Ce fichier est pourtant plein d'absences — 43 681 lignes sans client, 16 514 clients sans e-mail, 3 421 montants sans montant, 47 retours sans montant d'aucune sorte. L'absence n'est pas un état de la cellule : c'est une **décision d'écriture**, prise par quelqu'un, quelque part, avec une convention que vous devez retrouver avant de choisir un traitement. Cinq mécanismes produisent une absence ; cinq traitements existent ; et le chapitre se termine par le prix de chacun, en francs.
>
> **Base de travail.** `01_socle_donnees/data/brut/ventes_brutes.csv` (243 360 lignes × 20 colonnes), `01_socle_donnees/data/brut/clients.csv` (23 912 lignes × 11 colonnes), `01_socle_donnees/data/brut/stocks_quotidiens.csv` (60 000 lignes), `01_socle_donnees/data/reference/ventes_propres.csv` et sa cible `verites_terrain.csv` (15 419 985 157 FCFA). Les nombres du chapitre viennent de ces fichiers, relevés par `01_socle_donnees/scripts/chiffres_manuel.py`.

---

1. Objectifs du chapitre

À la fin de ce chapitre, vous saurez :

- **distinguer** les cinq mécanismes qui produisent une valeur absente, et dire lequel s'applique à une colonne donnée ;
- **reconnaître** les conventions qui cachent une absence (le `0`, le tiret, la chaîne vide, la ligne absente) ;
- **choisir** entre les cinq traitements — assumer, joindre, dériver, imputer, supprimer — et l'écrire en décision chiffrée ;
- **mesurer** le coût d'une politique d'absence sur le total, avant de l'appliquer ;
- **refuser** l'imputation automatique sur toute colonne qui porte une décision de gestion, et le justifier par un nombre.

2. Pourquoi cette notion est importante

- La colonne `id_client` vaut `0` sur **43 681 lignes** du fichier de ventes, soit **17,9 %**. Si vous traitez ce `0` comme une valeur, votre premier client est trois chiffres ; si vous le traitez comme un manque et que vous jetez la ligne, vous perdez **2 813 668 030 FCFA** de chiffre d'affaires pour une raison qui n'existe pas : c'est la vente au comptoir, et la règle de gestion du socle l'écrit noir sur blanc.
- Chez les clients, l'e-mail est renseigné pour **7 398 lignes sur 23 912**, soit **30,9 %**. Une campagne envoyée à « tous les clients » toucherait un client sur trois ; une campagne construite après avoir supprimé les lignes sans e-mail toucherait un client sur trois **en croyant viser toute la base**. C'est la même erreur, avec un dénominateur différent, et c'est la plus coûteuse à détecter après coup.

- Les montants : 3 421 lignes n'ont pas de nombre. Les combler à la moyenne donne un total de 15 642 826 075 FCFA, soit 222 840 918 FCFA de plus que la vérité. Les reconstituer par la règle de gestion (`montant_ht + montant_tva`) donne 15 639 751 286 FCFA, trop gros pour une autre raison que C01 avait déjà vue : les lignes en trop. Deux façons de « traiter les vides », trois ordres de grandeur d'erreur, et une seule qui se corrige en ajoutant un point de la grille.
- Enfin, une absence est parfois une **bonne nouvelle** : ces 16 514 e-mails absents sont exactement ce qu'il faut pour publier un fichier d'exercice sans exposer une personne. Le C06 y revient, et l'ordre des idées compte : on ne supprime pas une colonne par principe, on décide après avoir compté.

3. Explication simple

Un dossier médical ne dit pas « inconnu » de la même façon selon le cas. Le patient n'a pas de groupe sanguin parce qu'on ne l'a pas demandé ; il n'a pas de traitement en cours parce qu'il n'en prend pas ; il n'a pas de numéro de sécurité sociale parce qu'il l'a oublié. Les trois cases sont vides, et ne se traitent pas de la même main.

Une base de données, c'est la même chose, avec un danger de plus : **l'outil ne fait pas la différence**. Pour lui, une case vide, un `0`, une case remplie de `N/A` et une ligne qui n'existe pas sont quatre états distincts, mais aucun ne s'appelle « je ne sais pas ». Le tri par date, la moyenne, la jointure : tous traiteront ces quatre états comme quatre présences.

Retenez une phrase, elle vous sauvera des semaines : **`0` n'est jamais « rien » ; `0` est une valeur que quelqu'un a écrite.** Et « rien », dans un fichier, ça peut aussi être une ligne absente — la plus difficile des absences à trouver, parce qu'elle ne se voit que par le décompte des attendus : les 486 lignes de vente sans client référencé (point 9 de la grille) sont de celles-là.

4. Vocabulaire essentiel

Français — English	Définition simple	Piège à éviter
Valeur manquante — Missing value	Ce qui, dans le monde, existe mais n'est pas dans le fichier.	Confondre l'absence de valeur et l'absence d'objet.
Non applicable — Not applicable	Le champ n'a pas de sens pour cette ligne (le comptoir n'a pas de client).	Le combler : on invente un client pour une vente qui n'en a pas.
Valeur de sentiment — Sentinel value	Un code écrit pour dire « pas de valeur » : `-1`, `9999`, `N/A`, `inconnu`.	L'oublier dans le paramétrage de l'import : `-1` entre dans la moyenne.
Mécanisme d'absence — Missing mechanism	La raison pour laquelle la valeur manque : non-collectée, non-répondue, perdue, non-applicable, déguisée.	Traiter sans savoir lequel : les cinq n'ont pas le même remède.
Imputation — Imputation	Remplacer une absence par une valeur estimée (constante, médiane par groupe, cas semblable, modèle).	L'appliquer à une colonne qui porte une décision, ou sans le dire.
Perte par jointure — Join-induced loss	Perte de lignes causée par une jointure intérieure sur une clé absente.	Compter les lignes après la jointure pour juger du nettoyage.
Marqueur de manque — Missing indicator	Colonne booléenne qui garde la trace : « cette valeur était absente ».	Oublier de le créer quand on impute : « on ne savait pas » est alors perdu.
Politique de valeurs manquantes — Missing data policy	Le texte qui dit, colonne par colonne, ce qu'une absence veut dire et comment on la traite.	Ne pas l'écrire : chaque repreneur décide pour soi, et les totaux divergent.

5. Cours approfondi : les cinq mécanismes, les cinq traitements

5.1 Mécanisme A — Le non-applicable : la valeur n'existe pas à l'état civil

La vente au comptoir n'a pas de client. Ce n'est pas une donnée perdue, c'est une **absence de l'objet**. Le fichier l'écrit `0` dans `id_client`, sur 43 681 lignes. Le traitement utile n'est ni de l'effacer ni de le combler : c'est de **nommer** le compte. Dans un modèle, on crée une ligne « Comptoir » dans la table clients (le C06 montre l'écriture de cette ligne de convention) ; dans un tableau croisé, on remplace l'affichage du `0` par un libellé.

> **Définition.** Une **convention de codage** est un choix d'écriture, arbitraire mais écrit, qui donne à une absence un signe visible : `0`, `-1`, la chaîne vide, un enregistrement dédié. Un `0` sans texte pour le décrire n'est pas une convention, c'est une coïncidence — et une coïncidence se dément à la première reprise de fichier.

5.2 Mécanisme B — Le non-collecté : on n'a jamais posé la question

L'e-mail des clients : renseigné pour 7 398 sur 23 912. Personne n'a perdu ces adresses, elles n'ont jamais été demandées au guichet. C'est le seul des cinq mécanismes qui se corrige **en amont**, par un formulaire ou une habitude de caisse, et jamais en aval par un calcul. Un analyste qui comble un non-collecté fabrique une liste de relance adressée à des inconnus.

5.3 Mécanisme C — Le non-répondu : la question a été posée, la réponse non

La ville manque pour 1 911 clients sur 23 912 (8,0 %). Contrairement au cas précédent, le fait existe — le client est quelque part — et il est parfois récupérable ailleurs : une ville dans un autre champ, une région renseignée, le code postal d'une adresse de livraison. **Joindre**, voilà le traitement propre du non-répondu ; le remplissage par la ville majoritaire en est le traitement sale, et il s'appelle une décision commerciale, pas une donnée.

5.4 Mécanisme D — La perte à l'import ou à la conversion

C'est le mécanisme que le fichier de ventes contient le plus : 3 421 lignes où `montant_ttc` n'est pas un nombre, 47 lignes de retour sans montant exploitable, et, dans le fichier reçu en M01, 13 colonnes là où le brut en a 20. Rien n'a été décidé par un humain : la valeur a été **abîmée en route** — une virgule décimale, un séparateur de milliers, un encodage, une colonne que l'export n'emporte pas. Le remède n'est pas un remplissage mais une **reconstruction par la règle** : le socle écrit que `montant_ttc = montant_ht + montant_tva`, donc un montant perdu se reconstitue ; il ne s'impute pas.

> **Dans les faits.** Les trois quarts des « données manquantes » d'un export ERP ne sont pas manquantes : elles sont mal lues. Le réflexe qui fait gagner des heures est de rouvrir le même fichier avec le séparateur et l'encodage corrigés, et de comparer les compteurs avant de parler de nettoyage. Le point 1 de la grille (identité du fichier) n'est pas une formalité : c'est lui qui évite de nettoyer un défaut de lecture.

5.5 Mécanisme E — L'absence déguisée en valeur tranquille

Le plus discret. `taux_remise` vaut 0 sur **157 634 lignes**, soit 64,8 % du fichier. Est-ce « pas de remise » ou « on n'a pas saisi la remise » ? Le fichier seul ne répond pas. Répondre, c'est croiser : si la remise vaut 0 alors que le client a une remise consentie consignée dans `remises_manuelles.xlsx` (18 200 lignes, 2 043 692 000 FCFA), le 0 est un mensonge de saisie sur ces lignes-là. C'est le seul mécanisme dont le diagnostic passe par un **rapprochement entre deux fichiers**, et non par une formule dans une colonne.

> **Attention.** Toute valeur « tranquille » — le 0 de la remise, la date du 01/01/1900, le `-1` d'un code magasin, le texte `inconnu` — est candidate à l'absence déguisée. Le test est simple : demandez-vous ce que deviennent une moyenne, une somme et un tri si cette valeur est comprise comme ce qu'elle n'est pas. Si la réponse est « rien de grave », vous n'avez pas encore regardé la colonne des montants.

5.6 Traitements (1/2) : assumer, joindre

Assumer (le cas A, parfois B) : on garde la ligne, on renomme la valeur, on écrit la convention. Coût : zéro cellule modifiée. Gain : le total reste le total. C'est le traitement à préférer chaque fois qu'il est vrai, et le seul qui ne demande aucune autorisation.

Joindre (le cas C, parfois D) : aller chercher la valeur ailleurs dans le système, sans jamais la recopier à la main — la ville depuis une adresse de livraison, la catégorie depuis `dictionnaire_produits.csv`, le taux de TVA depuis la fiche produit. La jointure a un prix connu : elle perd les lignes sans correspondance, ici 486 lignes. On compte donc **avant et après**, et l'écart s'écrit dans le journal (C06).

> **Définition.** Une **jointure externe** (à gauche, à droite, pleine) conserve les lignes sans correspondance en laissant le champ complémentaire vide ; une jointure **intérieure** les supprime. Sur ce fichier, la différence entre les deux s'appelle 486 lignes : la jointure intérieure est le trou de mémoire le plus courant du nettoyage, et le seul qui se masque derrière un total qui a simplement « un peu baissé ».

5.7 Traitements (2/2) : dériver, imputer, supprimer

Dériver (le cas D) : recalculer ce que la règle de gestion autorise à recalculer. `montant_ttc = montant_ht + montant_tva` rend un nombre sur les 3 421 lignes. Vérification : sur le fichier réparé sans toucher aux doublons, le total donne 15 639 751 286 FCFA contre 15 419 985 157 FCFA attendus. La dérivation est bonne, et le reliquat d'écart est la preuve qu'il subsiste un défaut **d'une autre nature** — ce qui est un résultat, pas un échec.

> **Définition. Imputer par le cas le plus semblable** (hot-deck, ou k plus proches voisins) consiste à emprunter la valeur d'une ligne qui ressemble fort à celle qui manque : même produit, même magasin, même mois, même type de client. C'est la seule imputation qui se défende sur une colonne monétaire — et encore : elle se justifie par la ressemblance des lignes, jamais par la commodité de l'outil.

Imputer : remplacer par une valeur estimée. Quatre familles, du plus prudent au plus risqué : la constante documentée, la médiane par groupe (le prix médian du produit, pas du fichier entier), la valeur du cas le plus semblable (même client, même mois, même catégorie), le modèle (régression — M08). Règle absolue : **on n'impute pas une colonne qui porte une décision de**

gestion, et on crée le marqueur qui dit que la valeur est estimée. Le prix, mesuré : 15 642 826 075 FCFA contre 15 639 751 286 FCFA, soit `3 074 789` FCFA de plus, pour 1,4 % des lignes.

Supprimer : retirer les lignes. Uniquement si la ligne est inutilisable pour l'analyse — ligne vide, test, doublon confirmé (C03). La suppression se chiffre : « 6 720 lignes sur 243 360, soit 2,8 % du fichier, 2 111 789 FCFA d'un côté et 219 766 129 FCFA de l'autre » n'est pas une suppression, c'est un arbitrage.

> **Définition.** Un **marqueur de manque** est une colonne ajoutée, en général booléenne, qui survit à la réparation : `ttc_reconstitue = VRAI` sur les 3 421 lignes. Elle coûte une formule et rend possible la seule question qui fâche six mois plus tard : « ce chiffre, il vient du magasin ou de mon calcul ? »

5.8 Les cinq mécanismes et leurs chiffres, en une grille

Mécanisme	Marqueur dans le fichier	Colonne touchée ici	Nombre	Traitement qui convient
A · non-applicable	`id_client = 0`, écrit par la règle	`id_client`	43 681 lignes	assumer : compte « Comptoir »
B · non-collecté	chaîne vide, jamais demandé	`clients.email`	16 514 clients	assumer, et changer le formulaire
C · non-répondu	chaîne vide	`clients.ville`	1 911 clients	joindre, sinon compter
D · perdu à l'import	texte au lieu d'un nombre	`montant_ttc`	3 421 lignes	dériver par la règle
E · absence déguisée	une valeur tranquille	`taux_remise`	157 634 lignes	croiser avec un autre fichier

> **Attention.** Les compteurs de cette grille se lisent avec la colonne « traitement qui convient », jamais seuls. Le même nombre de lignes — 43 681 — appelle à supprimer si on lit « vides » et à assumer si on lit « convention écrite par la règle ». Un chiffre de qualité sans mécanisme est une invitation à la mauvaise opération, faite de bonne foi.

Une sixième ligne mériterait la grille : les lignes **qui devraient exister et n'existent pas**. Sur ce socle, le compteur est à zéro — 0 période manquante et 0 période qui se recouvre dans `couts_achat.csv` (1 694 lignes de périodes continues). Ce n'est pas un vide, c'est un **plein vérifié**, et le compter rend le reste crédible.

Les absences d'un fichier de ventes : cinq mécanismes, cinq traitements

243 360 lignes, 0 cellule vide : ici, l'absence est toujours écrite comme une valeur.

A	**non-applicable** id_client = 0 · écrit par la règle touchées : 43 681 lignes	→	**assumer, nommer « Comptoir »** une cellule modifiée : non marqueur de manque à poser : inutile
B	**non-collecté** clients.email jamais demandé touchées : 16 514 clients	→	**changer le formulaire, en amont** une cellule modifiée : non marqueur de manque à poser : inutile
C	**non-répondu** clients.ville laissée vide touchées : 1 911 clients	→	**joindre, sinon compter** une cellule modifiée : non marqueur de manque à poser : inutile
D	**perdu à l'import** montant_ttc devenu du texte touchées : 3 421 lignes	→	**dériver : ht + tva** une cellule modifiée : oui, par la règle marqueur de manque à poser : oui
E	**absence déguisée** taux_remise à 0, nulle part vide touchées : 157 634 lignes	→	**croiser un autre fichier** une cellule modifiée : non marqueur de manque à poser : oui

Le prix des deux traitements de la colonne D, mesuré sur le même fichier :
dériver par la règle · 15 639 751 286 FCFA | imputer à la moyenne · 15 642 826 075 FCFA
cible vérifiée : 15 419 985 157 FCFA. L'écart restant n'est pas une histoire d'absence : ce sont les lignes en trop (C03).

6. Exemple concret : la phrase qui fait basculer un tableau de bord

Lundi, 9 h 15. La direction commerciale veut « le chiffre d'affaires par client », pour récompenser les trois mieux disposés. Vous joignez les ventes aux clients, vous triez, vous publiez. Trois versions du même tableau, et la seule qui ne se discute pas est celle que personne n'obtient par défaut :

Version	Ce qu'on a fait	Total	Ce qui se passe
jointure intérieure	on a gardé ce qui a un client	minoré	les 486 lignes sans correspondance disparaissent sans message
`0` laissé tel quel	le comptoir est resté dans le classement	juste de montant, faux de sens	le « premier client » du magasin est `0`, avec 2 813 668 030 FCFA
comptoir nommé	le `0` devient « Comptoir »	juste et lisible	le dirigeant voit un canal, pas un client, et peut redécider

La leçon n'est pas « il faut supprimer les `0` ». Elle est que **la même absence produit un chiffre faux, un chiffre vrai mais muet, ou un chiffre vrai et parlant, selon le nom que vous lui donnez**. Une large part des erreurs de tableau de bord ne sont pas des erreurs de calcul : ce sont des absences auxquelles personne n'a donné de nom.

7. Démonstration pas à pas : de l'œil au compteur, puis à la décision

Sur la feuille `Ventes` (243 360 lignes × 20 colonnes, colonnes A à T), on écrit une feuille `MANQUES` : une ligne par colonne du fichier, six colonnes de compte. Les mêmes compteurs se rejouent sur `clients.csv`.

1. **Les vraies vides.** `=NB.SI(Ventes!N2:N243361;"")` sur `montant_ttc` → 0. Faites-le sur les vingt colonnes : le total est de 0 cellule vide. **Le fichier ne contient pas d'absence écrite comme un vide** — conclusion à écrire telle quelle, elle change tout le reste de la méthode.

2. **Les zéros, par colonne.** `=NB.SI(Ventes!G2:G243361;0)` sur `id_client` → 43 681. Le même compteur sur `est_retour` (colonne Q) rend `240 512`. Deux zéros, deux natures : à gauche un objet absent, à droite un booléen à « non ». Le signe est identique, le mécanisme non.
3. **Les valeurs de sentiment.** `=NB.SI(Ventes!C2:C243361;"-")`, puis sur `"N/A"`, `"inconnu"`, `"0/00/0000"`, `"-"&""` : **0 partout**, sur les vingt colonnes. Cette ligne de fiche est un résultat : elle dit que le producteur n'a aucune convention textuelle du manque, donc que tout ce qui ressemble à un manque a été écrit comme une valeur.
4. **Ce que le `0` coûte.** Somme conditionnelle des montants numériques sur les lignes à `id_client = 0` → 2 813 668 030 FCFA, à comparer à 15 419 985 157 FCFA : un septième du chiffre d'affaires dépend d'un caractère.
5. **La remise silencieuse.** `=NB.SI(Ventes!K2:K243361;0)` → 157 634 lignes. Croisez avec `remises_manuelles.xlsx` (18 200 lignes, 2 043 692 000 FCFA) : la question à poser au métier n'est pas « combien de zéros ? » mais « ces deux fichiers se contredisent-ils sur les mêmes clients, sur les mêmes mois ? ».
6. **L'absence de schéma.** Ouvrez le fichier reçu en M01, `ventes_magasin5_2025.csv` : 13 colonnes, 489 lignes, pas de `montant_tva`. Le brut en a 20. L'écart n'est pas un vide, c'est une **colonne qui n'existe pas** — la seule absence qui ne se voie pas dans une colonne, donc la dernière que l'on regarde.
7. **La décision, écrite.** Une ligne par colonne, un verbe parmi assumer / joindre / dériver / imputer / supprimer, le nombre, le coût, la date. C'est la colonne décision qui sépare un inventaire d'un travail : sans elle, votre fiche sera reprise par quelqu'un qui supprimera les 43 681 lignes.

> **Boîte à outils du chapitre.** Une grille à cinq colonnes — colonne, compteur, mécanisme, traitement, coût — et une case obligatoire « ce que je ne touche pas ». Cette dernière case est celle qui, en reprise, vous évite deux semaines d'explication sur pourquoi le total d'un fichier réparé ne ressemble plus à celui du fichier reçu.
>
> **À retenir.** Un traitement se juge à ce qu'il laisse intact. Assumer ne touche aucune cellule et garde le total ; joindre ne touche aucune cellule mais change le périmètre de la jointure ; dériver réécrit une colonne à partir d'une règle ; imputer invente une valeur ; supprimer détruit un fait. Du premier au dernier, l'initiative passe du fichier à vous — et c'est exactement dans ce sens qu'il faut choisir, jamais l'inverse.

8. Erreurs fréquentes

Symptôme visible	Cause réelle	Correction
« Aucune donnée manquante, le fichier est plein »	le comptage s'est limité à la cellule vide	compter aussi les zéros, les valeurs de sentiment, les colonnes absentes : mécanismes A à E
Le total baisse d'un septième après nettoyage	le `0` de `id_client` a été pris pour un manque	assumer la convention et nommer le compte, au lieu de supprimer la ligne
Le tableau croisé a moins de lignes que le fichier	jointure intérieure sur une clé absente	joindre en externe et compter les perdantes : ici 486 lignes, 372 clients
La moyenne des montants a bougé après remplissage	imputation à la moyenne générale sur une colonne monétaire à forte variance	reconstituer par la règle, ou imputer par groupe, et poser le marqueur
Des dates d'e-mails sont devenues 1900	la date vide est lue comme une date nulle par l'import	déclarer les valeurs de sentiment à l'import, et contrôler le `MIN` de la colonne de dates
« J'ai écrit Non renseigné, c'est propre »	une chaîne ajoutée pour dire « rien » crée une sixième modalité	ne pas introduire de modalité textuelle dans une colonne de clés : garder la ligne non jointe et la compter

9. Bonnes pratiques professionnelles

- **Un dictionnaire de colonnes avec une ligne « ce que veut dire un vide ».** C'est la seule ligne que tout le monde saute à la lecture, et la première consultée le jour où ça casse.
- **Comptez les absences par ligne, pas seulement par colonne.** Un fichier peut avoir 0 % de vides par colonne et des lignes inexploitables : les absences se cumulent sur les mêmes lignes, et c'est le lecteur qui trinque.
- **Jamais d'imputation sans marqueur.** Sans la colonne « valeur estimée ici », personne — y compris vous dans six mois — ne distingue le mesuré de l'inventé.
- **Un imputé ne se mélange pas à un vrai dans un verdict.** S'il faut un chiffre unique pour la direction, donnez-le et donnez la fourchette : 15 639 751 286 FCFA par la règle, et non moins de 15 419 985 157 FCFA une fois les lignes en trop retirées.
- **Un compteur à zéro se consigne.** « 0 valeur de sentiment », « 0 date au format mixte », « 0 période trouée » : ces lignes sont ce qui permet à un repreneur de contredire un rapport de qualité mal mesuré — le vôtre compris.
- **Sur une colonne de décision, l'absence est une information.** Un client à 0 % de remise n'est pas un client raisonnable, c'est un client à qui personne n'a rien accordé. Même valeur, deux lectures, et une seule se pilote.

> **Conseil professionnel.** Écrivez votre politique d'absence **avant** le premier nettoyage, et datée. Dans le sens inverse, on nettoie d'abord ce qui gêne, puis on rédige une règle qui justifie ce qui a été fait : le document ressemble alors à un procès-verbal, et il ne protège personne. Le test est simple : si votre politique rédigée ne prédisait pas les décisions que vous avez prises, ce n'est pas une politique.

- **Faites relire la politique par le métier, pas par un collègue analyste.** Si le directeur de magasin ne reconnaît pas le comptoir dans votre ligne `id_client`, la politique est fausse, quel que soit le nombre.

10. Exercice guidé — trier quatre colonnes en quatre traitements (30 min, /10)

Quatre colonnes, relevées à l'instant : `id_client` (43 681 zéros), `montant_ttc` (3 421 textes), `clients.ville` (1 911 vides), `taux_remise` (157 634 zéros). Vous décidez, je corrige à chaque étape.

Étape 1. Nommez le mécanisme de chacune. Attendu : A pour `id_client`, D pour `montant_ttc`, C pour la ville, E pour la remise. Si vous classez la ville en B (non-collecté), vérifiez : la colonne existe, donc la question a été posée — c'est la réponse qui manque, pas la question. B et C ne se réparent pas au même endroit : l'un au formulaire, l'autre à la jointure.

Étape 2. Pour `montant_ttc`, écrivez la dérivation. `=SI(ESTNUM(N2);N2;L2+M2)` (N = `montant_ttc`, L = `montant_ht`, M = `montant_tva`) rend un nombre sur les 3 421 lignes. Reportez le total ainsi obtenu, sans toucher aux doublons : 15 639 751 286 FCFA. Il reste un écart de +219 766 129 FCFA — ne le corrigez surtout pas ici : c'est la signature des lignes en trop, et son tour vient en C03.

Étape 3. Comparez avec l'imputation à la moyenne des montants positifs : 15 642 826 075 FCFA. Les deux méthodes sont « correctes » au sens du tableur ; l'une dérive d'une règle écrite, l'autre d'une distribution. **Écrivez la phrase de décision**, celle qui ira dans le journal. Une décision qui ne dit pas pourquoi l'autre méthode a été écartée n'est pas une décision, c'est une habitude.

Étape 4. Pour la ville, combien de lignes récupérables par jointure ? Réponse mesurée : 0 — `clients.csv` ne contient qu'une adresse, et le fichier de ventes ne contient pas de ville. Le traitement devient donc « assumer, et compter ». Un traitement qu'on ne peut pas appliquer n'est pas un choix, c'est un constat : écrivez-le comme tel, sinon on vous reprochera de ne pas l'avoir tenté.

Étape 5. Pour `taux_remise`, produisez le compteur de croisement : combien de clients ont une remise consignée dans `remises_manuelles.xlsx` **et** une remise à 0 sur la même période. C'est le seul des quatre qui ne se résout pas dans le fichier. Rendu attendu : « question ouverte au métier — 157 634 lignes portent un 0, 18 200 lignes du fichier de saisies affirment le contraire sur les mêmes clients ; en l'absence de réponse, le 0 est conservé et la moyenne des remises est retirée des indicateurs de pilotage ». **Note /10** : 3 points pour les quatre mécanismes nommés et justifiés, 4 points pour les deux totaux et la phrase de décision, 3 points pour l'honnêteté des constats non exécutables. Un nombre sans dénominateur, ici, vaut zéro.

11. Exercices autonomes

Exercice 2.1 (★) — Les quatre compteurs (10 min). Sur `clients.csv`, relevez les quatre compteurs d'absence (ville, e-mail, téléphone, segment) avec leur dénominateur. Écrivez pour chacun la phrase de politique, en un verbe parmi : assumer, joindre, dériver, imputer, supprimer.

Exercice 2.2 (★★) — Le prix d'un `0` (15 min). Calculez le total des montants numériques sur les lignes où `id_client` vaut 0, puis ce que devient le total du fichier si vous supprimez ces lignes. Donnez les deux nombres, leur part du total, et la phrase de deux lignes que vous enverriez au directeur financier pour refuser la suppression.

Exercice 2.3 (★★) — La colonne qui n'existe pas (12 min). Comparez l'en-tête du brut (20 colonnes) à celui du fichier reçu `ventes_magasin5_2025.csv` (13 colonnes, 489 lignes). Citez les colonnes absentes, et dites pour laquelle au moins l'absence change un calcul déjà fait en C01 ou en M03.

Exercice 2.4 (★★) — Le faux négatif du compteur (15 min). Vous cherchez les tirets `-` dans tout le fichier : la recherche rend 0 sur les vingt colonnes. Un collègue affirme que son ERP écrit systématiquement `-` pour les dates inconnues. Que vérifiez avant de le croire, et que mettez-vous dans la fiche ? Trois indices : la colonne `date_vente`, le nombre de lignes, le nom du fichier.

Exercice 2.5 (★★★) — Imputation contrôlée (25 min). Sur une copie, traitez les 3 421 `montant_ttc` de deux façons : (a) moyenne des montants positifs du fichier ; (b) reconstitution `montant_ht + montant_tva`. Pour chaque méthode : le total, son écart à 15 419 985 157 FCFA, et le nombre de lignes dont le montant imputé s'écarte de plus de `100 000` FCFA du montant reconstitué. Concluez en une phrase sur le sort d'une valeur monétaire absente dans un fichier de ventes.

12. Correction détaillée

Exercice 2.1. Ville : 1 911 vides sur 23 912 (8,0 %) → joindre, mais la jointure est impossible (voir 2.3), donc assumer et compter. E-mail : 16 514 sur 23 912 (69,1 %) → assumer ; non-collecté, et son absence est un actif pour la publication (C06). Téléphone : 0 vide sur 23 912 → rien à traiter, **mais à écrire** : c'est ce compteur à zéro qui autorisera l'appariement du C03. Segment : 0 vide → rien. Les deux compteurs nuls sont ceux qu'on est tenté de passer sous silence, et ce sont eux qui rendent défendables les deux autres.

Exercice 2.2. Total des lignes comptoir : 2 813 668 030 FCFA sur 43 681 lignes, soit 17,9 % du fichier en lignes et un peu plus de 18 % du total attendu. Phrase attendue : « supprimer les lignes sans `id_client` retire un septième du chiffre d'affaires ; ce n'est pas un nettoyage, c'est un changement de périmètre — le comptoir est un canal, pas un client manquant ». Notez le

glissement probable : la part en lignes (17,9 %) et la part en montant ne sont pas égales, et c'est la seconde qui décide.

Exercice 2.3. L'en-tête reçue est `n_ticket;date;heure;magasin;vendeur;client;produit;categorie;quantite;prix_unitaire_ht;remise;montant_ht;montant` Manquent, entre autres, `montant_tva`, `mode_paiement`, `canal`, `est_retour`, `poids_kg`, `mois`, `annee`. La colonne dont l'absence change un calcul déjà fait est **`montant_tva`** : sans elle, la dérivation du mécanisme D est impossible et les 3 421 montants texte deviennent réellement irréparables — d'où l'intérêt de compter les colonnes absentes du fichier que l'on va réparer, pas seulement de celui que l'on a sous les yeux. `est_retour` est la seconde : sans elle, les 2 801 montants négatifs n'ont plus de signification.

Exercice 2.4. Trois vérifications, dans cet ordre : que la recherche porte sur le **fichier texte**, pas sur la feuille importée (un `-` avalé par l'import devient un vide ou un nombre, et votre compteur ment poliment) ; que la colonne testée est bien `date_vente` et couvre les 243 360 lignes ; que le fichier décrit par le collègue est bien celui que vous avez (nom, taille : 28 516 920 octets). Ligne de fiche attendue : « 0 occurrence de `-` dans les 20 colonnes de `ventes_brutes.csv`, recherche sur le fichier texte, 243 360 lignes — la convention de l'ERP n'est donc pas arrivée jusqu'ici, ou l'export l'a transformée : question ouverte à la source ». Un résultat négatif consigné avec sa méthode n'est pas une absence de défaut, c'est une preuve.

Exercice 2.5. (a) moyenne : 15 642 826 075 FCFA, écart +222 840 918 FCFA ; (b) dérivation : 15 639 751 286 FCFA, écart +219 766 129 FCFA. Les deux restent gros parce qu'aucune des deux ne traite les lignes en trop, mais ils ne se valent pas : `3 074 789` FCFA séparent les deux méthodes — le prix pur de l'imputation, sur 1,4 % des lignes. Le nombre de lignes qui dévient de plus de `100 000` FCFA se lit par `=NB.SI.ENS(…)` sur une colonne utilitaire `=ABS(imputé-reconstitué)` ; il se compte en centaines, pas en dizaines. Conclusion : une valeur monétaire absente ne s'impute pas, elle se reconstitue par la règle ; si la règle ne permet pas de la reconstituer, la ligne sort du périmètre et la sortie s'affiche. Mettre la moyenne du magasin dans une ligne de facture n'est pas une estimation prudente, c'est une écriture de plus.

13. Mini-projet M04.P2 — « Politique de valeurs manquantes » (1 h)

Livrable : `politique_absences.md`, une page et demie, pour les vingt colonnes du fichier de ventes et les onze de `clients.csv`. Contraintes :

- une ligne **par colonne**, jamais par fichier, avec les quatre compteurs (vides, zéros, valeurs de sentiment, hors domaine) et le dénominateur ;
- pour chaque compteur non nul : le mécanisme (A-E), le traitement, et le **coût chiffré** du traitement choisi et du principal traitement écarté ;
- deux colonnes au moins doivent finir en « question ouverte à la source », la question écrite noir sur blanc (c'est le cas de `taux_remise`) ;
- la dernière section est la politique de **non-réparation** : ce que vous refusez de toucher, et pourquoi.

Barème (10 points). Complétude de la grille (3) · justesse des compteurs, dénominateur inclus (2) · coûts chiffrés des deux traitements pour au moins six colonnes (2) · questions à la source formulées et adressables (1) · section de non-réparation argumentée (2). Seuil de validation : 7. Rendu maximum 8 pour un projet sans section de non-réparation : un nettoyage qui ne dit pas ce qu'il laisse en l'état n'est pas un nettoyage, c'est un tri.

14. Résumé du chapitre

```
UNE CELLULE, QUATRE VISAGES          CINQ MÉCANISMES            CINQ TRAITEMENTS
   ┌───┐   ┌───┐   ┌───┐   ┌───┐    A non-applicable        ASSUMER (nommer)
   │   │   │ 0 │   │N/A│   │ - │    B non-collecté          (amont : le formulaire)
   └───┘   └───┘   └───┘   └───┘    C non-répondu           JOINDRE (autre source)
    vide   valeur sentiment  tiret   D perdu à l'import      DÉRIVER (la règle écrite)
                                     E absence déguisée      IMPUTER (+ marqueur)
                                                             SUPPRIMER (chiffré, en dernier)
```

Le fichier de ventes est le contre-exemple parfait de l'intuition : 0 cellule vide, et 43 681 lignes sans client, 3 421 lignes sans montant, 157 634 lignes dont on ignore si le 0 veut dire « rien » ou « personne n'a rien accordé ». L'absence se mesure par quatre compteurs et se traite par une décision écrite. Le reste — le remplissage automatique, la suppression de confort, le `0` compris comme un nombre — est exactement ce qui transforme un problème de 2 111 789 FCFA en un problème de 222 840 918.

15. À retenir

> **À retenir. ★ Quatre compteurs, pas un.** Vides, zéros, valeurs de sentiment, colonnes absentes. Ici : 0, 43 681, 0, 7. ★ **`0` n'est jamais « rien ».** Dans `id_client`, c'est le comptoir (17,9 % du fichier, 2 813 668 030 FCFA) ; dans `est_retour`, un booléen à « non » ; dans `taux_remise`, une question au métier. Même signe, trois sorts. ★ **Une valeur monétaire ne s'impute pas, elle se reconstitue.** La règle rend 15 639 751 286 FCFA ; la moyenne en ajoute `3 074 789`, et aucune des deux ne pardonne l'oubli des doublons. ★ **Toute imputation porte un marqueur.** Sinon l'information « on ne savait pas » est détruite, et le mesuré devient indistinguable de l'inventé. ★ **Un compteur à zéro se consigne.** « 0 date au format mixte », « 0 valeur de sentiment », « 0 période trouée » : ce sont ces lignes-là qui permettent de contredire un rapport de qualité écrit de mémoire.

16. Évaluation formative (auto-correction, 8 min)

Q1. Un fichier sans une seule cellule vide contient-il des absences ? (a) non ; (b) oui, si une valeur écrite tient lieu d'absence ; (c) seulement si une colonne est vide en entier. → **(b)**, et ici les deux premières formes coexistent : 43 681 lignes où le client est `0`, 3 421 montants non numériques. La cellule est remplie, le fait est absent.

Q2. Le `0` de `est_retour` est une absence. Vrai ou faux ? → **Faux** : c'est un booléen à « non ». Le piège n'est pas la valeur, c'est la colonne — le même signe dans `id_client` est un objet absent.

Q3. Vous joignez ventes → clients en interne pour enrichir les 243 360 lignes. Combien de lignes risquent de disparaître, et pourquoi ? → 486, dont 372 clients distincts inconnus. Une jointure intérieure ne prévient pas : c'est le comptage avant/après qui la fait parler.

Q4. Quelle imputation est défendable pour un `montant_ttc` absent ? (a) moyenne ; (b) médiane par catégorie ; (c) `montant_ht + montant_tva` ; (d) aucune, on exclut la ligne. → **(c)**, parce que c'est une règle écrite et vérifiable ; à défaut **(d)**, et on l'affiche. (a) et (b) servent à bornier, jamais à publier.

Q5. Comment un repreneur retrouve-t-il, six mois plus tard, si un montant était présent ou reconstitué ? → par le marqueur et par le journal (C06) : « 3 421 lignes, montant reconstitué par `ht+tva`, le [date] ; contrôle : total 15 639 751 286 contre cible 15 419 985 157 ».

Question ouverte (la plus importante). Reprenez `taux_remise` (157 634 zéros) et écrivez les deux phrases : celle que vous enverriez au métier pour poser la question, et celle que vous ins-

cririez dans la politique si le métier ne répond jamais dans le délai. Corrigé attendu : la première nomme le croisement demandé (`remises_manuelles.xlsx`, 18 200 lignes, 2 043 692 000 FCFA) et le nombre de clients en contradiction ; la seconde fige le traitement — garder le 0, ajouter `remise_non_verifiee`, et retirer la moyenne des remises des indicateurs de pilotage. Deux phrases, deux responsabilités : demander, puis assumer l'absence de réponse.

Module M04.C03 — Doublons : la clé qui ment, la clé qui dit vrai, et le dictionnaire de correspondances

Outil de ce chapitre : le tableur, sur `ventes_brutes.csv`, `clients.csv` et `dictionnaire_produits.csv`. Durée indicative : 5 h. Niveau : N2.

> **L'idée du chapitre.** Voici un fichier de 243 360 lignes dont la clé primaire est impeccable — 243 360 valeurs, toutes distinctes — et qui contient 3 360 lignes en trop. Le doublon n'est pas une anomalie de saisie que l'on débusque en triant : c'est un **choix de définition**, et ce choix s'écrit sur une clé métier que le fichier ne fournit pas. Ce chapitre fait quatre choses : il montre les quatre niveaux où l'on peut chercher un doublon et ce que chacun laisse passer ; il lit dans l'ordre des lignes le mécanisme qui a produit le défaut ; il énonce la règle de garde — laquelle des deux lignes on conserve, et pourquoi ; et il traite le jumeau du doublon, la **correspondance**, ce moment où deux écritures du même objet ne sont pas deux lignes mais bien une seule.
>
> **Base de travail.** `01_socle_donnees/data/brut/ventes_brutes.csv` (243 360 lignes × 20 colonnes), `01_socle_donnees/data/brut/clients.csv` (23 912), `01_socle_donnees/data/brut/produits.csv` (154), `01_socle_donnees/data/reference/dictionnaire_produits.csv` (154), `ventes_propres.csv` (240 000 lignes) et sa cible 15 419 985 157 FCFA. Tous les nombres viennent de ces fichiers.

---

1. Objectifs du chapitre

À la fin de ce chapitre, vous saurez :

- **nommer** le niveau auquel vous avez cherché un doublon — ligne entière, clé technique, clé métier, similarité — et dire ce que ce niveau ne voit pas ;
- **distinguer** une répétition structurelle normale (plusieurs lignes sur un même ticket) d'une répétition anormale ;
- **écrire** une règle de garde qui soit déterministe, et la justifier par ce que le fichier dit de son propre accident ;
- **construire** un dictionnaire de correspondances (15 écritures de catégorie vers 7 réelles, `0` de comptoir vers un compte nommé) et en mesurer l'effet sur les classements ;
- **appairer** deux écritures d'un même client sans casser les homonymes, et chiffrer le dégât de la règle naïve.

2. Pourquoi cette notion est importante

- Les 3 360 lignes en trop de notre fichier valent **+219 766 129 FCFA** une fois les montants réparés. C'est cent fois le montant que l'on perd à ne pas réparer les textes (2 111 789 FCFA) : le doublon est, de loin, le plus gros défaut de ce socle — et celui qu'aucun contrôle de clé primaire ne signale.
- Un dédoublonnage automatique sur `id_ticket` seul effacerait **97 199 lignes** parfaitement valides : c'est le nombre de lignes qui partagent un ticket parce qu'un ticket a plusieurs articles (1,67 ligne par ticket en moyenne, jusqu'à 6). La même commande, sur un fichier de factures, peut détruire 40 % du chiffre d'affaires.

- Côté clients, la règle « même nom » fusionne 11 570 lignes là où une règle « même nom et mêmes chiffres de téléphone » n'en fusionne que 412 : la différence, 11 158 lignes, ce sont des clients bien réels qui ont un nom banal. Un outil de dédoublonnage livré avec « le score de similarité par défaut » fait exactement ce premier choix.
- Une correspondance ratée ne change pas le total : elle change le **classement**. Nos 154 produits portent 15 écritures de catégorie là où le référentiel en admet 7 ; regrouper dessus produit 15 lignes de tableau croisé au lieu de 7, et le directeur qui lit « `MATÉRIAUX` devant `Matériaux` » tirera une conclusion sur des fantômes.

3. Explication simple

Deux frères jumeaux, même visage, même nom, nés le même jour : la police nationale dira qu'ils sont deux personnes, parce qu'ils ont deux numéros d'identité. Deux lignes de facture avec deux `id_vente` différents, même ticket, même produit, même quantité : un comptable dira qu'il y a deux fois la même vente, parce qu'il n'y a pas deux fois le même -article.

Un doublon n'est donc **jamais** une propriété des lignes : c'est une propriété d'une définition de l'objet. Vous décidez ce qu'est « une vente », et la réponse tombe. La décision se prend avec trois questions, dans cet ordre :

1. qu'est-ce qu'une ligne, dans le monde ? (ici : un article d'un ticket) ;
2. est-ce que deux lignes peuvent légitimement avoir les mêmes valeurs sur ce qui définit l'objet ? (ici : non) ;
3. si oui, quelles colonnes ajoutées rendraient la répétition anormale ? (ici : aucune — la quantité ne sauve rien, 7 640 des 8 453 lignes en collision ont aussi la même quantité).

> **Définition.** Un **doublon logique** est une paire de lignes qui décrivent le même fait alors qu'une seule pourrait le décrire. Un **doublon matériel** est une paire de lignes dont toutes les valeurs sont égales. Un fichier peut contenir des milliers de doublons logiques et pas un seul doublon matériel — c'est le cas du nôtre : 3 360 contre 0.

4. Vocabulaire essentiel

Français — English	Définition simple	Piège à éviter
Clé primaire — Primary key	Le champ que le fichier utilise pour garantir l'unicité.	Le prendre pour la définition du réel : ici il est propre, le défaut est là quand même.
Clé candidate — Candidate key	Tout jeu de colonnes dont les valeurs identifient une ligne.	Ne pas les tester toutes : si deux clés candidates se contredisent, l'une ment.
Clé métier — Business key	La combinaison qui, hors informatique, définit l'objet (ticket + produit + quantité).	L'écrire trop large ou trop serrée : trop large, tout est doublon ; trop serrée, rien ne l'est.
Dédoublonnage — Deduplication	Opération qui ne conserve qu'une ligne par groupe de doublons.	Sans règle de garde : « garder la première » est une décision, pas une technique.
Règle de garde — Survivorship rule	Le critère qui désigne la ligne conservée dans chaque groupe.	La laisser implicite ; elle doit être écrite et rejouable.
Registre des doublons — Duplicate register	Le fichier compagnon qui liste ce qui a été retiré et pourquoi.	Le jeter : il est la seule preuve que le nettoyage n'a pas inventé le total.
Dictionnaire de correspondance — Crosswalk	Table qui relie deux écritures d'un même objet (`MATÉRIAUX` → `Matériaux`).	La coder dans une formule recopiée au lieu d'en faire une table lisible.

Français — English	Définition simple	Piège à éviter
Appariement tolérant — Fuzzy matching	Rapprochement par similarité (norme, distance d'édition), avec seuil.	Croire qu'un seuil par défaut est un jugement métier.
Clé de normalisation — Normalized key	Valeur transformée (minuscules, sans accent, sans espace, sans ponctuation) servant de point de rencontre.	Comparer des chaînes au lieu de comparer leurs normes.
Faux positif / faux négatif	Fusionner deux objets distincts / en manquer un.	Publier un score sans les deux taux : le premier détruit, le second laisse vivre le défaut.

5. Cours approfondi

5.1 Les quatre niveaux où l'on peut chercher un doublon

Niveau	Test	Ce qu'il voit ici	Ce qu'il laisse passer
1 · ligne entière	toutes colonnes égales	0 ligne	tout doublon dont une valeur a bougé
2 · clé technique	`id_vente` distincts	0 doublon	le doublon lui-même
3 · clé métier complète	ticket + produit + quantité	3 440 lignes	—
4 · clé métier relâchée	ticket + produit	8 453 lignes concernées	un article légitimement commandé deux fois

Le niveau 2 est celui que tout le monde exécute, parce que la clé est là, bien nommée, et que le test tient en une formule. Les niveaux 3 et 4 sont ceux qui décident. Retenez la phrase : **une clé primaire propre avec une clé métier collisionnée, ça s'appelle un doublon ; l'inverse (clé primaire en double, métier propre) ça s'appelle un accident d'export**, et ça se répare autrement.

> **Attention.** Un décompte de doublons sans son dénominateur est une annonce, pas un résultat. « 3 440 » peut vouloir dire 1,4 % du fichier ou 40 % d'un magasin ; « 8 453 » est le nombre de lignes impliquées dans une collision, et non le nombre de lignes à retirer — la différence, `5 013` lignes, est exactement ce que l'on perd quand on lit un compteur de paires comme un compteur de lignes.

5.2 Le faux doublon : ce que la répétition a de normal

Sur 146 161 tickets, 97 199 lignes en partagent un autre. Ce n'est pas un défaut : un ticket de quincaillerie a plusieurs lignes, et **78 683 tickets n'en ont qu'une**. Le test de répétition ne dit donc rien tout seul ; c'est la question métier qui tranche : « un même article peut-il être saisi deux fois sur le même ticket, et légitimement ? » Réponse du socle, lue dans sa notice : non — une ligne par article, et la quantité porte le nombre.

> **Définition.** Un **doublon structurel** est une répétition que le modèle de données autorise et que le métier reconnaît : deux lignes d'un même ticket, deux relevés de stock le même jour dans deux magasins. Un **doublon accidentel** est une répétition que le modèle autorise mais que le métier refuse. Les deux ont le même profil informatique ; seul le métier les sépare — d'où la ligne « règle » dans le dictionnaire de données (C06).

5.3 Anatomie des 3 360 : ce que le fichier dit de son accident

On a retiré les lignes du fichier propre, et l'on a regardé ce qu'elles étaient.

Propriété mesurée	Valeur	Ce que ça veut dire
lignes retirées	3 360	l'écart brut − propre, le seul dénominateur honnête
retirées ayant un jumeau conservé sur ticket + produit	3 360	la totalité : le mécanisme est bien la répétition d'un article
montant identique entre jumeaux	3 346	pas un copié-collé intégral : 14 divergent
quantité identique	3 346	les mêmes 14, donc la divergence porte sur le couple quantité-montant
`id_vente` supérieur à celui de l'original	3 360	le doublon est toujours numéroté après
écart médian des `id_vente`	123 285	loin de l'original : ce n'est pas une double frappe
position médiane dans le fichier	99,3 %	un bloc ajouté en fin de fichier

Lecture du détective : un bloc de lignes recopiées, numéroté après tout le reste, collé à la fin du fichier, dont le montant reproduit l'original 3 346 fois sur 3 360. Voilà le portrait-robot d'**un export rejoué sur une période déjà exportée** — pas d'une caisse qui a double-frappé (dans ce cas les `id_vente` seraient contigus, à 838 près au mieux, et les lignes seraient dispersées dans le fichier). Cette lecture ne sert pas à incriminer un collègue : elle dicte la règle de garde.

5.4 La règle de garde : laquelle des deux lignes on conserve

Trois règles se présentent, et elles ne se valent pas :

- **garder la première du groupe** (le plus petit `id_vente`) : déterministe, et alignée sur le mécanisme — l'original est toujours avant. C'est la bonne ici.
- **garder la dernière** : la règle par défaut de beaucoup d'outils de tri « supprimer les doublons », et ici elle conserve les lignes du bloc recopié — le total serait juste d'un cheveu, le détail faux de 3 360 lignes.
- **garder la plus cohérente** (dont `montant_ttc = montant_ht + montant_tva`) : celle qu'il faut ajouter pour les 14 groupes dont les jumeaux divergent, parce qu'entre deux écritures du même article, on conserve celle qui respecte la règle de gestion.

Le contrôle qui valide la règle, c'est le total : réparer les montants texte **et** dédoublonner au niveau 3 en gardant le plus petit `id_vente` rend 15 419 985 157 FCFA — l'écart à la cible est de 0. Réparer sans dédoublonner rendait 15 639 751 286 FCFA, soit +219 766 129. Les deux chiffres, dans cet ordre, sont la démonstration que la règle de garde n'est pas un détail d'implémentation.

> **Conseil professionnel.** Écrivez la règle de garde dans le registre des doublons, avec le nombre de groupes concernés et le nombre de lignes retirées — les deux, pas un. Dans six mois, on ne vous demandera pas « combien de doublons ? » mais « pourquoi celle-là plutôt que l'autre ? », et le seul document qui y répond est le vôtre.
>
> **Définition.** Le **registre des doublons** est la liste nominative de ce qui a été retiré : une ligne par groupe, avec les deux identifiants en présence et la règle qui a départagé. Ce n'est pas un journal de modifications (C06) : il ne décrit pas ce qu'on a changé dans les cellules, il décrit ce qu'on a **choisi d'oublier**.

5.5 Le registre des doublons, premier livrable de traçabilité

Le fichier `doublons_registre.md` (ou `.csv`) contient une ligne par groupe : la clé du groupe, les `id_vente` en présence, la ligne gardée, la ligne retirée, le montant de part et d'autre, la règle appliquée. Trois colonnes sont non négociables : règle, date, outil. Un nettoyage sans registre est

une transformation invisible : il rend impossible la question « est-ce que le chiffre a changé parce que le fait a changé, ou parce que vous avez décidé ? ».

Sur ce socle, le registre fait 3 360 lignes, soit 1,4 % du fichier. Il n'est pas une annexe : il est la pièce qui permet à un repreneur de refaire votre décision — ou de la défaire.

5.6 Les correspondances : quand deux écritures ne font qu'un objet

Changement d'objet, même logique. `produits.csv` contient 15 valeurs de `categorie` ; le dictionnaire de référence, 7. La table de correspondance est donc une table à 15 lignes, dont la colonne norme porte un choix :

Écriture dans le brut	Norme retenue	Motif
`MATÉRIAUX`, `Materiaux`, `materiaux`, `Matériaux`	`Matériaux`	casse, accent absent, espace finale
`PEINTURE`, `peinture`, `Peintures`	`Peinture`	casse, pluriel
`QUINCAILLERIE`, `Quincaillerie`, `quincaillerie`	`Quincaillerie`	casse, espace finale
`Electricité`	`Électricité` ou `Electricité`, **au choix**	accent sur la majuscule initiale : à trancher et à écrire

Appliquer la correspondance ne change **aucun** total : le chiffre d'affaires de la semaine reste le même, que la ligne s'appelle `Peintures` ou `Peinture`. Ce qui change, c'est le nombre de lignes du tableau croisé (15 → 7) et les 44 références dont l'étiquette bouge. C'est la différence fondamentale entre un défaut de **valeur** et un défaut de **correspondance**, et elle explique pourquoi les seconds survivent des années dans les rapports : ils ne rendent aucun total suspect.

> **Définition.** Un **dictionnaire de correspondance** (crosswalk) est une table qui relie les identifiants ou les libellés de deux systèmes. Contrairement à une conversion, elle est **réversible et annotable** : on peut y inscrire « ces deux codes ne désignent pas la même chose », ce qu'aucune formule `RECHERCHEV` ne permet d'écrire.

5.7 L'appariement des clients : la règle qui ne casse pas les homonymes

412 clients du fichier brut ont été jugés doublons et retirés pour obtenir `clients_propres.csv` (23 912 → 23 500). Aucun n'a le même `id_client` (0 doublon sur cette clé), et aucun n'est une copie exacte : ce sont des **quasi**-doublons. Le test qui les trouve tous, sans rien casser :

```
clé d'appariement =  minuscule(nom, sans accents ni ponctuation)  +  "|"  +  chiffres(telephone)
```

- sur le **nom seul** normalisé : 11 570 lignes fusionnables — la règle est inutilisable, elle écrase 11 158 clients qui ont simplement un nom commun (les `Menage 00002` et autres libellés de synthèse) ;
- sur **nom + chiffres du téléphone** : 412, exactement ce que la référence retire, répartis en 412 groupes de 2 lignes, jamais plus ;
- le téléphone seul n'est pas non plus une clé : 23 499 numéros distincts pour 23 912 clients, donc 413 numéros portés par au moins deux lignes.

Où sont les 412 ? Le fichier donne à voir le mécanisme, et il faut le regarder avant d'écrire la règle. Les 412 lignes retirées forment **un bloc continu**, de la ligne 23 501 à la fin du fichier : elles n'ont pas été saisies dans le désordre, elles ont été **ajoutées après coup**, chacune derrière son original (412 fois sur 412). Le jumeau porte le même nom en majuscules, encadré d'espaces (`SARL K` face à `SARL k`), et un téléphone débarrassé de ses espaces (`+2267862218198` face à `+226 78 62 21 81 98`) : 412 fois sur 412 pour les deux anomalies à la fois, **0 fois** sur les lignes

conservées. Autour du couple, tout le reste concorde : ville, courriel, type de client et date de création sont identiques dans les 412 couples (412 sur 412).

Ces trois faits se lisent d'un coup et changent le statut de la règle. D'abord, la clé `nom + chiffres` ne produit **aucun faux positif** : les 412 groupes comptent exactement 2 lignes, jamais davantage. Ensuite, ajouter la ville à la clé n'aurait rien cassé ici — et c'est précisément le piège : une colonne d'accord par hasard n'apporte aucune discrimination, elle n'ajoute que de la fragilité (1 911 clients n'ont pas de ville ; leur ligne aurait été mariée à une autre ligne sans ville). Une colonne n'entre dans une clé que si son **désaccord** doit annuler le rapprochement. Enfin, l'écriture en majuscules et l'espace en trop ne sont pas des bruitages : ce sont les traces d'un import par accumulation : on recopie, on ne fusionne pas. Le C05 les normalise ; ici, elles servent d'abord à reconnaître le bloc.

> **Dans les faits.** Un seuil de similarité « par défaut » (0,85 sur un score de ressemblance de chaîne) appliqué à ce fichier de clients produit un tas de paires qui n'ont rien à voir, parce que les noms sont courts et que le corpus est synthétique : la bonne pratique n'est pas de régler le seuil plus haut, c'est de trouver la **colonne rare** qui discrimine — ici, le téléphone. Un appariement sans colonne discriminante est un tirage.

5.8 Les correspondances manquantes : 486 lignes et 372 clients

Un `id_client` présent dans les ventes et absent des clients n'est pas un doublon, c'est l'autre face du même problème : l'objet n'a pas de fiche. Ici, 486 lignes, sur 372 identifiants distincts. Le traitement est une table, pas une formule : `a_creer.csv` avec, par identifiant, le nombre de lignes, le premier et le dernier magasin, et la mention provisoire. Ce document a trois usages : il justifie le dénominateur de tout tableau par client, il permet de poser la question au bon service, et il empêche la jointure de faire disparaître 486 lignes sans que personne ne sache où elles sont passées.

Le contrôle symétrique doit être écrit lui aussi, avec son zéro : **0 ligne de vente ne pointe un `id_produit` inexistant** (154 références, toutes jointes). Un zéro non écrit est une question ouverte ; un zéro écrit est un risque fermé.

Doublons : ce que chaque niveau de clé voit, et ce qu'il laisse passer

Le même fichier, quatre questions. Une seule tombe sur le bon tas de lignes.

Ligne entièrement identique colonne par colonne, A à Z	**0 ligne**	aucun copié-collé intégral dans le fichier
Clé technique · id_vente 243 360 valeurs distinctes	**0 doublon**	la clé du fichier est vierge de soupçon
Ticket + produit + quantité collision sur la clé métier	**3 440 lignes**	le seul test qui voie le défaut
Ticket + produit, sans quantité toutes les lignes concernées	**8 453 lignes**	le dénominateur à citer, pas le résultat

L'entonnoir, et ce qui reste à traiter :

243 360 lignes → 97 199 partagent un ticket (normal : 1,67 ligne par ticket)
→ 8 453 lignes en collision → 3 440 collisions sur la clé complète
→ 3 360 lignes à retirer → dont 61 aussi en montant texte (le recouvrement)

La trace du mécanisme, lue dans l'ordre des identifiants :

3 360 doublons, tous numérotés après leur original — écart médian 123 285, le plus proche 838
position médiane dans le fichier : 99,3 % du parcours. Un bloc ajouté à la fin :
signal d'un export rejoué sur une période récente, jamais d'une double saisie au clavier.
Règle de garde : le plus petit id_vente du groupe, jamais le dernier.
3 346 fois sur 3 360 le montant est identique des deux côtés ; les 14 autres divergent —
leur règle ne peut pas être « la première venue » mais « la première conforme à ht + tva ».

> **Attention.** Un registre de 3 360 lignes ne se relit pas : il se **requête**. Écrivez-le en table structurée — une ligne par cas, des colonnes nommées — et non en prose, sinon il ne sert à rien le jour où l'on vous demande « quelles lignes avez-vous gardées pour ce ticket ? », question qui, en production, arrive toujours.

6. Exemple concret : le tableau croisé qui accuse un magasin

Un analyste charge le fichier, retire les doublons avec l'outil (bouton « Supprimer les doublons », toutes colonnes cochées, garde la première occurrence), puis livre le CA par magasin. Résultat : le fichier fait toujours 243 360 lignes moins 0, puisque **aucune** ligne n'est une copie intégrale. Son tableau croisé garde donc les 3 360 lignes en trop, et le magasin qui a reçu l'export rejoué affiche un CA surévalué — de 219 766 129 FCFA sur l'ensemble, mais concentré sur le tiers de ses journées. Personne ne voit l'erreur au total (il est gros, comme d'habitude), on la voit dans le classement des magasins : c'est ainsi qu'un défaut de doublon se manifeste en production, rarement par un chiffre aberrant, presque toujours par un **rang** qui n'est pas le bon.

7. Démonstration pas à pas : trouver, décider, journaliser

Feuille `Ventes` (243 360 × 20). Objectif : la liste des groupes, la ligne gardée, la ligne retirée.

1. **Clé métier.** Colonne utilitaire `=Ventes!B2&"|"&Ventes!H2&"|"&Ventes!I2` (ticket | produit | quantité) — à remplir sur les 243 360 lignes, puis à figer en valeurs avant le tri.
2. **Rang dans le groupe.** Colonne `=NB.SI($U$2:$U2;U2)` : il vaut 1 sur la première occurrence, 2 sur la seconde. C'est la règle de garde écrite en clair — et elle est déterministe tant que l'ordre des lignes ne bouge pas, d'où le figeage de l'étape 1.
3. **Compteur de collision.** `=SOMME.SI($U$2:$U$243361;"="&U2;$U$2:$U$243361)` remplacé par un comptage simple : `=NB.SI(Ventes!$U$2:$U$243361;U2)>1` → VRAI sur 8 453 lignes au niveau 4, sur 3 440 × 2 au niveau 3 selon la façon de compter : c'est le moment d'écrire lequel vous publiez, et de le publier avec son dénominateur.
4. **Sélection des retirées.** `=SI(ET(Ventes!$V2>1;ESTNUM(...)))` — en pratique : filtre sur `rang > 1`, copier vers une feuille `RETRAITES`, qui devient le noyau du registre (3 360 lignes attendues).
5. **Vérification arithmétique.** `=243360-3360` → 240 000, le nombre de lignes du fichier propre. Si ce compte n'y est pas, votre règle de garde n'est pas la même que celle du producteur du corrigé — et l'un des deux a tort.
6. **Vérification monétaire.** `=SOMME` des montants réparés sur les lignes conservées → 15 419 985 157 FCFA, écart à la cible : 0. Les deux vérifications (lignes et montants) sont obligatoires : l'une attrape une règle fausse, l'autre une réparation fausse.
7. **Registre.** Sur `RETRAITES`, ajouter quatre colonnes : `id_vente_original`, `motif` (« collision clé métier »), `regle` (« garde du plus petit id_vente »), `date`. Cinq minutes, et le nettoyage devient défendable.

> **À retenir.** Un dédoublonnage se prouve par deux contrôles de sens opposé : le **compte** (243 360 - 3 360 = 240 000 lignes) et le **montant** (15 419 985 157 FCFA, écart 0 à la cible). L'un sans l'autre ne vaut rien : le compte ne voit pas une réparation fausse, le montant ne voit pas une règle de garde fausse — et ce fichier fournit les deux cas, les 14 groupes divergents et les 61 lignes à double défaut.

8. Erreurs fréquentes

Symptôme visible	Cause réelle	Correction
« Aucun doublon : la clé est unique »	seul le niveau 2 a été testé	tester le niveau 3 sur la clé métier ; 3 440 lignes ici
Suppression de 97 199 lignes valides	dédoublonnage sur `id_ticket` seul	définir la répétition inattendue, pas la répétition : 1,67 ligne par ticket est normal
Le total baisse un peu, puis beaucoup	règle de garde « garder la dernière » appliquée à un bloc ajouté en fin	garder le plus petit identifiant, et le justifier par la position dans le fichier
Deux lignes identiques conservées alors qu'un doublon est annoncé	comparaison de chaînes non normalisée (espace finale, accents)	normaliser avant d'appairer ; comparer des normes, pas des textes
« 15 catégories » au lieu de 7 dans le rapport	défaut de correspondance non traité	table de correspondance écrite, jamais de `SI()`. imbriquées
Le repreneur ne retrouve pas les lignes supprimées	pas de registre	registre des doublons avec la règle, la date, l'outil ; 3 360 lignes à consigner

9. Bonnes pratiques professionnelles

> **Boîte à outils du chapitre.** Quatre colonnes utilitaires et rien d'autre : la clé métier (concaténation des colonnes qui définissent l'objet), le rang dans le groupe (un `NB.SI` borné vers le haut), la conformité du montant (`=ET(ESTNUM(N2);ABS(L2+M2-N2)<=1)`), et la position dans le fichier. Avec ces quatre colonnes, le dédoublonnage, la règle de garde et le registre s'écrivent en dix formules, sans macro et sans outil tiers.

- **Écrivez la clé métier avant de chercher les doublons.** Une phrase dans le dictionnaire : « une ligne = un article d'un ticket » — et le test en découle, au lieu d'être deviné par l'outil.
- **Testez les quatre niveaux, publiez les quatre résultats.** Même les zéros : « 0 ligne entière dupliquée » est ce qui prouve que le nettoyage n'a pas détruit une seconde écriture du même article.
- **Une règle de garde = un ordre + un critère + une exception.** Ici : ordre croissant d'`id_vente`, critère montant conforme à ht + tva, exception sur les 14 groupes divergents, traités à la main et notés comme tels.
- **Ne supprimez jamais sans liste.** Le registre est un livrable au même titre que le fichier propre.
- **Séparez le dédoublonnage de la réparation de types.** Les 61 lignes qui cumulent les deux sont le seul cas où l'ordre des opérations change le résultat (C01, § 10) : réparez d'abord, dédoublonnez ensuite.
- **Sur une correspondance, gardez la valeur d'origine à côté.** Une colonne `categorie_source` permet de revenir sur un choix de norme ; l'écraser dans la colonne d'origine rend le débat impossible.
- **Faites valider la norme par le métier, par écrit.** `Electricité` prend-il un accent ? La réponse ne change pas un total et change une identité de marque : c'est une décision de direction, pas de technicien.

10. Exercice guidé — reproduire le fichier propre, puis prouver que c’est le bon (35 min, /10)

Étape 1. Construisez la clé métier et le rang dans le groupe (§ 7, étapes 1-2). Combien de lignes ont un rang supérieur à 1 au niveau « ticket + produit + quantité » ? → **3 360**. Si vous trouvez 3 440, vous avez compté les collisions et non les lignes excédentaires : relisez le § 5.1, la différence est exactement une ligne par groupe.

Étape 2. Appliquez la règle de garde (le plus petit `id_vente`) et comptez les lignes restantes : 243 360 − 3 360 = 240 000. Comparez au fichier propre : le compte y est. Vous n’avez rien « nettoyé au propre », vous avez retrouvé la règle du producteur — la nuance est que la vôtre est maintenant écrite.

Étape 3. Traitez les 14 groupes dont les jumeaux divergent en montant ou en quantité. Décision attendue : garder la ligne dont `montant_ttc = montant_ht + montant_tva`, et si les deux sont conformes, garder la première et l’écrire dans le registre. Ces 14 lignes sont le seul endroit du socle où la règle par défaut donne un résultat différent de la règle métier — donc le seul endroit où un correcteur pourra vous contredire si vous ne les citez pas.

Étape 4. Réparez les 3 421 montants texte (`=SI(ESTNUM(N2);N2;L2+M2)`) **puis** dédoublonnez. Total obtenu : 15 419 985 157 FCFA, écart à la cible **0**. Écrivez l’ordre dans le registre : réparation, puis dédoublonnage. L’inverse ne marche pas — les 61 lignes à double défaut le montrent : traitées dans l’autre ordre, l’une des deux opérations leur laisse un montant vide.

Étape 5. Rédigez les trois lignes du registre qui résument l’opération : 3 360 lignes retirées · règle de garde · deux contrôles (240 000 lignes, écart monétaire 0). **Note /10** : 2 points pour la distinction 3 440 / 3 360, 3 points pour la règle de garde écrite et appliquée, 2 points pour les 14 groupes, 3 points pour le total et son écart à zéro avec l’ordre des opérations justifié.

11. Exercices autonomes

Exercice 3.1 (★) — Les quatre compteurs (10 min). Sur `ventes_brutes.csv`, produisez les quatre résultats du § 5.1 avec, pour chacun, son dénominateur. Écrivez la phrase qui dit ce que seul le niveau 3 a vu.

Exercice 3.2 (★★) — Le faux positif du nom (15 min). Sur `clients.csv`, comptez les lignes fusionnables par (a) `id_client`, (b) nom normalisé seul, (c) nom normalisé + chiffres du téléphone, (d) téléphone normalisé seul. Donnez les quatre nombres, et indiquez celle des quatre règles qui détruirait le plus de clients réels.

Exercice 3.3 (★★) — La table de correspondance (15 min). Écrivez la table à 15 lignes qui ramène les catégories de `produits.csv` aux 7 du dictionnaire, et comptez les références dont l’étiquette change. Vérifiez que le total du fichier de ventes ne bouge pas d’un franc : dites ce que cette invariance prouve, et ce qu’elle ne prouve pas.

Exercice 3.4 (★★) — Les orphelins à créer (12 min). À partir des 486 lignes sans client référencé, produisez `a_creer.csv` : une ligne par identifiant (372), avec le nombre de lignes, le ou les magasins concernés et la mention « provisoire ». Combien de ces identifiants ont une ligne dont le montant est en texte ? (Indice : le recouvrement des deux défauts est publié.)

Exercice 3.5 (★★★) — Le double export reconstitué (25 min). En vous servant uniquement de l’ordre des `id_vente` et de la position dans le fichier, écrivez la règle qui identifie le bloc ajouté en fin, et vérifiez qu’elle retrouve les 3 360 lignes. Donnez ensuite le CA contenu dans ce bloc. Contrainte : votre règle ne doit contenir ni liste de lignes ni copié-collé, seulement des comparaisons — sinon elle ne rejoue pas le mois prochain.

12. Correction détaillée

Exercice 3.1. Niveau 1 : 0 ligne entièrement dupliquée sur 243 360. Niveau 2 : 243 360 `id_vente` distincts, donc 0 doublon de clé. Niveau 3 : 3 440 lignes en collision sur ticket + produit + quantité. Niveau 4 : 8 453 lignes impliquées dans une collision ticket + produit, dont 7 640 avec la même quantité. La phrase attendue : « le défaut est invisible à la clé primaire et aux copies intégrales ; il n'apparaît que lorsque la répétition est jugée sur ce qui définit l'objet — un article par ticket — et il faut alors retirer 3 360 lignes, pas 3 440 ».

Exercice 3.2. (a) 0 doublon : chaque ligne a son identifiant. (b) 11 570 lignes « en trop » par nom normalisé : règle inutilisable, elle retirerait 11 158 lignes de trop (11 570 − 412). (c) 412 : la règle défendable, exactement ce que le référentiel retire (23 912 − 23 500). (d) Par téléphone normalisé seul : 413 numéros portés par au moins deux lignes — 23 499 numéros distincts pour 23 912 clients. C'est (b) qui détruit le plus, et c'est précisément la règle que propose un outil « dédoublonnage intelligent » si on ne lui donne que le nom. La lecture à écrire : ni la clé technique (trop stricte : elle ne voit rien) ni le nom (trop lâche : il voit trop), mais le **couple** nom + chiffres du téléphone, chaque membre du couple étant insuffisant seul.

Exercice 3.3. La table : 15 lignes, colonnes `ecriture_brute`, `categorie_normale`, `motif` (casse / accent / espace / pluriel), `date`, `valide_par`. Les références dont l'étiquette change : **44 sur 154**. Le CA reste 15 419 985 157 FCFA avant et après correspondance, à 0 près — l'invariance prouve que la correspondance ne touche pas aux faits ; elle ne prouve pas que le **classement** est bon : c'est justement lui qui change, de 15 lignes à 7, et donc tous les pourcentages et tous les « top 3 ». Conclusion à écrire : un défaut de correspondance est un défaut de lecture, pas un défaut de montant, et il se débusque sur les tableaux, jamais sur le total.

Exercice 3.4. 372 lignes dans `a_creer.csv` ; le total des lignes concernées est 486, soit 0,2 % du fichier — faible en volume, décisif sur un classement par client. Le recouvrement avec le défaut de type est de **61** lignes : ce sont 61 des 486 orphelines dont le montant est aussi en texte, donc 61 fiches « à créer » qui ne seront pas chiffrables avant réparation du type. D'où l'ordre : réparer les types, puis créer les fiches, puis contrôler la jointure. Le pigiste qui fera l'inverse aura créé 372 fiches dont il faudra reprendre 61.

Exercice 3.5. Règle qui marche : « une ligne est dans le bloc ajouté si son `id_vente` est supérieur au plus petit `id_vente` d'un groupe en collision, et qu'elle est en position supérieure à 99 % du fichier » — mesurée, elle retrouve les 3 360, avec un écart médian de 123 285 sur les identifiants et minimum 838. Le CA du bloc est exactement l'écart constaté entre le fichier réparé non dédoublonné et la cible : 15 639 751 286 − 15 419 985 157 = **219 766 129 FCFA**. Deux remarques à intégrer au corrigé : la règle marche parce que le doublon a été ajouté en bloc — un dédoublonnage écrit sur la position dans le fichier est une règle fragile, qui ne survit pas à un réordonnancement ; et le contrôle de cohérence entre les deux approches (position, montant) est la preuve qu'il ne s'agit pas d'un artefact d'écriture. La version robuste, celle qu'on livre, est la clé métier plus la règle de garde : la position ne sert qu'à comprendre, pas à décider.

13. Mini-projet M04.P3 — « Le dossier de dédoublonnage » (1 h 30)

Livrables, tous exigés, aucun optionnel :

1. `cle_metier.md` : la définition écrite de l'objet (une ligne = ?), la ou les clés candidates, la raison du rejet des autres — avec les compteurs des quatre niveaux ;
2. `doublons_registre.csv` : les 3 360 lignes retirées, leurs jumeaux conservés, la règle, la date, et la colonne `divergent` (les 14 groupes) ;
3. `a_creer.csv` : les 372 clients sans fiche, avec leur volume de lignes (486) et le drapeau texte (61) ;

4. `correspondances.csv` : les 15 écritures de catégorie et leur norme, plus les deux écritures de téléphone (espacée / collée) traitées par la même table ;
5. `controles.md` : les trois vérifications de fin — 240 000 lignes, total 15 419 985 157 FCFA, écart 0 — avec la formule ou l'outil qui les a produites ;
6. une page de décision : ce que vous auriez fait de différent si le bloc dupliqué avait été dispersé dans le fichier au lieu d'être en fin.

Barème (10 points). Clé métier écrite et défendue (2) · registre complet, règle de garde explicite (3) · correspondances et `a_creer` utilisables par un tiers (2) · contrôles chiffrés avec méthode (2) · page de décision (1). Seuil de validation : 7. Un rendu sans la page de décision plafonne à 8 : la capacité à dire ce qui changerait est la compétence, pas le compteur.

14. Résumé du chapitre

```
CE QU'ON CHERCHE          TEST                      RÉSULTAT DU FICHIER     DÉCISION
copie intégrale    égalité colonne à colonne          0 ligne              rien à retirer
clé technique      id_vente distincts                 243 360 / 243 360    la clé ne dit rien
clé métier         ticket + produit + quantité        3 440 en collision    3 360 à retirer
jumeaux clients    nom normalisé + chiffres tél.      412 groupes de 2      412 à retirer
correspondance     15 écritures → 7 normes            44 sur 154            total inchangé,
orphelins          id_client hors référentiel         486 lignes / 372      classement faux
```

Deux leçons se tiennent par la main. La première : **le doublon se définit, il ne se voit pas** — et sa définition est une phrase sur le monde (« une ligne par article sur un ticket »), pas une option de menu. La seconde : la correspondance est le même problème vu de l'autre côté, deux écritures pour un objet — avec cette différence de poids qu'elle laisse le total intact, ce qui la rend invisible au contrôle le plus courant. Le dénominateur commun des deux, c'est le registre : ce que vous avez fusionné, ce que vous avez retiré, pourquoi, avec quel ordre d'opérations.

15. À retenir

> **À retenir. ★ Une clé primaire propre n'innocente pas le fichier.** 243 360 identifiants distincts et 3 360 lignes en trop : le doublon se cherche sur la clé métier, sinon il ne se trouve pas. **★ Retirer, ce n'est pas compter les collisions.** 8 453 lignes impliquées, 3 440 en collision, 3 360 à retirer : les trois nombres sont vrais, un seul est une liste d'action. **★ La règle de garde se justifie par le mécanisme.** Tous les doublons portent un `id_vente` supérieur à celui de leur original (écart médian 123 285) et siègent à 99,3 % du fichier : on garde le plus petit identifiant, et on l'écrit. **★ Réparer avant de dédoublonner.** 61 lignes cumulent les deux défauts ; dans l'autre ordre, elles ressortent avec un montant vide et le total ne se réconcilie plus (écart de +219 766 129 FCFA). **★ Une correspondance ratée ne casse pas le total, elle casse le classement.** 15 catégories au lieu de 7, 44 références à retoucher : aucun contrôle de montant n'aurait signalé cela, seul le regard sur le tableau l'a fait.

16. Évaluation formative (auto-correction, 8 min)

Q1. Un fichier a une clé primaire sans doublon et 0 ligne entièrement dupliquée. Combien de lignes en trop peut-il contenir ? (a) 0 ; (b) autant que le permet une collision sur la clé métier ; (c) impossible à savoir sans la table clients. → **(b)**, et l'exemple vaut 3 360 lignes.

Q2. Pourquoi ne pas dédoublonner sur `id_ticket` ? → parce que 97 199 lignes partagent légitimement un ticket (1,67 ligne par ticket, maximum 6) : la répétition est structurelle, pas accidentelle. Le test utile ajoute le produit, et la quantité pour trancher les cas ambigus.

Q3. Deux lignes d'un même groupe ont des montants différents (14 groupes sur 3 360). Quelle ligne garder ? → celle dont le montant satisfait `montant_ht + montant_tva = montant_ttc` ; à égalité, la plus petite en `id_vente`, **et on le note dans le registre**. Une règle silencieuse n'est pas une règle.

Q4. Vrai ou faux : « une table de correspondance modifie les montants ». → **Faux.** Elle modifie les regroupements. C'est pourquoi elle se valide sur les classements et sur les effectifs (154 références, 44 étiquettes changées), jamais sur le total.

Q5. Vous devez livrer le nettoyage à un repreneur. Quels artefacts, dans quel ordre ? → dictionnaire de correspondance, registre des doublons (3 360 lignes, règle, date), liste des orphelins à créer (372), et les trois contrôles finaux avec leur formule — avant le fichier lui-même, qui se régénère.

Question ouverte (la plus importante). Reprenez la règle « garder la ligne en position la plus haute » (le réflexe de beaucoup d'outils de tri) et démontrez, chiffres à l'appui, ce qu'elle aurait produit ici. Corrigé attendu : elle conserve le bloc recopié de fin de fichier, donc le **même nombre** de lignes (240 000) et un total identique puisque les montants sont les mêmes 3 346 fois sur 3 360 — mais les 14 divergents se retrouvent du mauvais côté, et les détails par ticket sont faux. La leçon n'est pas que la règle est fausse au total : c'est qu'un nettoyage peut être juste sur les deux seuls contrôles que tout le monde fait (nombre de lignes, somme) et faux partout ailleurs. Voilà pourquoi le registre existe.

Module M04.C04 — Valeurs aberrantes et règles de contrôle : une règle doit rendre une décision, pas un nombre

Outil de ce chapitre : le tableur, sur `ventes_brutes.csv`, `produits.csv` et `stocks_quotidiens.csv`. Durée indicative : 5 h. Niveau : N2.

> **L'idée du chapitre.** On vous demandera de « repérer les valeurs aberrantes », et l'outil répondra avec une formule toute faite : quartiles, écarts-types, seuils à trois sigmas. Appliquée à notre fichier de ventes, cette recette signale **24 180 lignes** de prix — 9,9 % du fichier — et ne désigne pas une seule erreur. La règle qui voit juste s'appuie sur le référentiel des produits, et elle signale **0** violation : ce zéro est une preuve, pas une paresse. Le chapitre garde un deuxième tour de vis, plus désagréable : sur les 973 lignes où la remise est aberrante, la réparation évidente ne répare rien — diviser par 100 laisse 258 lignes hors du domaine contractuel, parce que la valeur fausse n'a jamais été la valeur juste multipliée par cent. Une valeur aberrante n'est pas une valeur éloignée : c'est une valeur **incompatible avec un document**.
>
> **Base de travail.** `01_socle_donnees/data/brut/ventes_brutes.csv` (243 360 lignes × 20 colonnes), `01_socle_donnees/data/brut/produits.csv` (154 références), `01_socle_donnees/data/brut/stocks_quotidiens.csv` (60 000 lignes), `01_socle_donnees/data/reference/verites_terrain.csv` (15 419 985 157 FCFA).

---

1. Objectifs du chapitre

À la fin de ce chapitre, vous saurez :

- **classer** une anomalie en trois natures — impossible, invraisemblable, licite-surprenante — et traiter chacune autrement ;
- **écrire** une règle de contrôle qui renvoie une décision : bloquer, corriger, surveiller, ignorer tracé ;
- **calibrer** un seuil sur deux preuves — la distribution observée et le document métier — et dater cette calibration ;
- **préférer** le contrôle croisé (colonne contre référentiel) au contrôle univarié, en sachant ce qu'il ne dit pas ;
- **mettre à l'épreuve une réparation** : avant d'appliquer `÷100`, vérifier qu'elle ramène les valeurs dans le domaine.

2. Pourquoi cette notion est importante

- Les prix unitaires du fichier vont de 575 à 45 000 FCFA, médiane 6 400, quartiles 3 650 et 11 050, écart-type 8 372. La règle « hors de [Q1 − 1,5 IQR ; Q3 + 1,5 IQR] » marque **24 180 lignes** ; « au-delà de 3 σ », **3 894** ; « 6 σ », **0**. Aucune de ces lignes n'est une erreur : le magasin vend du sac de ciment et de l'ossature, et la dispersion est le métier.
- La règle juste est relative, pas absolue : le prix d'une ligne doit rester dans la bande négociée autour du catalogue. Ratio observé : **0,958 à 1,306** ; violations d'une bande [0,95 ; 1,35] : **0**. En revanche, exiger l'égalité au franc avec le catalogue signaler **232 981 lignes** sur 152 produits des 154 — la règle est fausse, pas le commerce.

- Le défaut le plus instructif du socle ne casse **aucun** total : les 973 lignes à remise aberrante ont un `montant_ht` correct et une `taux_remise` fausse (la TVA, elle, rend 18 % du HT sur les 243 360 lignes, à l'unité près : 0 écart). Un contrôle de cohérence des montants ne verra jamais ce défaut ; seule la règle de domaine le voit — et il ne coûte que des analyses de remise fausses.
- Enfin, la règle d'alerte fournie avec les stocks est un cas d'école d'aberration de contrôle : `qte_physique < seuil_alerte` déclenche sur 1 800 lignes, exactement comme `qte_physique < 0`. Le seuil ne filtre rien.

3. Explication simple

Un médecin ne réagit pas de la même façon à une tension à 90, à 220 ou à zéro. Il distingue l'**impossible** (on réanime, on ne débat pas), l'**invraisemblable** (on soupçonne le brassard avant de traiter) et le **licite surprenant** (90 chez un coureur : on note). Les données ont la même échelle, et l'erreur du débutant est de tout traiter au milieu — parce que c'est là que se trouvent les seuils par défaut des outils.

La seconde différence avec la médecine est dans la conclusion : un bilan rend une **conduite à tenir**, pas un pourcentage de patients hors normes. Un contrôle de données doit rendre la même chose : bloquer la ligne, corriger par la règle, surveiller le lot, ignorer avec trace. Si votre règle n'autorise aucune de ces quatre phrases, ce n'est pas encore une règle : c'est un compteur, et un compteur ne protège personne.

> **Définition.** Une **règle de contrôle** est une affirmation vérifiable sur une donnée, assortie d'une décision et d'un destinataire : « le prix d'une ligne reste dans la bande négociée du catalogue, sinon la ligne est retournée au magasin pour validation ». Les trois membres — affirmation, décision, destinataire — sont obligatoires ; en production, deux sur trois ne suffisent jamais.

4. Vocabulaire essentiel

Français — English	Définition simple	Piège à éviter
Valeur aberrante — Outlier	Valeur qui surprend au regard du contexte.	La confondre avec une erreur : une surprise peut être parfaitement vraie.
Impossible — Invalid	Contredit une loi du domaine (remise hors [0 ; 1], quantité négative hors retour).	La laisser passer : c'est la seule catégorie qu'on bloque à l'entrée.
Invraisemblable — Implausible	Compatible avec le domaine, mais contredit un document voisin.	La corriger sans preuve : on remplace un fait possible par une moyenne.
Licite surprenante — Legitimate extreme	Hors distribution, juste dans le métier.	La supprimer « parce qu'elle fausse la moyenne ».
Règle univariée — Univariate rule	Contrôle porté par une seule colonne (seuil, IQR, z-score).	Son incapacité à voir les défauts de relation entre colonnes.
Contrôle croisé — Cross-field rule	Confronte deux colonnes, ou une colonne et un référentiel.	L'écrire sur une jointure non validée : l'orphelin devient une anomalie fantôme.
Test de réparation — Repair plausibility check	Vérifier que la correction proposée ramène la valeur dans le domaine.	Appliquer `÷100` sans vérifier : ici, 258 lignes restent hors domaine après « réparation ».
Échelle de gravité — Severity ladder	Bloquant · corrigeable · surveillé · ignoré tracé.	Un seul niveau : tout devient urgent, donc plus rien ne l'est.
Taux de déclenchement — Alert rate	Part des lignes signalées par une règle.	Un taux de 9,9 % : c'est un descripteur de colonne, pas une alerte.

Français — English	Définition simple	Piège à éviter
Faux négatif silencieux — Silent false negative	Le défaut que la règle ne pouvait pas voir.	Croire qu'aucune alerte prouve l'absence de défaut.

5. Cours approfondi

5.1 Les trois natures, comptées sur le fichier

Nature	Test sur le socle	Lignes	Lecture
Impossible	`quantite ≤ 0` hors retour	0	les 2 848 quantités négatives sont toutes des retours signalés
Impossible	`taux_remise < 0`	0	aucune remise négative
Impossible	`montant_tva ≠ 18 % du HT`	0	l'arithmétique fiscale tient sur les 243 360 lignes
Impossible	`id_produit` absent du catalogue	0	aucune ligne sans produit référencé
Invraisemblable	`montant_ht ≠ quantite × prix × (1 − remise)`	973	exactement le lot des remises > 1
Licite surprenante	`quantite > 50`	800	commandes de chantier, plausibles en quincaillerie
Licite surprenante	`poids_kg > 5 000`	7 674	sacs palettisés et ossatures, médiane 9,5 kg par unité

Quatre zéros d'abord, puis un nombre, puis deux. Cet ordre est le plus rentable qui soit : établir ce qui est impossible et vérifier qu'on n'en trouve pas, puis chercher ce qui contredit un document, enfin qualifier ce qui n'est qu'étonnant. Un diagnostic qui commence par le troisième étage produit des rapports de 24 180 lignes signalées et 0 ligne corrigée — c'est-à-dire un rapport qui se jette.

> **Attention.** Le nombre de lignes signalées ne mesure pas la qualité d'une règle. « Prix différent du catalogue au franc près » signale 232 981 lignes et 152 des 154 produits : c'est la règle qui est fausse, pas la moitié du commerce. Une règle se juge à trois choses : ce qu'elle attrape, ce qu'elle laisse passer, et ce qu'elle oblige à faire.

5.2 Quatre familles de règles, et leur rendement mesuré

- **Le domaine** — `remise ∈ [0 ; 1]` : **973 lignes**, maximum lu 30. C'est la seule famille qui n'ait **aucun paramètre à justifier** : la borne vient de la définition de la variable. Premier contrôle à écrire, et le seul qui se défende tout seul.
- **Le seuil métier** — `quantite ≤ 50` : **800 lignes**. Il exige un document (politique d'achat, bon type). Sans document, c'est un IQR déguisé, avec un chiffre rond de plus.
- **La bande relative au référentiel** — prix de la ligne dans [0,95 ; 1,35] du catalogue : **0 violation**, ratio observé 0,958 à 1,306. Le zéro a une valeur parce que la bande est serrée et que les extrêmes sont cités.
- **La cohérence entre colonnes** — `poids_kg = quantite × poids_unite_kg` : **8 380 lignes**, à instruire. C'est le seul des quatre qui signale un doute sans désigner le coupable : la colonne poids, le poids unitaire du catalogue, ou la quantité.

> **Définition.** Un **test de réparation** consiste à appliquer la correction en simulation et à repasser la règle de domaine dessus. Les 973 remises à 30, 25, 18, 12 divisées par 100 donnent 0,30 ; 0,25 ; 0,18 ; 0,12 — et **258 lignes restent hors du domaine** (la remise maximale contractuelle est 0,25). Une réparation qui ne fait pas passer le test n'est pas une réparation, c'est un changement de conventions fait à l'aveugle.

5.3 Ce que le fichier dit vraiment des 973 lignes

Reconstituons la remise depuis les montants : `remise = 1 − montant_ht / (quantite × prix)`. Résultat sur les 973 lignes : étendue 0,000 à **0,110**, et **973 sur 973** dans le domaine [0 ; 0,25]. Autrement dit :

- le `montant_ht` est **juste** (il est cohérent avec la TVA, donc avec le total, donc avec le chiffre d'affaires) ;
- la colonne `taux_remise`, elle, ne décrit plus rien de mesurable : ni 30 %, ni 0,30, ni un montant de remise ;
- 630 de ces lignes ont même un `montant_ht` strictement égal au prix plein, ce qui aurait été impossible avec une remise de 12 à 30 % appliquée.

> **Attention.** « Le total est bon » n'est pas un contrôle, c'est une coïncidence tranquille. Les 973 lignes ont un `montant_ht`, une TVA et un total parfaitement cohérents : vos rapprochements de chiffres d'affaires les plus soignés ne peuvent rien voir. Un contrôle de total vérifie une **égalité entre colonnes**, pas la véracité d'une colonne de taux — et c'est pour cette raison que le registre doit couvrir les colonnes déclaratives, pas seulement les colonnes monétaires.

La leçon déborde largement cet exemple. **Le défaut le plus grave d'un fichier n'est pas forcément celui qui change les totaux** : ceux-là ne changent aucun montant et rendent toutes les analyses de remise fausses — le genre de défaut qui survit des années parce que le contrôle de cohérence des totaux est vert. D'où deux réflexes : (1) une règle de domaine sur chaque colonne déclarative, pas seulement sur les montants ; (2) avant de « corriger » une valeur, chercher si le fichier permet de la **reconstituer** — ici oui, et la reconstitution tient sur 973 lignes sur 973, ce qui n'est pas le cas de la division par cent.

> **Conseil professionnel.** Ne dites jamais « j'ai corrigé les remises ». Dites : « j'ai recalculé la remise à partir des montants, la valeur saisie ne pouvant plus être reliée à rien ; 973 lignes concernées, toutes revenues sous le plafond de 0,25 ». La première phrase est une opération, la seconde est une décision — et une seule des deux se défend devant un auditeur.

5.4 L'échelle de gravité, appliquée au socle

Niveau	Action	Défaut affecté ici	Ce que ça coûte
1 · bloquant	la ligne n'entre pas dans la base	aucun (0 impossible)	—
2 · corriger par la règle	recalcul documenté, puis recontrôle	3 421 montants texte · 973 remises à reconstituer	2 111 789 FCFA · toutes les analyses de remise
3 · surveiller	on garde, on suit, on interroge	8 380 poids incohérents · 800 quantités > 50	un devis de transport faussé
4 · ignorer, tracé	on écrit pourquoi on ne touche pas	51 dates après le 31/08/2026, avant d'être datées	une comparaison annuelle faussée

Le premier niveau est vide sur ce fichier : le producteur a validé les types à l'entrée, et l'écrire est utile — ça disculpe le fichier et charge le contrôle. Le niveau 4 est celui qu'on saute, et c'est le plus coûteux : ignorer sans tracer, c'est transformer une date de 2026-12 en une hausse artificielle lors de la comparaison annuelle.

> **Définition.** Un **registre de contrôles** est la table qui énonce, pour chaque colonne ou groupe de colonnes : la règle, son niveau, sa cadence, son destinataire, son taux de déclenchement attendu et sa mesure du jour. C'est l'objet que laisse un analyste qui part — la liste des règles, pas la liste des défauts.

5.5 Une règle rend une décision : la forme à écrire

```
règle       : le prix unitaire d'une ligne reste dans la bande [0,95 ; 1,35] du prix catalogue
destination : magasin émetteur, courriel hebdomadaire, liste nominative des lignes
niveau      : 3 · surveiller — le catalogue n'est pas la facture
exception   : remises contractuelles hors bande => liste de produits à joindre au contrôle
mesure      : 0 violation · ratio observé 0,958 à 1,306 · 243 360 lignes passées · calibré le
[date]
```

Cinq lignes, et n'importe qui peut reprendre le contrôle six mois plus tard, savoir s'il a dérapé et à qui il s'adresse. Comparez avec « j'ai appliqué le seuil IQR sur les prix, il y a 24 180 outliers » : même travail de calcul, zéro exploitabilité — et, dans un an, quelqu'un remettra cette règle en production.

5.6 Calibrer un seuil : deux preuves, pas une habitude

Toujours le même mouvement. D'abord la **distribution** : 575 à 45 000 FCFA, médiane 6 400, quartile 3 à 11 050, écart-type 8 372 — elle dit où est le corps du fichier et où commencent les queues. Ensuite le **document** : politique commerciale, contrat de marge, seuil de commande de gros. Calibré sur la seule distribution, un seuil est descriptif ; calibré sur le seul document, il est inapplicable s'il contredit la pratique. Ici les deux se rejoignent sur la bande de prix, et se contredisent sur le poids : 7 674 lignes dépassent `5 000` kg alors que la médiane de poids unitaire est de 9,5 kg — le désaccord ne se tranche pas dans le fichier, il s'écrit au niveau 3 et se porte au magasin.

> **Dans les faits.** Un seuil se périme plus vite qu'une règle de domaine. La bande de prix survivra à l'année ; « quantité ≤ 50 » mourra à l'ouverture d'un dépôt ou d'un compte chantier. Écrivez la **date de calibration** à côté du seuil : c'est ce qui permet de dire, deux ans plus tard, que le bruit vient du seuil et non des saisies.
>
> **Définition.** Deux règles sont **redondantes** quand elles signalent exactement le même jeu de lignes. Le test coûte une formule — la somme du produit des deux drapeaux — et rapporte gros : une règle redondante est une règle à supprimer, parce qu'elle double le volume d'alertes sans jamais rien apprendre. C'est le cas, ici, de `qte_physique < seuil_alerte` et `qte_physique < 0` sur les stocks.

5.7 Retourner une règle qui hurle : le cas des stocks

`stocks_quotidiens.csv` fournit `qte_theorique`, `qte_physique`, `seuil_alerte` sur 60 000 lignes. La règle livrée, `qte_physique < seuil_alerte`, déclenche sur 1 800 lignes ; `qte_physique < 0` déclenche sur 1 800 lignes aussi. La coïncidence dit l'essentiel : le seuil n'ajuste rien, c'est le signe

qui décide. Trois sorties, et il faut en choisir une — le pire étant de laisser les deux règles en parallèle, ce qui double le bruit pour un seul message :

- **scinder** : intégrité (`qte_physique < 0`, 1 800 lignes, niveau 2, à instruire) et risque (`qte_theorique − qte_physique > k`, à calibrer sur l'écart observé : maximum 532, moyenne 2,77) ;
- **relever le seuil** pour ramener le taux d'alerte à ce que l'exploitation peut traiter par jour ;
- **assumer une alerte large mais hiérarchisée** : les plus gros écarts d'abord, le reste en liste.

Le point n'est pas la valeur de `k` : c'est qu'un contrôle se lit aussi à son **taux de déclenchement**. 3 % d'alerte sur un stock quotidien, ça se traite. 9,9 % sur une colonne de prix, c'est une description du fichier, pas un contrôle.

5.8 Les règles statistiques, remises à leur place

Règle	Paramètre	Sur `prix_unitaire_ht`	Lignes signalées	Erreurs désignées
Fourchette inter-quartiles	1,5 × IQR	`Q1` 3 650 · `Q3` 11 050	24 180	0
Écart-type	3 σ	σ 8 372	3 894	0
Écart-type	6 σ	idem	0	0
Bande relative au catalogue	[0,95 ; 1,35]	ratio 0,958 → 1,306	0	0, et c'est un résultat

Trois de ces quatre lignes disent la même chose : la distribution des prix est large et saine. Ce que les règles statistiques font bien, ce n'est pas contrôler, c'est **surveiller un changement de nature** : le nombre de lignes hors IQR, suivi hebdomadairement, devient un bon indicateur le jour où un produit change de gamme ou où une unité de mesure dérape. Tant qu'il reste à 24 180, il ne signale rien ; s'il double, il signale que le fichier n'est plus le même — pas qu'il est sale.

Quatre familles de règles — rendement mesuré sur 243 360 lignes

Le nombre de lignes signalées ne dit rien de la qualité de la règle : la deuxième colonne si.

Famille	Règle	Signalées	Erreurs réellement désignées	Décision
Domaine	remise dans [0 ; 1]	**973 lignes**	973 erreurs désignées	2 · corriger par reconstitution
Seuil métier	quantité au-dessus de 50	**800 lignes**	0 erreur — à surveiller	3 · liste au magasin
Bande au référentiel	prix dans [0,95 ; 1,35]	**0 ligne**	0 : le zéro est un résultat	3 · indicateur de dérive
Statistique (IQR)	Q1 - 1,5 IQR à Q3 + 1,5 IQR	**24 180 lignes**	0 erreur — bruit pur	retirer du registre
Égalité au catalogue	prix = prix affiché	**232 981 lignes**	0 erreur — règle fausse	retirer du registre

Le contrôle qui résiste à tout le monde : la réparation avant de corriger

973 remises à 12, 18, 25 ou 30. Divisées par 100 : 258 lignes restent hors du plafond de 0,25.
Reconstituées depuis les montants : 0,000 à 0,110, les 973 dans le domaine. Le montant, lui, n'a jamais été faux.

6. Exemple concret : la ligne à 75 662 kg

Une ligne pèse 75 662 kg quand la médiane du fichier est de 9,5 kg. Réflexe outillé : alerte, exclusion, « nettoyage ». Regardons la ligne : sa quantité, le poids unitaire du produit au catalogue, et le calcul.

- Si `quantite × poids_unite` rend le même poids, la ligne est **juste** : une palette d'enduit ou un chargement de ciment pèse, et le magasin a saisi une quantité, pas une erreur. Rien à faire, sinon s'assurer que le destinataire du devis de transport lit bien des poids totaux.
- Si le calcul contredit, on a une ligne à instruire : quantité fausse, ou poids unitaire du catalogue faux. Et dans ce second cas, ce ne sont pas 8 380 lignes à reprendre mais **une colonne de référence** — la discussion change d'interlocuteur, et de service.

C'est toute la supériorité du contrôle croisé sur le contrôle univarié : le premier dit quelle table est suspecte, le second dit seulement quelle ligne est grosse.

7. Démonstration pas à pas : la grille de contrôles en une heure

Feuille `Ventes` (243 360 × 20), feuille `Produits` (154 × 11), feuille `CONTROLES`.

1. **Joindre le référentiel.** Deux colonnes utilitaires par `RECHERCHEX` : `prix_ref` et `poids_ref` depuis `Produits`. Contrôler le joint d'abord : `=ESTNA(prix_ref)` sommée doit rendre 0 — et il rend 0 sur les 243 360 lignes. Un contrôle croisé assis sur une jointure cassée produit un faux positif de masse.
2. **Ratio et bande.** `=J2/prix_ref2`, puis `=SI(OU(r<0,95;r>1,35);1;0)` sommée → 0. Écrivez à côté les deux extrêmes observés (0,958 et 1,306) : ce sont eux qui justifient la bande, et c'est ce que regardera le repreneur.
3. **Domaine.** `=NB.SI(K2:K243361;">1")` → 973. Puis, au lieu de corriger : `=1-(L2/(I2*J2))` recopée, et `=NB.SI(reconstituée;">0,25")` → **0**. La reconstitution tient ; `=NB.SI(K2:K243361/100;">0,25")` aurait rendu
4. Gardez les deux lignes dans la fiche : c'est la démonstration, pas l'opération.
5. **Seuil de saisie.** `=NB.SI(I2:I243361;">50")` → 800. Niveau 3, liste au magasin.
6. **Cohérence physique.** `=SI(ABS(R2-(I2*poids_ref2))>1;1;0)` sommée → 8 380, avec une colonne de plus que d'habitude : « table suspecte ? » — le contrôle ne le dit pas, l'enquêteur si.
7. **Taux de déclenchement.** Une colonne par règle : alertes ÷ 243 360. 973 à 0,4 %, 8 380 à 3,4 %, 24 180 à 9,9 %. C'est ce tableau qui se montre en réunion, et la dernière ligne est une règle à retirer du registre.
8. **Fiche.** Une ligne par règle : règle · niveau · destinataire · cadence · taux attendu · mesure · date. Douze règles maximum : au-delà, la fiche n'est pas lue, et un contrôle non lu est un contrôle qui ne protège pas.

> **À retenir.** Une règle se juge d'abord à son taux de déclenchement et à la décision qu'elle attache à une alerte. 0 ligne signalée avec une bande justifiée vaut mieux que 24 180 avec un quartile ; et 1 800 alertes qui ne distinguent pas le négatif du risqué ne protègent personne.
>
> **Boîte à outils du chapitre.** Cinq colonnes utilitaires et rien d'autre : le ratio au référentiel, la conformité arithmétique (`=ET(ESTNUM(N2);ABS(L2+M2-N2)<=1)`), le domaine (un `ET()` borné), la reconstitution (la valeur recalculée depuis ce qui est sûr), et le test de réparation (la correction simulée repassée au domaine). Avec cela, les douze règles usuelles d'un fichier de ventes s'écrivent en une matinée, sans macro et sans outil tiers.

8. Erreurs fréquentes

Symptôme visible	Cause réelle	Correction
« 24 180 valeurs aberrantes dans les prix »	seuil statistique sur une colonne à dispersion structurelle	bande relative au catalogue (0 violation) ; la règle statistique devient un indicateur de dérive
« 232 981 lignes ne respectent pas le catalogue »	le catalogue n'est pas le contrat	bande de ratio, jamais l'égalité au franc
« J'ai divisé les remises par 100, c'est corrigé »	la valeur fausse n'était pas la valeur juste × 100	test de réparation : 258 lignes restent hors domaine ; reconstituer depuis les montants
Alerte quotidienne sur la même colonne, sans suite	seuil qui ne discrimine pas (`qte < seuil` ≡ `qte < 0`)	scinder intégrité et risque ; calibrer sur l'écart observé (max 532, moyen 2,77)
Le contrôle ne signale plus rien depuis des mois	jointure cassée : les lignes orphelines sortent du périmètre	contrôler le taux de jointure avant le taux d'alerte (486 lignes, 372 clients)
Les anomalies sont « lissées » pour faire taire l'alerte	une valeur licite écrasée par une moyenne	niveau 3 = instruire, jamais réécrire ; conserver la colonne `_source`
Un seul niveau de gravité dans toute la fiche	tout devient urgent, donc rien ne l'est	quatre niveaux, un destinataire par niveau, une cadence

9. Bonnes pratiques professionnelles

- **Les règles de domaine d'abord, les seuils ensuite.** Elles ne demandent aucune justification statistique et ne bougent pas avec les saisons : remise en proportion, quantité entière, date renseignée, taux de TVA unique.
- **Une règle, un destinataire nommé.** « Magasin émetteur, chaque lundi » devient une habitude en un mois ; une alerte sans destinataire devient du bruit en trois semaines.
- **Écrivez le taux attendu avant la règle.** « Doit signaler entre 0,1 % et 0,5 % des lignes » : au-delà, c'est la règle qui a dérapé. La phrase est à écrire avant, sinon on la réécrit après coup pour coller au résultat.
- **Ne supprimez jamais une ligne pour faire taire une alerte.** Vous ignorez encore si l'erreur est dans la ligne ou dans la table de référence — c'est précisément ce que cherche l'étape 5 de la démonstration.
- **Testez toute réparation en simulation**, puis revalidez le domaine et la cohérence. Une correction qui fait passer un `> 1` en un `> 0,25` n'a rien réparé, elle a changé de convention sans autorisation.
- **Gardez la valeur d'origine dans une colonne `_source`** et la valeur retenue à côté : les deux sont des données ; la seconde sans la première est une réécriture de l'histoire.
- **Un contrôle croisé sans contrôle de jointure est un mensonge partiel.** Comptez les lignes qui n'ont pas trouvé leur référence (0 côté produits, 486 côté clients) avant de publier le résultat du contrôle.
- **Testez une règle en l'inversant.** Serrez la bande à ±0,1 % : 232 981 lignes. L'absurdité est plus parlante que n'importe quel discours de robustesse, et elle se démontre en une formule.

10. Exercice guidé — la règle de prix, de la distribution à la décision (35 min, /10)

Étape 1. Décrivez `prix_unitaire_ht` en cinq nombres : minimum, premier quartile, médiane, troisième quartile, maximum. Attendu : 575 · `3 650` · 6 400 · 11 050 · 45 000 (écart-type 8 372). Cette étape ne juge rien, elle dimensionne — mais écrivez-la : un seuil posé sans distribution sera relevé par quelqu'un d'autre, et vous ne pourrez pas expliquer le vôtre.

Étape 2. Appliquez IQR à 1,5 ×, puis z-score à 3 σ et à 6 σ. Attendu : 24 180 · 3 894 · 0 lignes signalées. Le rapport est de six à un entre les deux premiers, et de l'infini à zéro entre le deuxième et le troisième. Une règle dont le résultat varie de six fois selon son curseur n'est pas une règle : c'est un curseur. Notez aussi que la borne basse de l'IQR tombe à `−7 450` : un contrôle dont une borne est inatteignable par construction ne mesure que la moitié de ce qu'il prétend mesurer.

Étape 3. Construisez le contrôle croisé : ratio au catalogue, violations de [0,95 ; 1,35]. Attendu : 0, pour un ratio de 0,958 à 1,306. Un 0 ne vaut que si la bande est serrée et que vous citez les extrêmes : avec une bande [0,5 ; 5], le même 0 ne dirait rien du tout.

Étape 4. Traitez les 973 remises. Deux voies : `÷100` (puis contrôle du domaine : 258 lignes hors [0 ; 0,25]) et reconstitution depuis les montants (0,000 à 0,110, 973 sur 973 dans le domaine). Écrivez la décision au format du § 5.5, en nommant ce que la réparation choisie ne restaure pas : le `montant_ht` n'avait pas besoin de l'être. Si vous avez écrit « correction par ÷100 », l'étape 3 de la démonstration vous a échappé — retournez-y.

Étape 5. Décidez de la sorts des règles : bande de prix → niveau 3 · domaine remise → niveau 2 avec reconstitution · IQR → retiré, phrase de retrait écrite. **Note /10** : 3 points pour la distribution et les trois compteurs, 2 points pour la bande justifiée par les extrêmes, 3 points pour le test de réparation des 973 lignes, 2 points pour l'affectation des niveaux et la phrase de retrait. Un exercice sans décision écrite plafonne à 5.

11. Exercices autonomes

Exercice 4.1 (★) — Trois natures (10 min). Classez en impossible / invraisemblable / licite-surprenant, en citant le document qui permet de trancher : 2 848 quantités négatives · 800 quantités supérieures à 50 · 973 remises supérieures à 1 · 7 674 poids supérieurs à `5 000` kg · 51 dates postérieures au 31/08/2026 · 3 421 montants non numériques.

Exercice 4.2 (★★) — IQR contre bande (15 min). Sur `prix_unitaire_ht`, calculez les bornes IQR, le nombre de lignes hors bornes, puis les violations de la bande [0,95 ; 1,35]. Laquelle des deux règles garde le registre, et quelle phrase justifie le retrait de l'autre ?

Exercice 4.3 (★★) — La règle qui hurle (15 min). Stocks : 1 800 lignes sous le seuil d'alerte, 1 800 quantités physiques négatives, écart théorique-physique maximal 532, moyen 2,77, sur 60 000 lignes. Proposez deux règles au lieu d'une, avec pour chacune le taux de déclenchement estimé et le destinataire. Que devient le taux si vous bornez l'écart à 100 unités, et pourquoi ce n'est pas la question centrale ?

Exercice 4.4 (★★) — Le test de réparation (12 min). Prenez la remise maximale lue (30) et passez-la au domaine après division par 100. Combien de lignes restent hors du plafond de 0,25, et qu'est-ce que ça dit de l'hypothèse « pourcentage saisi en points » ? Recalculez la remise depuis les montants : quelle étendue obtient-on ?

Exercice 4.5 (★★★) — Le registre complet (25 min). Rédigez le registre de contrôles du fichier de ventes : douze règles maximum, avec règle, niveau, destinataire, cadence, taux attendu et mesure du jour. Vous devez y faire figurer au moins un contrôle à 0, au moins un contrôle croisé sur le référentiel produits, au moins un **test de réparation**, et une règle volontairement fausse documentée comme telle avec le nombre qu'elle produirait.

12. Correction détaillée

Exercice 4.1. Les 2 848 quantités négatives sont **licites** : la règle du socle veut qu'un retour ait quantités et montants négatifs — le document qui tranche est la notice. Les 800 quantités > 50 sont **licites-surprenantes** : seul un seuil d'achat dit si c'est anormal. Les 973 remises > 1 sont

impossibles, par définition de la variable. Les 7 674 poids > `5 000` kg sont **invraisemblables** : le doute porte sur la nature de la colonne (poids total ou unitaire), pas sur la ligne. Les 51 dates après le 31/08/2026 sont impossibles pour la période annoncée et valides en elles-mêmes : la borne est un document, pas une donnée. Les 3 421 montants non numériques ne sont pas une aberration du tout : le défaut est dans l'écriture, pas dans la valeur — c'est le C05. Morale : quatre des six décisions ne se prennent pas dans le fichier mais à côté de lui.

Exercice 4.2. IQR : `7 400` d'écart inter-quartiles, bornes `[-7 450 ; 22 150]`, **24 180 lignes** hors bornes, **0 erreur**. Bande catalogue : **0 violation** pour un ratio de 0,958 à 1,306. On garde la bande et on écrit le retrait : « règle écartée le [date] : elle signale 9,9 % des lignes d'une colonne à dispersion structurelle et ne désigne aucune erreur ; le contrôle équivalent est la bande relative au catalogue, à 0 violation sur une bande de ±3,5 % autour de la moyenne observée ». Le piège de la borne basse mérite sa phrase à lui : `−7 450` est inatteignable, donc la moitié de la règle ne peut rien dire — et une règle à moitié inactive donne un faux sentiment de sévérité.

Exercice 4.3. Règle 1 · intégrité : `qte_physique < 0` → 1 800 lignes sur 60 000, soit 3,0 %, niveau 2, responsable de dépôt, quotidienne. Règle 2 · risque : `qte_theorique − qte_physique > k` → 532 comme maximum observé et 2,77 comme moyenne disent d'abord que la queue est courte ; avec `k = 100`, le taux tombe très en dessous des 3 % et il faudra le mesurer, pas le promettre. Ce qui change vraiment : les deux signalements ne se recouvrent plus, donc chaque alerte porte une action distincte. Et la question centrale n'est pas le taux mais l'action : une alerte à 0,1 % que personne ne traite vaut moins qu'une alerte à 3 % qui déclenche un comptage.

Exercice 4.4. 30 ÷ 100 = 0,30 → **258 lignes** au-dessus du plafond de 0,25. L'hypothèse « pourcentage saisi en points » est donc insuffisante : elle n'explique pas le lot. Recalcul depuis les montants : étendue 0,000 à 0,110, les 973 dans le domaine. Et 630 de ces lignes ont un `montant_ht` égal au prix plein, ce qui serait impossible avec une remise de 12 à 30 % appliquée. Conclusion à écrire : la colonne a été **remplacée** par une valeur qui ne la décrit plus, elle n'a pas été échelonnée — d'où le choix de la reconstitution plutôt que de la conversion.

Exercice 4.5. Attendu en substance : douze lignes, dont (a) un contrôle à 0 daté — `id_produit` orphelin = 0 sur 243 360, `tva_non_conforme` = 0, `dates_non_ISO` = 0, avec la mention « le zéro est un résultat » ; (b) le croisé poids (8 380, niveau 3, avec la colonne « table suspecte ») ; (c) un **test de réparation** écrit comme tel — le ÷100 des remises, 258 hors domaine, règle rejetée ; (d) la règle fausse nommée (IQR prix, 24 180 alertes, motif du retrait). Barème du rendu : exhaustivité et non-duplication (4) · taux et dénominateurs (2) · décisions et destinataires (2) · règle fausse documentée (2). Le dernier point est le plus rare en production, donc le plus noté.

13. Mini-projet M04.P4 — « Le registre de contrôles du fichier de ventes » (1 h 30)

Livrable : `controles_ventes.md`, précédé d'une page de garde (fichier, date, taille 28 516 920 octets, 243 360 lignes). Contenu imposé :

- douze règles maximum, dont au moins **trois contrôles croisés** avec un référentiel et au moins **deux** dont le résultat mesuré est 0 ;
- pour chaque règle : l'affirmation vérifiable, le niveau sur l'échelle, le destinataire nommé, la cadence, le taux attendu, la mesure du jour avec son dénominateur, et la date de calibration du seuil s'il y en a un ;
- une annexe « **réparations rejetées** » : pour chaque correction envisagée puis écartée (le `÷100` des remises est le cas fourni), le nombre qu'elle laisse subsister et la raison ;
- une annexe « **règles retirées** » avec le volume d'alertes de chacune ;
- une dernière section : ce que ce registre ne couvre pas — trois défauts du fichier que vos douze règles laissent passer, chiffrés (les 61 lignes à double défaut, par exemple).

Barème (10 points). Affirmations vérifiables et non dupliquées (2) · croisés et zéros explicites (2) · niveaux, destinataires, cadences, dates de calibration (2) · annexes des réparations et retraits (2) · section de non-couverture (2). Seuil de validation : 7. Un registre qui prétend tout couvrir sera cru, puis contredit : la dernière section vaut autant que les quatre premières.

14. Résumé du chapitre

```
NATURE DE L'ANOMALIE     OUTIL QUI LA VOIT          ICI                ACTION
impossible               règle de domaine           0 · 0 · 0 · 0      bloquer à l'entrée
contredit un doc         contrôle croisé            973 · 8 380        instruire, nommer la table
seuil de pratique        seuil métier documenté     800                surveiller au destinataire
dispersion réelle        IQR · z-score              24 180 · 3 894     RETIRER du registre
écriture abîmée          type de cellule            3 421              normaliser (C05)
```

Le chapitre tient dans une asymétrie : les contrôles **sans paramètre** (domaine, égalité arithmétique, référentiel joint) trouvent les vrais défauts ; les contrôles **à paramètre** (quartiles, sigmas, seuils d'alerte) trouvent le profil du fichier. Ces derniers restent utiles — dimensionner, calibrer, surveiller une dérive — à condition de ne jamais être la règle qui décide. Et gardez le cas des 973 remises en mémoire : un défaut qui ne change aucun total est le seul à pouvoir survivre à tous vos contrôles de totaux.

15. À retenir

> **À retenir. ★ Aberrant ≠ erroné.** 24 180 lignes hors IQR, 3 894 au-delà de 3 σ, 0 à 6 σ — et pas une erreur : la dispersion est le métier (575 à 45 000 FCFA, du sac de ciment à la palette d'ossature). **★ La règle qui voit juste croise deux sources.** Bande relative au catalogue : 0 violation, ratio 0,958 à 1,306. Égalité au catalogue : 232 981 « anomalies » — c'est la règle qui est fausse. **★ Testez la réparation avant de l'appliquer.** `÷100` sur les remises laisse 258 lignes hors du domaine ; la reconstitution depuis les montants laisse 973 lignes dans le domaine, entre 0,000 et 0,110. **★ Un défaut qui ne change aucun total est le plus dangereux.** Les 973 lignes laissent le chiffre d'affaires intact (TVA conforme sur 243 360 lignes) et rendent fausse toute analyse de remise. **★ Une règle sans décision n'est pas un contrôle.** Bloquer · corriger · surveiller · ignorer tracé : quatre niveaux, un destinataire par niveau, une date de calibration par seuil, un taux attendu écrit à l'avance.

16. Évaluation formative (auto-correction, 8 min)

Q1. Une valeur à 6 écarts-types est : (a) une erreur ; (b) une valeur rare ; (c) une valeur à supprimer. → **(b)**, et ici (a) ne s'applique à aucune des 243 360 lignes de prix : la borne à 6 σ ne mord sur rien, celle à 1,5 × IQR en signale 24 180.

Q2. Pourquoi la règle de domaine sur la remise l'emporte-t-elle sur le z-score ? → elle n'a aucun paramètre à justifier, elle est reproductible d'un lot à l'autre, et elle ouvre une correction vérifiable. Le z-score trouve des lignes, pas des erreurs, et son résultat dépend du reste du fichier.

Q3. Le contrôle croisé poids signale 8 380 lignes. Que ne sait-il pas dire ? → quelle table est suspecte : `poids_kg` de la ligne, `poids_unite_kg` du catalogue, ou la quantité. C'est pourquoi il est de niveau 3, pas de niveau 2.

Q4. La règle `qte_physique < seuil_alerte` déclenche sur les mêmes 1 800 lignes que `qte_physique < 0`. Conclusion ? → le seuil ne discrimine pas, la condition réelle est le signe. Une règle qui reproduit le résultat d'une règle plus simple doit être scindée ou retirée.

Q5. On vous propose « diviser les remises par 100 ». Quelle est la première question à poser ? → qu'est-ce que ça donne au contrôle de domaine ? Ici : 258 lignes hors [0 ; 0,25], donc la réponse est « insuffisant » — et la reconstitution depuis les montants est le seul remède qui passe le test.

Question ouverte (la plus importante). Reprenez une règle statistique du § 5.8 et rendez-la utile sans en faire un contrôle : dites ce qu'elle surveille, à quelle cadence, avec quel taux attendu, et à qui l'alerte s'adresse. Corrigé attendu : un **indicateur de dérive** — le nombre de lignes hors IQR, hebdomadaire, alerte si l'écart à la référence dépasse 20 % (24 180 cette semaine ; le double, et quelque chose a changé dans la nature du fichier, pas dans sa propreté). Une règle statistique bien employée ne dit pas quelles lignes sont fausses : elle dit quand le fichier a changé.

Module M04.C05 — Normaliser et rendre comparable : changer l'écriture, jamais la valeur

Outil de ce chapitre : le tableur (formules de texte et Power Query), sur `ventes_brutes.csv`, `clients.csv`, `produits.csv`, `objectifs_de_ca.csv`, `remises_manuelles.xlsx` et `tarif_fournisseur_peinture.csv`. Durée indicative : 5 h. Niveau : N2.

> **L'idée du chapitre.** Nettoyer retire des lignes ; normaliser ne retire rien. Normaliser change **l'écriture** d'une valeur en gardant **la valeur**, pour que deux lignes qui parlent du même objet puissent se reconnaître, et pour qu'une colonne cesse d'être du texte quand elle devrait être un nombre. Le socle de données donne de cette opération quatre leçons mesurées, et chacune se vérifie en une ligne de contrôle. (1) Les 3 421 montants écrits en texte se reconvertissent par deux voies indépendantes — découper le texte, ou sommer `montant_ht + montant_tva` — et les deux rendent 15 639 751 286 FCFA, écart de 0 franc. (2) La mise en forme est une information : les 412 lignes que la référence retire du fichier clients portent **toutes** deux anomalies d'écriture, et **aucune** des 23 500 autres ne les porte. (3) Normaliser ne rend pas juste : 0 date de vente sur 243 360 n'est hors norme, et pourtant 51 lignes portent une date postérieure à la date du relevé. (4) L'écriture fabrique les comparaisons : la table d'objectifs attend 220 cases, en contient 218, et ce n'est qu'après avoir typé la clé que l'on voit lesquelles manquent.
>
> **Base de travail.** `01_socle_donnees/data/brut/ventes_brutes.csv` (243 360 lignes × 20 colonnes), `01_socle_donnees/data/brut/clients.csv` (23 912), `01_socle_donnees/data/brut/produits.csv` (154), `01_socle_donnees/data/brut/objectifs_de_ca.csv` (218 lignes, séparateur `;`), `01_socle_donnees/data/brut/remises_manuelles.xlsx` (18 200), `01_socle_donnees/data/brut/tarif_fournisseur_peinture.csv` (38 lignes utiles, encodage Windows-1252), `01_socle_donnees/data/reference/ventes_propres.csv`, `clients_propres.csv` et `dictionnaire_produits.csv`. Tous les nombres de ce chapitre viennent de ces fichiers, relevés par `01_socle_donnees/scripts/chiffres_manuel.py`.

---

1. Objectifs du chapitre

À la fin de ce chapitre, vous saurez :

- **distinguer** trois gestes qu'on confond : normaliser (changer l'écriture), nettoyer (retirer des lignes), changer de convention (redéfinir ce que la colonne veut dire) — et nommer lequel vous êtes en train de faire ;
- **convertir** une colonne monétaire stockée en texte par un découpage dont on prouve qu'il ne perd rien, et écrire le contrôle de convergence qui ferme le dossier à 0 franc près ;
- **mesurer le gain** d'une normalisation avant de l'écrire (`12 640` noms distincts deviennent `12 342`, la ville ne gagne rien : 6 modalités avant, 6 après) et supprimer la formule qui n'apporte rien ;
- **importer sans perte** un fichier à séparateur `;`, à décimales par virgule, en Windows-1252 et à en-tête décalé, en chiffrant ce que le mauvais réglage coûte de caractères ;
- **ranger les opérations** dans l'ordre qui fait que chaque étape est vérifiable, et refuser les trois normales qui abîment : l'arrondi des montants, la casse sur un code, la réécriture sur place.

2. Pourquoi cette notion est importante

Un tableau croisé qui affiche douze lignes « par mois » au lieu de dix, une jointure qui rend 0 correspondance alors
que les deux fichiers parlent des mêmes clients, un total de chiffre d'affaires en texte que la fonction SOMME ignore

> tous ces accidents viennent du même endroit, et aucun n'est un problème de saisie. Voici ce que le socle montre quand on pose la question « la valeur est-elle juste ? » séparément de « son écriture est-elle comparable ? ».

Ce que vous croyez avoir	Ce que le fichier contient	Ce que ça coûte
Une colonne de montants numériques	239 939 nombres et 3 421 textes dans la même colonne N	le SOMME est faux de 3 421 lignes, le filtre « plus grand que » les ignore
Un fichier clients sans doublon	412 lignes au nom encadré d'espaces et au téléphone sans espaces	412 clients comptés deux fois, 11 158 fusionnés à tort si l'on corrige à l'envers
Une table d'objectifs complète	218 lignes là où la grille en prévoit 220	un magasin sans objectif pendant deux mois, classé « sous performance » par un tool qui lit 0
Une jointure entre deux fichiers	`id_client` nombre dans le classeur, texte dans le CSV	0 ligne jointe, sans message d'erreur, dans Power Query

Le point commun : **aucun de ces défauts ne produit d'erreur à l'écran**. Le tableur accepte le mélange de types, la jointure ratée rend un tableau vide proprement, le texte non converti se somme à zéro silencieusement. C'est pourquoi la normalisation est un métier et non un réglage : elle consiste à rendre visible, avant de rendre propre.

> **Définition.** On appelle **normalisation** toute transformation bijective à l'identifiant près appliquée à une valeur, qui change sa représentation et non sa signification : `+226 77 76 25 98 96` et `+2267776259896` désignent le même abonné ; `16 638 FCFA` et `16638` désignent la même somme. Le test de la normalisation est donc double : deux écritures qui désignent le même objet doivent devenir identiques (**rapprocher**), et deux écritures qui désignent des objets différents doivent rester distinctes (**ne pas confondre**). Une transformation qui ne fait que le premier est une perte d'information ; une transformation qui rate le second est une faute.

3. Explication simple

Regardez deux lignes du fichier clients, telles qu'elles sont écrites dans `/home/user/formation-data-bi/01_socle_donnees/data/brut/clients.csv` (ligne 2 et ligne 23 502, à vérifier avec `grep -n "SARL K" clients.csv`) :

```
2,     SARL k, Entreprise, Koudougou, ..., +226 78 62 21 81 98, , 2022-03-15, ...
23501,   SARL K , Entreprise, Koudougou, ..., +2267862218198,    , 2022-03-15, ...
```

Même ville, même date de création, mêmes conditions de paiement, même plafond, même courriel absent. Le nom change de casse et gagne deux espaces ; le téléphone perd ses espaces et ne change pas de numéro. Un être humain lit un seul client. Le tableur lit deux chaînes différentes, le tableau croisé affiche deux lignes, la jointure compte deux clients. Ce n'est pas que le fichier soit sale : c'est que **deux écritures n'ont rien de comparable tant qu'elles n'ont pas été ramenées à une norme d'écriture commune**.

Maintenant l'opération inverse, celle qui fait le plus de dégâts. On décide d'« harmoniser » : `SUPPRESPACE()` pour rogner les espaces, `MINUSCULE()` pour unifier la casse. Regardons ce que ça donne sur la colonne qui compte vraiment, le montant :

```
16 638 FCFA  →  SUPPRESPACE →  16 638 FCFA   (rien ne change : les espaces sont à l'intérieur, pas
autour)
16 638 FCFA  →  CNUM(SUPPRESPACE(...)) →  #VALUE!   (le texte n'est pas un nombre)
16 638 FCFA  →  CNUM(SUBSTITUE(...;" FCFA";"")) →  #VALUE!   (l'espace de milliers reste collé)
16 638 FCFA  →  CNUM(SUBSTITUE(SUBSTITUE(...;" FCFA";"");" ";"")) →  16638   (enfin)
```

Quatre formules, une seule marche. Cette progression est tout le chapitre : la normalisation n'est pas un bouton, c'est une suite de décisions sur **ce que signifie chaque caractère** de la cellule — ici, trois espaces dont deux sont des séparateurs de milliers et un est un séparateur de devise.

4. Vocabulaire essentiel

Terme	Ce que ça désigne, dans ce chapitre
Écriture (représentation)	La suite de caractères de la cellule : `16 638 FCFA`
Valeur	Ce que l'écriture décrit : seize mille six cent trente-huit francs
Typage	Attribuer à une colonne le bon type : nombre, date, texte, booléen
Élagage	Retirer ce qui entoure la valeur : `SUPPRESPACE()` pour les espaces, `CLEAN()` / `Text.Clean` pour les caractères de contrôle
Harmonisation de casse	Ramener majuscules et minuscules à une forme unique — utile sur un nom, interdite sur un code
Séparateur de milliers	L'espace (ou la virgule, ou le point) qui découpe les groupes de trois chiffres dans un nombre écrit en texte
Encodage	La table qui traduit des octets en caractères : UTF-8, Windows-1252, ISO-8859-1
BOM	Bytes-mark : trois octets `EF BB BF` en tête d'un fichier UTF-8, qui deviennent `ï»¿` si on se trompe de table
Conversion d'unité	Multiplier par un facteur défini par la physique : 1 m³ = 1 000 L — ce n'est **pas** une normalisation de texte
Table de correspondance	Un fichier qui traduit une écriture en une autre, avec une décision documentée derrière chaque ligne
Clé de jointure	La ou les colonnes sur lesquelles deux tables se reconnaissent ; elle doit être normalisée **avant** la jointure
Protocole de normalisation	La liste ordonnée des transformations appliquées, avec pour chacune l'effet mesuré sur le nombre de lignes et de valeurs distinctes

5. Cours approfondi

5.1 Trois objets, trois traitements — et ne pas les mélanger

Une colonne peut être fautive à trois niveaux, qui ne se réparent pas du tout de la même façon.

Niveau	Question à poser	Opération licite	Dans le socle	Ce qui serait une faute
Le type	La cellule est-elle du nombre ou du texte ?	Convertir, en gardant la valeur	3 421 montants en texte	Imputer, arrondir, recalculer le montant
L'écriture	Deux formes décrivent-elles le même objet ?	Élaguer, unifier la casse, retirer les séparateurs	412 noms surchargés d'espaces	Retirer une ligne, fusionner deux homonymes
Le sens	La valeur dit-elle ce qu'on croit ?	Reconstituer ou documenter, jamais « corriger »	973 taux de remise à 12, 18, 25, 30	Diviser par 100 sans test de réparation

Le troisième niveau est celui du chapitre précédent (C04) et il mérite d'être rappelé ici, parce que la tentation est grande de le traiter comme un problème de format. Diviser `taux_remise` par 100, c'est **changer la convention** de la colonne : on décide que le nombre 30 signifiait 30 %. Si c'est vrai, ça se prouve (ça ne se vérifie pas : le test de réparation du C04 montre que 258 lignes sortent alors du domaine contractuel). Une normalisation, elle, ne prétend rien sur le sens. Elle ne fait que rendre lisible une valeur dont tout le monde est d'accord.

> **À retenir.** La question qui précède toute formule : est-ce que je change l'écriture, la ligne, ou le sens ? Écrire la réponse en tête de chaque colonne ajoutée (`_ecriture`, `_ligne`, `_sens`). Vous verrez alors immédiatement si une même colonne cumule deux natures d'opération — et il faudra la scinder.

5.2 Les montants texte : découper, puis prouver par une seconde voie

Colonne `montant_ttc` (colonne N de `ventes_brutes.csv`) : 239 939 cellules numériques, 3 421 non vides et non numériques, 0 vide. Anatomie de ces 3 421, mesurée caractère par caractère :

Ce qu'on cherche	Combien	Ce que ça impose
Suffixe « FCFA » détaché par une espace	3 421 sur 3 421	Un `SUBSTITUE` suffit, aucun `STXT`
Au moins deux espaces (séparateur de milliers)	3 410 sur 3 421	Les 11 autres sont des montants à trois chiffres : `680 FCFA`
Virgule décimale	0	Pas de problème de locale à régler ici
Espace insécable (U+00A0)	0	Le `SUBSTITUE(...;CAR(160);"")` de réflexe ne sert à rien — et le vérifier évite un après-midi de recherche
Formes distinctes après avoir masqué les chiffres	8	Le suffixe est toujours détaché par une espace, le signe moins précède le nombre, et le seul écart notable est le nombre de groupes de trois chiffres
Montants négatifs	47	Ce sont les 47 retours sans montant, pas des erreurs
Réparables par le découpage `FCFA` + espaces	3 421 sur 3 421	La formule tient, sans exception à traiter à la main

La ligne à `3 782 047 FCFA` (le plus gros texte) et celle à `-193 284 FCFA` (le plus petit) passent toutes les deux. Et surtout : **les 47 textes négatifs sont exactement les 47 lignes de retour dont le montant manquait**. Une fois converties, le contrôle « toute ligne de retour porte un montant négatif » passe sur 2 848 lignes sur 2 848, alors qu'il échouait sur 47 avant. La normalisation ne s'est pas contentée de changer un type : elle a fait passer un deuxième contrôle de 47 lignes ratées à zéro.

Reste à prouver que le découpage n'a rien inventé. Deux voies indépendantes :

```
Voie A (découpage du texte) :  somme des 3 421 textes convertis + somme des 239 939 nombres = 15
639 751 286
Voie B (reconstitution)      :  somme de (montant_ht + montant_tva) sur les mêmes lignes        =
15 639 751 286
Contrôle                     :  écart = 0
```

Zéro franc d'écart, à l'unité près. C'est ce qui autorise à écrire dans le registre « colonne reconvertie, aucune valeur perdue » — et non « colonne reconvertie, semble correcte ».

> **Définition.** Un **contrôle de convergence** fait arriver le même résultat par deux chemins qui n'ont aucune raison de se ressembler : l'un lit ce que l'opérateur a tapé, l'autre recalcule depuis d'autres colonnes. Convergence à 0 ⇒ la transformation est une écriture différente de la même information. Écart non nul ⇒ au moins un des deux chemins est faux, et il faut chercher lequel **avant** de décider lequel est juste.
>
> **Attention.** La convergence à 0 n'a été possible que parce qu'aucune autre colonne n'a été « réparée » au passage. Si vous aviez divisé `taux_remise` par 100 (voir C04 §5.3) avant de faire ce calcul, `montant_ht` ne serait plus indépendant du texte : les deux voies se ressembleraient par construction et le contrôle ne prouverait plus rien. Un contrôle croisé exige que les deux voies restent **indépendantes** — donc il se fait avant les corrections de sens, pas après.

5.3 Mesurer le gain avant d'écrire la formule

Le réflexe « appliquer `SUPPRESPACE` et `MINUSCULE` à toutes les colonnes de texte » coûte trois colonnes, ralentit le fichier et ne répare souvent rien. Le gain se mesure, en une ligne, colonne par colonne : nombre de valeurs distinctes avant, après ; nombre de lignes modifiées.

Colonne	Modalités avant	Modalités après normalisation	Verdict
`clients.nom` (brut → normalisé)	12 640	12 342	298 écritures différentes d'un même nom : à normaliser, et c'est ce qui rend la clé du C03 possible
`clients.telephone`	23 499 textes distincts	23 499 numéros après retrait des non-chiffres	un numéro porté par deux écritures : à normaliser pour la clé, pas pour l'affichage
`clients.ville`	6	6	aucun gain : `MAJUSCULE()` serait du décor
`produits.unite`	4	4	aucun gain textuel — mais 4 **unités physiques**, qui est un autre problème (voir 5.5)
`clients.email`	7 398 remplis	7 398	rien à reformater ; les 16 514 cas sans `@` sont des champs vides, pas des adresses cassées

Deux enseignements. D'abord, une anomalie d'écriture peut être un **indice** plus précieux qu'une saleté : les 412 lignes retirées de `clients.csv` portent toutes un nom encadré d'espaces (412 sur 412) et un téléphone sans espaces (412 sur 412), alors que **0** des 23 500 lignes conservées porte la première anomalie. Autrement dit, la mise en forme signale ici exactement le bloc à retirer. C'est le genre de coïncidence qu'on ne découvre qu'en comptant, et qui change la nature d'un contrôle : on ne supprime plus « les doublons trouvés par une règle de similarité », on supprime « les lignes marquées par une anomalie de format », ce qui est déterministe et rejouable.

> **Dans les faits.** Le lot des 412 doublons clients a été repéré à l'aveugle par une règle de similarité, puis reproduit par une règle de format (412 sur 412, et 0 faux positif sur les 23 500 lignes conservées). Les deux méthodes tombent d'accord une fois ; elles ne le seront plus la suivante, parce que la règle de format décrit un import passé et non une propriété durable du client. Un protocole de normalisation doit donc distinguer explicitement la **règle de décision** (le contenu de la clé) de l'**indice de rejeu** (la mise en forme du lot), et écrire à côté de l'indice qu'il ne vaut plus rien dès que la source change de poste de saisie.

Ensuite, un champ **vide** n'est pas un champ mal formaté. `m04_clients_email_sans_a_robase` vaut 16 514 : ce ne sont pas 16 514 adresses illisibles, ce sont 16 514 absences (le C02 a nommé le mécanisme). Une normalisation appliquée à une absence produit un objet faux : `NORMALISE("")` peut rendre `@`, `0`, une ligne blanche selon la formule — un texte non vide là où il y avait une ab-

sence, donc une valeur comptable là où il n'y en avait pas. Le contrôle d'après-normalisation est simple : le nombre de cellules vides ne doit pas bouger.

> **Conseil professionnel.** Écrivez la mesure de gain **avant** la formule, dans le même onglet : trois cellules, `=NBCAR.SI(...)` ou un compteur de modalités, avant, après, écart. Quand on vous demandera dans six mois pourquoi la colonne `nom_normalise` existe, vous répondrez avec un chiffre, pas avec une intention.

5.4 Les dates : la norme est respectée, la réalité non

Sur `date_vente` des 243 360 lignes : 0 valeur qui ne soit pas exactement `AAAA-MM-JJ`. Sur `date_creation` des 23 912 lignes de `clients.csv` : 0 hors norme. Il n'y a donc **aucune conversion de date à écrire** dans ce socle — et c'est une information, pas un hasard : le format ISO est précisément conçu pour que l'ordre alphabétique soit l'ordre chronologique, et pour que la locale du poste n'ait aucune prise sur lui.

Ce qui ne va pas, ce sont les valeurs : 51 lignes portent une date postérieure au 31 août 2026, la plus tardive étant le 2026-12-06, alors que le relevé s'arrête fin août. Ces 51 dates sont **parfaitement normales** et **physiquement impossibles** : elles passent tous les contrôles de format et aucun contrôle de domaine.

La conséquence pratique est la règle suivante : jamais de contrôle de validité sur une date réduit à « est-ce que ça se lit comme une date ». Celui-là ne dit que ce que le tableur sait déjà. Trois contrôles, dans l'ordre, et le premier n'est pas celui qu'on croit :

Contrôle	Forme à écrire	Résultat sur le socle
Homogénéité de la norme	toutes les valeurs matchent `^[0-9]{4}-[0-9]{2}-[0-9]{2}$`	0 exception
Domaine	`date` entre la première date d'exploitation et la date du relevé	51 lignes hors domaine
Cohérence entre colonnes	`date_vente` de la ligne = date du ticket (toutes lignes d'un même `id_ticket`)	contrôle à écrire, c'est celui qui attrape les dates recopiées de travers

> **À retenir.** Normaliser une date, c'est aligner son écriture. Valider une date, c'est la confronter au monde. Les 51 lignes futures ne se réparent pas par une formule : elles se décident (les exclure du périmètre d'analyse et le dire, ou les faire corriger par la source). Écrire « dates normalisées » pour ce lot serait un mensonge technique — l'écriture n'était pas le problème.

5.5 Unités et correspondances : ce qui demande un facteur, pas une formule

> **Définition.** Une **conversion d'unité** multiplie une valeur par un facteur fixé par une définition extérieure au fichier (1 m³ = 1 000 L), et change le nom de la colonne en conséquence : `quantite` devient `quantite_litres`. Une **harmonisation d'écriture** ne change ni le facteur ni le nom : `LITRE`, `litre` et `Litre` deviennent `litre`. Confondre les deux produit un nombre juste dans une colonne mal nommée, ce qui est le genre d'erreur qui se propage à tous les tableaux suivants sans jamais plus se signaler.

`produits.csv` porte une colonne `unite` avec 4 modalités : 122 produits en `unité`, 14 en `litre`, 12 en `sac`, 6 en `m3`. Aucune variante de casse, aucun espace parasite : en tant que texte, cette colonne est propre. En tant que physique, elle est inerte — et c'est là que le piège se referme. Une somme de « volumes » sur ce champ additionne des mètres cubes et des litres comme s'ils

étaient la même chose. Le correctif n'est pas une fonction texte : c'est un **facteur de conversion** porté par une table, 1 m³ = 1 000 L, et une colonne `quantite_en_litres` qui n'existe que pour les lignes où la question a un sens.

Le fichier `tarif_fournisseur_peinture.csv` montre la même collision, aggravée. Ses 38 lignes utiles ont toutes `conditionnement = 10 L` (1 modalité seulement) et `unite_mesure = litre`, alors que les libellés décrivent des tubes en millimètres vendus au mètre linéaire (`Tube PVC évacuation 100 mm (ml)`). Trois colonnes, trois vocabulaires d'unité, et aucune ne permet de raisonner sur le stock. Ce n'est pas une saleté à élaguer : c'est un **dictionnaire à écrire avec le magasin**, et il se solde par une colonne de plus, pas par une formule de moins.

Côté catégories, la leçon est inverse : `produits.csv` en compte 15 modalités, le dictionnaire de référence 7, et 44 lignes portent un intitulé hétéroclite. Ici, la normalisation **par table de correspondance** est la bonne opération, et le C03 a montré pourquoi : une correspondance est une décision documentable, une formule ne l'est pas. La règle de tri est donc :

- si la transformation s'énonce sans dictionnaire (`SUPPRESPACE`, minuscule, retrait d'un suffixe), c'est de la normalisation : elle s'écrit en formule et elle est gratuite à rejouer ;
- si elle exige de dire « cette écriture-là veut dire cet objet-ci » (`enduits` + `peinture` → un seul poste, `m3` → des litres), c'est une correspondance ou une conversion : elle s'écrit dans une **table**, avec une source et une date, sinon vous n'aurez pas à montrer votre travail.

5.6 Importer sans rien casser : séparateur, décimales, encodage, en-tête

Un fichier est quatre décisions de lecture, et le socle en contient un exemplaire de chaque mauvaise :

1. **Le séparateur.** `ventes_brutes.csv` est à virgule ; `objectifs_de_ca.csv` et `tarif_fournisseur_peinture.csv` sont à point-virgule. Ouvrir le second avec le réglage du premier produit une seule colonne de texte, et le tableur ne dit rien : il affiche une très longue colonne A.
2. **La décimale.** `tarif_fournisseur_peinture.csv` écrit ses prix `2900,57` et sa TVA `0,18` ; `objectifs_de_ca.csv` écrit sa marge `0.195`. En français, l'un se lit correctement et l'autre devient du texte (ou l'inverse selon la version d'Excel). Le contrôle est immédiat : la colonne importée doit être **de type nombre** et son total doit égaler `SOMME` calculé sur le texte découpé.
3. **L'encodage.** `tarif_fournisseur_peinture.csv` pèse 2 482 octets dont 95 ne sont pas ASCII, et ne se décode pas en UTF-8 du tout : l'octet 73 est illégal. C'est du Windows-1252. La nuance qui fait mal : Windows-1252 et ISO-8859-1 (le « Latin 1 » du sélecteur Excel) sont identiques sur 223 des 256 octets, et **differents sur les 39 octets** que ce fichier contient justement entre `0x80` et `0x9F` — les tirets demi-cadratin `–` qui séparent le libellé de la référence fournisseur. Choisir « ISO Latin 1 » n'affiche pas de caractère bizarre : ça insère des caractères de contrôle invisibles, que la recherche ne retrouve plus et que l'impression avale.
4. **L'en-tête décalé.** Les trois premières lignes du fichier de tarif sont du bruit d'export (`export genere depuis ELODIE v3`, `edition le 17/09/2026`, le nom du fournisseur). L'en-tête réelle est ligne 4. Un import aveugle nomme les colonnes d'après la ligne 1 et perd le vrai nommage.

Quant aux CSV proprement dits : 5 des 10 fichiers du dossier `data/brut/` portent un BOM (trois octets `EF BB BF` avant le premier caractère). Lus en UTF-8 sans retirer le BOM, rien d'anormal ne se produit ; forcés en Windows-1252 — le réglage par défaut de l'assistant d'importation legacy — l'en-tête de `ventes_brutes.csv` ne s'appelle plus `id_vente` (8 caractères mesurés dans le fichier) mais `ï»¿id_vente` (11 caractères mesurés après décodage). Onzième caractère invisible pour l'œil, premier caractère de tous vos noms de champs. Le contrôle coûte deux secondes : regarder la longueur du premier en-tête importé.

> **Définition.** Un **encodage** est une table de correspondance octet → caractère. UTF-8 en est une où les caractères non ASCII occupent plusieurs octets ; Windows-1252 et ISO-8859-1 en sont où chaque octet vaut un caractère, avec 39 cas de divergence. Une erreur d'encodage est **réversible** tant que vous n'avez pas sauvegardé, irréversible après : les octets d'origine ont été remplacés. D'où la règle de non-réécriture : on ne sauvegarde jamais un import dans le fichier source, on écrit un fichier dérivé.

5.7 Les clés : normaliser pour que les jointures disent quelque chose

Trois accidents de ce socle, tous mesurés, viennent de ce qu'une clé n'a pas été typée avant usage.

- **Nombre contre texte.** `remises_manuelles.xlsx` porte `id_client` en entier ; `clients.csv` le porte en texte. Le contenu, lui, est bon : 1 400 clients distincts dans les remises, **0** absent de la fiche clients. Dans Power Query, une fusion sur ces deux colonnes ne lève aucune erreur et rend un tableau vide. La première étape de toute requête est donc un `Table.TransformColumnTypes` sur les colonnes de clé — pas une élégance, une nécessité.
- **La grille attendue n'est pas la grille reçue.** `objectifs_de_ca.csv` couvre 5 magasins et 44 mois : la grille complète aurait 220 lignes, elle en a 218. Les 2 manquantes sont consécutives (février et mars 2024) et n'affectent qu'un magasin, le n° 4 — signature d'un export interrompu ou d'un collé mal fini, pas d'un oubli de saisie. Et le magasin 6 n'apparaît nulle part dans ce fichier : 1 magasin sans objectif, trouvé en comparant deux jeux de clés et non en lisant le fichier.
- **L'absence, une fois encore, se déplace.** Tant que la clé n'est pas normalisée, un trou ne se voit pas : la jointure externe rend une ligne avec `0` ou `null` que le tableau croisé affiche comme une valeur. Après normalisation et jointure explicite, le trou devient une ligne nommable : « magasin 4, 2024-02, aucun objectif ».

C'est le moment de nommer l'ordre des opérations, qui est la colonne vertébrale du chapitre :

Étape	Opération	Contrôle qui la ferme	Effet attendu
1	Typage des colonnes de clé et des montants	type de colonne + total de contrôle	0 changement de valeur
2	Élagage et harmonisation des colonnes-clés et étiquettes	nb de lignes inchangé, nb de modalités réduit	ici : 298 noms fusionnés
3	Traduction par tables de correspondance	0 clé inconnue après traduction	catégories 15 → 7
4	Conversion d'unités (facteur)	ordre de grandeur du total	1 000 × vérifié sur 6 lignes `m3`
5	Puis seulement : dédoublonnage (C03), règles de contrôle (C04), imputation (C02)	écart à la cible	ici : écart 0

5.8 Ce qu'une normalisation ne doit jamais faire

Quatre interdictions, chacune motivée par un chiffre de ce module.

- **Ne pas changer le nombre de lignes.** Le contrôle est brutal : 243 360 lignes avant, 243 360 après. Si le compte bouge, vous avez fait du nettoyage sous un nom poli — et vous ne l'avez pas dit.
- **Ne pas arrondir un montant.** Les 3 421 textes se convertissent à l'unité près ; l'écart de convergence est de 0 parce que rien n'a été arrondi. Un `ARRONDI(...;-2)` sur une colonne monétaire de 243 360 lignes produit une erreur de plusieurs dizaines de millions de francs, et surtout la rend indétectable par un contrôle de total.

- **Ne pas toucher à la casse d'un code.** `id_ticket` a une structure porteuse (`T01-230101-000000`, où `T01` est le magasin) : `MAJUSCULE()` ne change rien ici, mais sur un code sensible à la casse, l'opération fusionne deux objets distincts. La casse s'harmonise sur les libellés destinés à l'humain, jamais sur les identifiants.
- **Ne pas écrire par-dessus.** La réversibilité est un livrable : on garde `montant_ttc_brut` et l'on ajoute `montant_ttc_num`. Un protocole qui détruit la source ne peut plus être audité, et ne peut plus être rejoué quand on découvre, deux semaines plus tard, que le suffixe était parfois `F CFA` avec une espace de trop.

> **Attention.** Le piège le plus coûteux de ce chapitre est la normalisation **par la jointure** : on rapproche deux colonnes de clés en faisant confiance à la table de correspondance, sans compter les clés inconnues. Ici, ce réflexe aurait fusionné 11 158 clients homonymes en plus des 412 doublons réels, parce que la clé « nom seul » est valide comme écriture et fausse comme identité. Toute jointure se ferme par deux compteurs : lignes non jointes à gauche, lignes non jointes à droite — et les deux ne doivent pas être zéro par omission, mais par mesure.
>
> **Boîte à outils du chapitre.** Cinq fonctions et deux réglages, pas davantage : `SUPPRESPACE` (élagage), `SUBSTITUE` (retrait d'un suffixe ou d'un séparateur), `CNUM` et son frère explicite `CNUMPARAGRAPHE` (conversion avec point ou virgule choisis, la seule qui ne dépende pas de votre clavier), `TEXTE(v;"00")` (remise à la même largeur d'une clé courte), `DATE` ou `DATEVAL` (reconstruction d'une date depuis trois colonnes). Côté Power Query, deux réglages : le type de colonne appliqué **en première étape**, et l'option `Encoding` / `Delimiter` écrite dans la requête plutôt que laissée à l'assistant.

Normaliser n'est pas nettoyer — le gain de chaque colonne, mesuré

Gain = valeurs distinctes avant, moins valeurs distinctes après. Une formule à gain nul est un décor.

Colonne	Opération envisagée	Avant	Après	Verdict
montant_ttc (N)	retirer le suffixe puis les espaces	**3 421 textes**	3 421 nombres	type · à convertir
clients.nom	élaguer, minuscule, ponctuation	**12 640**	12 342 · gain 298	écriture · c'est la clé du C03
clients.telephone	garder les chiffres seuls	**23 499**	23 499 · gain 0	écriture · pour la clé, pas l'affichage
clients.ville	élaguer et harmoniser la casse	**6**	6 · gain 0	ne pas toucher : formule à retirer
produits.unite	élaguer et harmoniser la casse	**4**	4 · gain 0	propre en texte, muette en physique
date_vente	aucune conversion à écrire	**0 hors norme**	51 dates impossibles	le format n'est pas le monde

La preuve qui ferme le dossier : deux voies indépendantes, un écart à l'unité

Voie A → somme des 3 421 textes découpés et des 239 939 nombres : 15 639 751 286 FCFA
Voie B → somme de (montant_ht + montant_tva) sur les mêmes lignes : 15 639 751 286 FCFA · écart 0
Contrôle de rattrapage : les retours à montant positif passent de 47 à 0 sur 2 848 lignes.

6. Exemple concret : la cellule `16 638 FCFA`

Quatre formules sur la même cellule N, avec ce que chacune rend et ce qu'on en conclut.

Formule écrite	Résultat	Diagnostic
`=CNUM(SUPPRESPACE(N2))`	erreur	l'espace est à l'intérieur, pas autour : l'élagage ne peut rien

Formule écrite	Résultat	Diagnostic
`=CNUM(SUBSTITUE(N2;" FCFA";""))`	erreur	reste `16 638`, l'espace de milliers bloque la conversion
`=CNUM(SUBSTITUE(SUBSTITUE(N2;" FCFA";"");" ";""))`	16638	correct ; 3 421 lignes sur 3 421 passent
`=CNUMPARAGRAPHE(SUBSTITUE(SUBSTITUE(N2;"  FCFA";"");" ";"");".";",")`	16638	correct et insensible à la locale du poste — à préférer dans un fichier qui circule

Trois précisions qui évitent des heures de recherche :

- les 11 lignes sans séparateur de milliers (`680 FCFA`, `977 FCFA`) ne sont pas un cas à part pour la formule — le double `SUBSTITUE` les traite — mais elles **le deviennent** pour une formule qui découperait « les trois derniers chiffres » à la main ;
- `CAR(160)` (espace insécable) ne sert pas ici : mesuré 0 occurrence. Cette mesure est un gain de temps, pas une perte de temps ;
- le total de contrôle doit être écrit **avant** de recopier la formule vers le bas sur 243 360 lignes, sinon vous ne saurez plus contre quoi le comparer.

7. Démonstration pas à pas : le protocole de normalisation en dix étapes

L'exercice se fait sur une copie de `ventes_brutes.csv` importée par Power Query, et sur `clients.csv`. Notez que chaque étape se termine par un chiffre écrit dans un onglet `CONTROLES`.

```
0. Import.  Données > À partir d'un fichier texte/CSV. ventes_brutes.csv : délimiteur "virgule",
            encodage "UTF-8 (avec BOM)", en-tête présent. objectifs_de_ca.csv : délimiteur "point-
virgule".
            tarif_fournisseur_peinture.csv : encodage "1252 : Europe occidentale (Windows)", 3
lignes à sauter.
1. Compter avant.  Lignes = 243 360 ; clients = 23 912 ; objectifs = 218. Écrire ces trois
nombres.
2. Typer les clés.  Table.TransformColumnTypes sur id_vente, id_ticket, id_client, id_produit,
date_vente,
                    montant_ht, montant_tva, montant_ttc : la conversion échoue sur 3 421 lignes
-> elles
                    deviennent null, et c'est exactement ce qu'on cherchait : le compteur de null
signale le lot.
3. Réparer les null de montant_ttc par découpage.  En requête :
                    Text.Select([montant_ttc]; {"0".."9"; "-"})  puis  Number.FromText
                    (retient les chiffres et le signe, laisse "FCFA" derrière ; 3 421 valeurs, 0
échec).
4. Contrôler la convergence.  Somme colonne N après réparation = 15 639 751 286 ; somme de
(montant_ht + montant_tva)
                    sur les mêmes lignes = 15 639 751 286 ; écart = 0. Écrire l'écart, pas
seulement les deux totaux.
5. Normaliser les colonnes d'étiquette, et mesurer le gain.  clients.nom : SUPPRESPACE + MINUSCULE
+ ponctuation
                    retirée -> 12 640 modalités deviennent 12 342. clients.ville : 6 -> 6.
Conclusion : on garde la
                    colonne nom_normalise, on ne crée pas ville_normalise.
6. Écrire la clé, pas la valeur, quand la jointure l'exige.  cle_client = nom_normalise "|" +
Text.Select(telephone;
                    {"0".."9"}) -> 412 groupes de deux lignes, 0 groupe de trois (le C03 en a
besoin, pas la ville).
7. Remettre à la même largeur les clés courtes.  ici, rien à faire : le test (12 modalités de mois
avant, 12 après)
                    montre que la padding n'est pas le problème de ce fichier. Écrire "0
transformations nécessaires".
8. Traduire par table.  catégories : jointure à dictionnaire_produits.csv (15 -> 7) ; 0 code
produit inconnu après
                    traduction, sinon on s'arrête et on complète la table.
9. Convertir les unités avec facteur, pas avec formule.  produits.unite : table de conversion (m3
-> 1 000 L) ;
                    ne pas additionner `m3` et `litre` dans un même total de volume.
10. Compter après.  Lignes = 243 360 (inchangé), clients = 23 912 (inchangé). Écarts = 0. Le
protocole est clos,
                   et il ne contient aucune suppression : le dédoublonnage viendra après, au C03,
avec son propre compteur.
```

8. Erreurs fréquentes

- **Confondre absence et format.** Appliquer une normalisation aux 16 514 courriels vides crée des chaînes non vides là où il n'y avait rien ; le compte de valeurs « renseignées » augmente, ce qui a l'air d'une amélioration.
- **N'utiliser que `SUPPRESPACE`** sur `16 638 FCFA` : les espaces intérieures survivent, la conversion échoue, et l'opérateur conclut que le fichier est irrécupérable.
- **Supposer l'espace insécable** sans le compter, puis chercher trois heures un caractère qui n'existe pas.
- **Normaliser en écrasant la colonne source** : après sauvegarde, plus aucun contrôle croisé n'est possible, et le BOM mal décodé est devenu le nom définitif de vos champs.
- **Choisir ISO-8859-1 plutôt que Windows-1252** sur le fichier de tarif : 39 caractères deviennent invisibles, et la recherche sur un libellé ne retrouve plus la ligne.
- **Faire la jointure avant le typage** : 0 ligne rapprochée entre le classeur de remises (clés numériques) et le CSV clients (clés texte), alors que 0 client n'existe pas dans la fiche.
- **Arrondir « pour que ça tienne »** : 3 421 conversions à l'unité près valent un écart de 0 ; un arrondi à la centaine fait disparaître le seul contrôle qui prouvait la justesse du découpage.

9. Bonnes pratiques professionnelles

Pratique	Mise en œuvre au tableur	Ce qu'elle garantit
Types fixés en première étape	`Table.TransformColumnTypes` sur toutes les colonnes de clé et monétaires	aucune jointure silencieusement vide
Une colonne ajoutée, jamais écrasée	`x_brut` reste, `x_normalise` s'ajoute	réversibilité et audit
Gain mesuré avant écriture	nb de modalités avant / après, nb de lignes modifiées	pas de formule décorative dans le fichier
Contrôle de convergence à l'unité	deux chemins, un écart, une case	la réparation ne invente rien
Compteurs de non-jointure aux deux bouts	lignes sans correspondance gauche, puis droite	les trous deviennent nommables
Encodage et séparateur écrits dans la requête	options explicites, jamais l'assistant par défaut	le rejou est identique sur un autre poste
Protocole daté dans le classeur	onglet `PROTOCOLE` : ordre, auteur, date, effet mesuré	on répond à une question dans six mois sans rouvrir le code

10. Exercice guidé — reconvertir une colonne monétaire et le prouver (35 min, /10)

Contexte : on vous demande de livrer un chiffre d'affaires mensuel. La colonne `montant_ttc` refuse de se sommer entièrement. Vous devez produire une colonne exploitable et la preuve qu'elle ne déforme rien.

1. **1 pt** — Importer `ventes_brutes.csv` en gardant toutes les colonnes en texte (réglage explicite du séparateur et de l'encodage). Combien de lignes ?
2. **1 pt** — Taper `montant_ttc` en nombre. Combien de `null` apparaît ? Que faut-il en conclure sur la nature du défaut : absence, ou écriture ?
3. **1 pt** — Compter, sur ces lignes seulement, les trois traits suivants : présence de « FCFA », au moins deux espaces, présence d'une virgule. Quel réglage de conversion cela impose-t-il ?
4. **1 pt** — Écrire la formule de conversion qui passe sur les 3 421 lignes. En déduire ce que deviennent les 11 lignes à trois chiffres.
5. **1,5 pt** — Calculer le total de la colonne réparée, puis celui de `montant_ht + montant_tva` sur les mêmes lignes, et écrire l'écart.
6. **1 pt** — Compter les lignes de retour au montant positif avant et après votre conversion. Quel second contrôle vient de passer ?
7. **1,5 pt** — Écrire dans l'onglet `CONTROLES` la fiche de l'opération : source, transformations dans l'ordre, nombre de lignes avant/après, les deux totaux, l'écart, le nom des deux colonnes ajoutées, la date.
8. **2 pt** — Un collègue propose d'ajouter `ARRONDI(...; -2)` pour « lisibilité des tableaux de bord ». Chiffrez ce que ça change au total annuel et concluez. (Indice : vous avez 240 000 lignes propres à comparer, et l'écart accepté sur le chiffre d'affaires était de 2 111 789 FCFA avant réparation.)

11. Exercices autonomes

Exercice 1 — Le gain, avant la formule (20 min). Sur `clients.csv`, écrivez pour `nom`, `ville`, `telephone`, `email`, `conditions_paiement` une ligne de mesure : modalités avant, modalités après normalisation (élagage + casse + ponctuation retirée), lignes modifiées. Classez les colonnes en

trois colonnes : à normaliser pour une clé / à normaliser pour un affichage / à ne pas toucher. Justifiez chaque « à ne pas toucher » par un chiffre.

Exercice 2 — L'import hostile (25 min). Importez `tarif_fournisseur_peinture.csv` deux fois : une fois en Windows-1252 avec 3 lignes sautées, une fois en ISO-8859-1 sans rien sauter. (a) Combien de colonnes dans chaque cas, et pourquoi ? (b) Comptez les caractères dont les deux imports divergent ; vérifiez que ce nombre correspond à une fenêtre précise d'octets, en comparant les deux tables. (c) Le prix `2900,57` : combien de lignes se lisent en nombre, combien en texte, selon la locale choisie ? Écrivez la formule qui rend le résultat indépendant de la locale.

Exercice 3 — Les clés qui manquent (30 min). Construisez la grille complète `magasin × mois` à partir de `objectifs_de_ca.csv` (5 magasins, 44 mois) et joignez-y la table reçue. (a) Combien de cases attendez-vous, combien en trouvez-vous, lesquelles manquent ? (b) Ajoutez `magasins.csv` à la comparaison : que découvre-t-on de plus ? (c) Votre tableau croisé affichera-t-il les trous comme des zéros ? Écrivez la mesure à ajouter pour que la différence soit explicite.

Exercice 4 — Le format comme preuve (20 min). Sur les 412 lignes que `clients_propres.csv` a retirées, testez la règle « le nom porte un espace en surplus » et la règle « le téléphone n'a pas d'espace après le préfixe ». Combien de lignes conservées portent chacune de ces anomalies ? Écrivez la règle de dédoublonnage alternative qui découle de ces deux compteurs, et dites ce qu'elle a de plus fragile qu'une règle sur le contenu.

12. Correction détaillée

Exercice 1. Réponses mesurées : `nom` 12 640 → 12 342 modalités, à normaliser (c'est la clé du C03) ; `ville` 6 → 6, à ne pas toucher ; `telephone` 23 499 valeurs distinctes pour 23 912 lignes, à normaliser en chiffres seulement (413 numéros portés par au moins deux lignes : la normalisation doit réduire l'écriture, pas l'identité) ; `email` rien à reformater, les 16 514 sans `@` sont vides ; `conditions_paiement` à conserver en l'état pour l'affichage, la valeur contractuelle n'est pas une écriture. Le barème est atteint dès qu'une case « à ne pas toucher » porte un chiffre au lieu d'une intention.

Exercice 2. (a) En ISO-8859-1 sans saut de lignes : une seule colonne, nommée d'après la ligne 1, et 42 lignes dont 3 de bruit ; en Windows-1252 avec 3 lignes sautées : 6 colonnes nommées correctement, 38 lignes. Le séparateur est le `;` — la différence de nombre de colonnes vient du séparateur, pas de l'encodage. (b) 95 octets non ASCII, dont 39 dans la fenêtre `0x80-0x9F` : c'est exactement le nombre de caractères qui diffère entre les deux tables, et ce sont les tirets demi-cadratin des libellés ; en Latin-1 ils deviennent des caractères de contrôle non recherchables. (c) Sous une locale à virgule, les 38 prix se lisent en nombres et les marges à point des objectifs deviennent du texte ; l'inverse sous locale à point. La formule insensible est `=CNUMPARAGRAPHE(SUBSTITUE(D2;" ";"");".";",")`, où l'on énonce explicitement le séparateur décimal et le séparateur de milliers au lieu de les hériter du poste.

Exercice 3. (a) 5 × 44 = 220 cases attendues, 218 reçues, 2 manquantes : février et mars 2024 au magasin 4 — consécutives, donc import ou collé interrompu plutôt qu'oubli de saisie. (b) En confrontant à `magasins.csv` (6 lignes), on découvre le magasin 6 : 0 case d'objectif, ce que l'exercice (a) ne pouvait pas voir, faute de référentiel. (c) Non : un tableau croisé affichera une absence si la ligne n'existe pas dans la source, un zéro si elle existe avec 0 — et rien ne distingue les deux à l'écran. La mesure à écrire est un compteur de lignes jointes par `magasin × mois`, comparé au nombre de cellules du croisé ; les deux doivent être égaux, sinon le croisé ment.

Exercice 4. Les deux anomalies sont portées par 412 lignes sur 412, et par **0** ligne conservée. La règle alternative est donc : retirer toute ligne dont le nom contient un espace en surplus et le téléphone n'est pas formaté, ce qui rend exactement les 412 lignes de la référence. Ce qu'elle a de plus fragile : elle décrit un lot d'import passé, pas une propriété du client. Le jour où un poste de saisie soigné produira des doublons bien formatés, la règle ne verra plus rien — alors qu'elle

aura l'air de fonctionner, parce qu'elle rendra 0 ligne à retirer. D'où la conclusion à écrire : on documente la règle de format comme **indice de rejeu**, et la règle métier (nom normalisé + chiffres du téléphone) comme **règle de décision**.

13. Mini-projet M04.P5 — « Le protocole de normalisation » (1 h 30)

Livrable unique : l'onglet `PROTOCOLE` d'un classeur, plus le fichier `ventes_normalisees.csv` qu'il décrit. Le protocole se présente sous la forme d'une table de cinq colonnes, une ligne par transformation :

colonne à produire	contenu attendu
`ordre`	1 à 8, l'ordre d'application, sans trou
`operation`	au tableur : nom de la formule ou de l'étape Power Query
`objet`	la colonne visée, et sa nature (type / écriture / correspondance / unité)
`effet`	deux nombres : lignes modifiées, valeurs distinctes avant → après
`controle`	la case qui prouve que rien n'a été perdu, avec sa valeur mesurée

Quatre exigences de forme, qui seront notées :

- aucune ligne du protocole ne retire de ligne au fichier (le compteur de lignes est identique en tête et en queue du protocole) ;
- le total de la colonne monétaire réparée apparaît avec son écart à la voie de reconstitution, écart écrit à 0 ;
- la normalisation des clés apparaît **avant** la ligne qui décrit la jointure aux objectifs ;
- une ligne du protocole dit explicitement « aucune transformation nécessaire » là où la mesure a montré un gain nul (ville, unité en tant que texte).

Barème (/10) : protocole complet et rejouable, 3 ; contrôles chiffrés et écart de convergence, 3 ; distinction des niveaux type / écriture / correspondance / unité respectée, 2 ; absence de destruction de la source, 1 ; lisibilité de la table (une ligne, une opération, un effet), 1.

14. Résumé du chapitre

Normaliser change l'écriture sans changer la valeur, et se reconnaît à ce qu'il ne retire aucune ligne : les trois niveaux d'anomalie — type, écriture, sens — se traitent séparément, le troisième étant l'affaire des règles de contrôle et non des formules. Sur ce socle, la conversion des 3 421 textes monétaires se fait par un double `SUBSTITUE` qui passe sans exception, et sa justesse se prouve par convergence : 15 639 751 286 FCFA par le découpage, le même nombre par la reconstitution `montant_ht + montant_tva`, écart de 0 franc ; le même geste fait passer le contrôle des retours de 47 lignes ratées à 0. La mise en forme n'est pas un détail : l'espace en trop et le téléphone sans espaces désignent les 412 lignes à retirer avec une exactitude que la ressemblance des noms n'a pas. Le gain d'une normalisation se mesure avant de s'écrire — 298 noms gagnés, 0 sur la ville — et le zéro est une conclusion qui s'écrit. Enfin, la norme ne dit rien du monde : 0 date mal formatée, 51 dates impossibles, et 4 unités physiques que seule une table de conversion rend comparables.

15. À retenir

- **Normaliser n'est pas nettoyer** : le premier garde les lignes, le second les compte et les retire. Écrivez « lignes avant = après » à la fin de toute normalisation.

- **Deux voies indépendantes, un écart à l'unité** : c'est la seule preuve qu'une conversion de type n'a rien inventé. Le contrôle se fait **avant** toute correction de sens, pour que les voies restent indépendantes.
- **Mesurez le gain avant d'écrire la formule** : modalités distinctes avant / après, lignes modifiées. Une formule à gain nul est du décor qui vieillit mal.
- **Un format peut être une preuve** : 412 lignes sur 412 portent deux anomalies d'écriture, 0 ligne conservée. Consignez-le comme indice de rejeu, pas comme règle de décision.
- **La norme n'est pas la justesse** : 0 date hors norme ISO, 51 dates impossibles. Un contrôle de format tout seul ne contrôle rien.
- **Les unités ne se normalisent pas au texte** : `m3` et `litre` demandent un facteur et une table, pas une fonction.

16. Évaluation formative (auto-correction, 8 min)

1. Citez les trois niveaux d'anomalie d'une colonne et l'opération licite à chacun.
2. Pourquoi le contrôle de convergence sur le total doit-il précéder la correction de `taux_remise` ?
3. Combien de lignes des 3 421 textes monétaires ne comportent qu'une seule espace, et qu'est-ce que ça change à la formule ?
4. Une normalisation fait chuter le compteur de lignes de huit unités. Que s'est-il passé, et quelle case du protocole le voyait avant de livrer ?
5. `ville` a 6 modalités avant, 6 après normalisation. Que fait-on de la colonne `ville_normalisee` ?
6. Quels deux réglages d'importation suffisent à rendre le fichier de tarif lisible, et combien de caractères coûtent le mauvais choix d'encodage ?

Corrigé : (1) type → convertir ; écriture → élaguer, harmoniser la casse, retirer les séparateurs ; sens → reconstituer ou documenter, jamais forcer. (2) Parce que la correction de sens rendrait `montant_ht` fonction de la convention choisie : les deux voies ne seraient plus indépendantes et leur accord ne prouverait plus rien. (3) 11 — les montants à trois chiffres (`680 FCFA`) ; rien pour le double `SUBSTITUE`, tout pour une découpe par positions. (4) On a supprimé 8 lignes sous prétexte de format ; le compteur avant/après, écrit avant de commencer, l'aurait signalé. (5) On ne la crée pas : un gain nul s'écrit comme une décision. (6) Séparateur `;` et encodage Windows-1252 (plus 3 lignes sautées) ; 39 caractères, ceux de la fenêtre `0x80-0x9F`.

Module M04.C06 — Documenter, nommer, protéger : la dernière étape est une écriture, pas un calcul

Outil de ce chapitre : le tableur et l'explorateur de fichiers, sur le socle entier (24 fichiers de données), `NOTICE.md`, `verites_terrain.csv` et les paires brut → propre. Durée indicative : 5 h. Niveau : N2.

> **L'idée du chapitre.** Cinq chapitres vous ont appris à rendre un fichier digne de confiance. Le sixième pose la question que personne ne veut poser en réunion : et si vous partiez la semaine prochaine ? La réponse ne tient ni dans une formule ni dans un classeur : elle tient dans quatre artefacts texte — l'arborescence, les noms, le README de données, le journal des transformations — plus un contrôle final qu'on sait lire et un périmètre de protection des données personnelles. Le socle que vous nettoyez est, lui-même, un mauvais exemple : `NOTICE.md` compte 56 lignes et n'y fait mention d'aucun journal, d'aucune version, d'aucun encodage, d'aucune anonymisation — et il n'existe aucun fichier de journal dans `data/reference/`. Le chapitre se termine donc là où le socle s'arrête : en vous demandant d'écrire ce qui manque, pour le socle lui-même.
>
> **Base de travail.** Les 24 fichiers de `01_socle_donnees/data/` (10 dans `brut/`, 12 dans `reference/`, 2 dans `projection/`), `01_socle_donnees/data/reference/NOTICE.md` (56 lignes, 6 sections), `01_socle_donnees/data/reference/verites_terrain.csv`, et les quatre paires brut → propre : `ventes_brutes.csv` (243 360) / `ventes_propres.csv` (240 000), `clients.csv` (23 912) / `clients_propres.csv` (23 500), `produits.csv` / `dictionnaire_produits.csv`, `tarif_fournisseur_peinture.csv` / `dictionnaire_produits.csv`. Tous les nombres viennent de ces fichiers.

---

1. Objectifs du chapitre

À la fin de ce chapitre, vous saurez :

- **organiser** un dossier de données en trois zones — brut, référence, projection — et dire pourquoi le brut se lit mais ne s'écrit jamais ;
- **nommer** des fichiers et des colonnes sans créer les deux pièges du socle : même nom, deux sens ; même chose, deux noms — et chiffrer le premier (12 colonnes portées par deux fichiers ou plus) ;
- **écrire** un README de données de six sections et un **journal des transformations** dont chaque ligne se referme par un compteur avant → après ;
- **lire** un contrôle de cohérence final qui n'est pas vert à zéro : le fichier propre garde 80 collisions, et c'est un résultat correct, pas un échec ;
- **délimiter** le périmètre des données personnelles sur le socle (23 912 clients nommés, 22 vendeurs, 6 responsables, 3 rôles) et chiffrer le risque de ré-identification (87,3 % des clients désignés par quatre colonnes anonymes).

2. Pourquoi cette notion est importante

Tout ce que les cinq chapitres précédents ont produit est **rejetable** — c'est-à-dire qu'il sera rejoué par quelqu'un d'autre, dans un an, sur un fichier qui n'est pas tout à fait le même. Un dédoublonnage sans journal, c'est une décision qu'on devra prendre deux fois : la première fois, on

l'a prise pour de bon ; la seconde, on la prend pour la même. Et en attendant, trois accidents déjà présents dans le socle montrent ce que ça coûte :

- **12 colonnes** portent le même nom dans deux fichiers bruts ou plus (`nom`, `ville`, `region`, `id_client`, `id_magasin`, `id_produit`, `id_vendeur`, `fournisseur`, `tva`, `mois`, `annee`, `annee_mois`) — et `nom` ne veut pas dire la même chose dans `clients.csv` (un client) et dans `magasins.csv` (un magasin). Sans dictionnaire, la jointure `clients.nom = magasins.nom` s'écrit naturellement et produit un fichier absurde que personne ne contrôlera ;
- **2 fichiers** sur 24 portent un nom qui rompt la convention du dossier (`NOTICE.md` en majuscules, `ventes_magasin5_2025_ATTENDU.csv` avec un segment en majuscules) — rien ne casse aujourd'hui, mais le script qui listera « tous mes CSV » en 2027 tranchera sur une convention ;
- le socle **n'a aucun journal des transformations**. On sait que 3 360 lignes ont disparu de `ventes_brutes.csv` en passant à `ventes_propres.csv` — on ne sait pas, dans les livrables, par quelle règle, quand, par qui. Le chapitre C03 l'a déduite en forensique (bloc ajouté en fin de fichier, garde au plus petit identifiant) : une déduction qui a coûté une heure de mesures, et que le jour du rejeu personne n'a l'obligation de refaire si une ligne de journal l'énonce.

La documentation n'est pas la récompense du travail terminé : c'est la condition de sa reproductibilité. Un contrôle qui n'est pas rejouable ne prouve rien — et un nettoyage qui n'est pas écrit ne s'est pas fait.

3. Explication simple

Imaginez une boîte que vous recevez : dix sachets de poudre, une fiche d'instructions en six phrases, aucune date, aucun nom, aucun numéro de lot. Les poudres sont bonnes — vous les avez testées, tout est conforme. Mais vous ne pouvez ni remonter la production (pas de lot), ni prévenir un défaut (pas de date), ni dire à qui il manque un ingrédient (pas de nom). La qualité était là ; la **traçabilité** n'était pas là, et c'est elle que vous vendez vraiment.

Un dossier de données est cette boîte. Le brut est le sachet tel qu'il a été ouvert — on ne le réempaquete pas, on le conserve pour l'expertise. La référence est la fiche de contrôle : ce que les poudres devraient donner. La projection est l'échantillon qu'on donne à qui veut tester sans ouvrir la caisse. Le journal est le carnet de bord : « le 3, on a tamisé la poudre 2 ; avant : 5 kg, après : 4,98 kg ; raison : sable. » Et la protection, enfin, est la décision de ne pas envoyer la boîte à tout le monde : si la fiche porte des noms de clients, l'échantillon qu'on distribue ne les porte plus — ou bien on explique pourquoi on les y laisse.

Chaque pièce se vérifie en une minute, et les six pièces ensemble font la différence entre « les chiffres sont bons » et « les chiffres sont bons **et je peux le démontrer à un inconnu** ».

4. Vocabulaire essentiel

Terme	Ce que ça désigne, dans ce chapitre
Zone de données	Un sous-dossier à règle d'écriture unique : `brut` (lecture seule), `reference` (la vérité), `projection` (les échantillons)
Convention de nommage	La règle commune des noms : `minuscules_et_tirets` ici ; elle se vérifie par un scan, pas par un coup d'œil
Clé d'homonymie	Une colonne qui porte le même nom dans deux fichiers avec deux sens (`nom`) — le piège n° 1 des jointures
README de données	La fiche du dossier : ce que c'est, quelles tables, quelles règles, ce qui manque volontairement
Journal des transformations	Le registre des opérations : date, fichier, opération, règle, compteur avant → après, auteur

Terme	Ce que ça désigne, dans ce chapitre
Contrôle de cohérence final	Le passage complet de la grille de 12 points (C01) sur le fichier propre, lu comme un résultat et non comme un feu vert
Identifiant direct	Une colonne qui désigne une personne sans intermédiaire : nom, téléphone, e-mail
Quasi-identifiant	Une colonne (ou une combinaison) apparemment anonyme qui désigne une personne dans une part mesurable des cas
Anonymisation	Transformer des données pour qu'une personne ne soit plus identifiable — irréversible par construction, ou documentée comme pseudo-anonymisation
Faux anonymisage	Supprimer l'identifiant direct en laissant un quasi-identifiant unique : 87,3 % des clients du socle restent désignés

5. Cours approfondi

5.1 L'arborescence : trois zones, une règle d'écriture chacune

Le socle contient **24 fichiers** de données, répartis en trois sous-dossiers qui ne jouent pas le même rôle :

Zone	Fichiers	Rôle	Règle d'écriture
`brut/`	10	l'arrivée, telle qu'on l'a reçue	lecture seule — jamais réécrite
`reference/`	12	la vérité : fichiers propres, dictionnaires, `verites_terrain.csv`, `NOTICE.md`	s'écrit par décision, jamais à la volée
`projection/`	2	les échantillons distribués aux modules M01/M03	dérivé : se recrée à tout moment

La règle qui rend cette organisation utile n'est pas le tri, c'est l'**asymétrie d'écriture**. Tant qu'une même main peut écrire dans le brut et dans le propre, personne ne sait plus d'où vient un chiffre : le brut est supposé être le point de départ, mais si on y touche, il n'est plus une preuve. Le socle obéit à la règle — c'est pour cela qu'on a pu, dans le C03, raisonner sur les 3 360 lignes disparues en comparant les deux zones sans jamais se demander si la zone de départ avait bougé.

> **Dans les faits.** La projection prouve la règle. `projection/ventes_magasin5_2025.csv` n'est pas un extrait coupé avec le ciseau : il est recréé à chaque génération par le script, et `reference/` porte le fichier `ventes_magasin5_2025_ATTENDU.csv` qui dit ce que doit donner la recomposition. Si demain quelqu'un modifie la projection à la main, la comparaison avec l'`ATTENDU` le révèle. C'est exactement ce que vous devrez faire pour vos propres échantillons : **un échantillon sans son attendu est une zone grise.**

5.2 Le nommage : mesurer la convention avant de la proclamer

Le scan du dossier dit : **68 colonnes distinctes** dans les 10 fichiers bruts, **toutes** en `minuscules_et_ tirets_bas` — aucune majuscule, aucun espace, aucun accent. Côté fichiers, la convention est `minuscules.csv` : **2 fichiers sur 24** la rompent (`NOTICE.md`, `ventes_magasin5_2025_ATTENDU.csv`). Un scan qui matche 22 fichiers sur 24 a déjà une information ; un scan qui ne matche rien n'en a pas — c'est pour cela qu'on écrit le nombre de fichiers testés à côté du nombre de violations, et pas seulement le chiffre « 2 ».

Le vrai risque n'est pas l'orthographe, c'est le **sens**. **12 colonnes** portent le même nom dans deux fichiers ou plus, et voici ce que le socle en fait :

Nom partagé	Où	Ce qu'il désigne dans chaque fichier
`nom`	`clients.csv`, `magasins.csv`	le nom du client / le nom du magasin — deux sens, un nom
`ville`, `region`	`clients.csv`, `magasins.csv`	la ville du client / la ville du magasin — jointure tentante, résultat faux
`fournisseur`	`produits.csv`, `couts_achat.csv`	le fournisseur du produit / le fournisseur du coût — même objet, à vérifier
`id_client`, `id_magasin`, `id_produit`, `id_vendeur`	3, 5, 4, 2 fichiers	des clés — mais `id_client` est texte dans les CSV et nombre dans `remises_manuelles.xlsx`
`mois`, `annee`, `annee_mois`	3, 3, 2 fichiers	des temps — en entier dans le classeur, en texte dans les CSV
`tva`	2 fichiers	le taux dans `produits.csv` et le taux dans `tarif_fournisseur_peinture.csv` — même sens ici (0,18 des deux côtés), mais le dictionnaire doit le certifier, car « tva » peut être un taux d'un côté et un montant de l'autre

Ajoutez à cela l'inverse du piège : **même chose, deux noms** — `nom` dans `clients.csv` et `nom_complet` dans `vendeurs.csv`. C'est la documentation (un dictionnaire de colonnes) qui tranche, jamais le nom. Et le premier fichier qu'on ouvre sans dictionnaire est celui qui casse la jointure.

> **Attention.** Le BOM est le nom qui se change sans qu'on le voie. 5 fichiers bruts sur 10 commencent par trois octets `EF BB BF` ; lus en Windows-1252 (le réglage par défaut de l'assistant d'importation), le premier en-tête de `ventes_brutes.csv` ne s'appelle plus `id_vente` (8 caractères) mais `ï»¿id_vente` (11 caractères). Une colonne dont le premier caractère est invisible change tous les scripts qui la citent par son nom, sans changer un seul chiffre. La convention de nommage couvre donc aussi **l'encodage déclaré** : un nom n'est constant que si tous le lisent de la même façon.

5.3 Le README de données : six sections, pas une de plus

`NOTICE.md` est le README du socle : **56 lignes, 6 sections** — le scénario, les tables, les défauts injectés (avec leur corrigé `rapport_defauts.json`), les règles de gestion, l'extrait M01/M02, et « ce qui n'est **pas** dans le jeu ». Cette dernière section est la plus utile du dossier : elle dit d'avance que la marge nette ne se calcule pas (pas de coûts de structure) et que 2026 est une année partielle — les deux questions que pose nécessairement n'importe quel lecteur.

Le README de données ne contient pas le travail, il contient les **quatre réponses** que le lecteur ne peut pas trouver ailleurs :

1. **Qu'est-ce que c'est ?** — le scénario, en cinq lignes, et la taille réelle : `ventes_brutes.csv` pèse 28 516 920 octets, 243 360 lignes, 20 colonnes ;
2. **Quelles tables, quel grain ?** — une ligne par fichier : « 1 ligne = 1 ligne de vente », « 1 ligne = 1 client » ;
3. **Quelles règles de gestion ?** — TVA 18 %, remise en proportion, retours négatifs, `id_client = 0` = comptoir ;
4. **Ce qui manque volontairement** — et pourquoi ça change le calcul.

Ce que la NOTICE ne contient pas, le chapitre l'a compté : 0 mention d'un **journal**, 0 d'une **version**, 0 d'un **encodage** (BOM, cp1252, séparateur `;` des objectifs), 0 d'une **anonymisation**. Ce ne sont pas des reproches contre le socle : c'est la liste de courses du chapitre. Un README qui ne dit pas comment relire le fichier (encodage, séparateur, en-tête) et comment le re-produire (génération, journal, version) rend chaque lecture un acte de foi.

> **Définition.** Un **README de données** est le document qui répond, sans ouvrir les fichiers : ce que contient le dossier, à quel grain, sous quelles règles de gestion, et ce qui y est volontairement absent. Son critère de qualité n'est pas la complétude mais le **temps de mise en route** : un analyste qui n'a jamais vu le dossier doit pouvoir produire, en moins d'une heure, un premier chiffre que le README lui laisse vérifier.

5.4 Le journal des transformations : une ligne par opération, un compteur par ligne

Le socle nous laisse voir, par comparaison des zones, **quatre opérations** qui ont produit les fichiers propres :

Date	Fichier	Opération	Règle	Avant	Après	Auteur
—	`ventes_brutes.csv` → `ventes_propres.csv`	dédoublonnage	garder le plus petit `id_vente` par ticket + produit (C03)	243 360	240 000	—
—	idem	reconvertion des montants texte	retirer « FCFA » puis les espaces (C05)	3 421 textes	0 texte	—
—	`clients.csv` → `clients_propres.csv`	retrait des quasi-doublons	nom normalisé + chiffres du téléphone (C03)	23 912	23 500	—
—	`produits.csv` → `dictionnaire_produits.csv`	harmonisation des catégories	table de correspondance 15 → 7 (C03)	16 modalités dont 1 vide	7	—

Chaque ligne se **referme** : on peut vérifier l'écart (3 360 = 243 360 − 240 000 ; 412 = 23 912 − 23 500), on peut la rejouer (la règle est citée), on peut la contredire (le chapitre C01 a montré que le corrigé fourni, `rapport_defauts.json`, ne colle pas au fichier — 3 360 annoncés contre 3 421 montants texte mesurés). Le « — » dans les colonnes date et auteur est exactement ce que le journal doit combler : le socle ne les fournit pas.

> **Définition.** Une ligne de **journal des transformations** est une quintuple complète : date, fichier, opération, règle, compteur avant → après (l'auteur faisant la sixième en production). Une ligne incomplète — sans règle ou sans compteur — n'est pas une ligne de journal, c'est une note ; et un journal de notes ne permet ni le rejeu ni l'audit, parce qu'il ne dit ni comment ni combien.
>
> **Conseil professionnel.** Écrivez le compteur **avant** l'opération, pas après. « 243 360 lignes, je vais retirer 3 360 » qui devient « 240 000 restantes » est une ligne de journal ; « j'ai nettoyé les doublons » est une confidence. La différence tient à un nombre écrit à l'avance — et c'est ce nombre qui, dans six mois, permettra de dire si le nettoyage de 2027 a fait le même travail, ou un autre.

5.5 Le contrôle final : un vert à 0 n'existe pas, et c'est une bonne nouvelle

On repasse la grille des 12 points (C01) sur `ventes_propres.csv`. Résultat mesuré, point par point :

Point	Sur le propre	Lecture
3 · complétude	0 cellule vide	le 0 est attendu
5 · unicité métier	**80 groupes** en collision sur ticket + produit + quantité	le 0 n'est **pas** attendu

Point	Sur le propre	Lecture
6 · types	0 montant non numérique	le 0 est attendu
7 · domaines	0 remise hors [0 ; 1]	le 0 est attendu
8 · cohérence TTC = HT + TVA	0 écart	le 0 est attendu
10 · actualité	0 date après le relevé	le 0 est attendu
Total TTC	15 419 985 157 FCFA	écart **0** à `verites_terrain.csv`

Le point 5 n'est pas vert à zéro, et c'est le résultat **correct**. Les 80 groupes (160 lignes) sont des tickets légitimes où le même client — ou pas — achète deux fois le même produit, à la même quantité, **à des heures différentes** : l'heure diffère dans 80 groupes sur 80, le client dans 77, le prix dans 72, et **aucun groupe n'est indistinct** une fois `id_vente` mis de côté (0 sur 80). Le contrôle qui prouve le dédoublonnage n'est donc pas « plus aucune collision », mais : les collisions restantes sont 80, et elles sont toutes distinctes par une autre colonne. Le C03 avait laissé 3 440 collisions dans le brut ; 3 360 sont parties avec le bloc dupliqué ; 80 restaient, et elles devaient rester.

> **À retenir.** Un contrôle final se lit par **résidus attendus**, pas par zéros universels. Écrivez ce que devrait rester (80 groupes multi-lignes, distincts par l'heure), et le contrôle devient capable de distinguer un fichier propre d'un fichier mal dédoublonné — alors que « 0 collision » ne ferait que rejeter le bon fichier.

5.6 Le périmètre des données personnelles : quatre fichiers, un risque mesuré

Où sont les personnes dans le socle ? Mesuré fichier par fichier :

Fichier	Ce qu'il contient	Lignes concernées
`clients.csv`	nom (23 912), téléphone (23 912), e-mail (7 398), ville (22 001), date de création	23 912 — l'intégralité
`vendeurs.csv`	`nom_complet` et `date_embauche` de 22 salariés	22
`magasins.csv`	`responsable` de 6 magasins	6
`remises_manuelles.xlsx`	`saisi_par` : 3 **rôles**, pas de noms (« Assistante commerciale », « Itinérant », « Directeur magasin »)	18 200 lignes, 3 rôles

Le dernier est l'exception qui enseigne : un fichier du socle a déjà pris la bonne décision — séparer la **personne** du **rôle**. Les trois autres la laissent ouverte.

Et voici le chiffre qui doit changer votre façon de « protéger » : si l'on retire les identifiants directs du
fichier clients (nom, téléphone, e-mail) et qu'on garde quatre colonnes anonymes — ville, type, segment, date de
création —, **18 171 combinaisons sur 20 809 désignent un client unique : 87,3 % des clients restent identifiables**.
Trois colonnes seulement (ville, type, segment) ne désignent personne (0 combinaison unique sur 72) ; c'est la
date qui fait tout le travail de ré-identification. Inversement, l'identifiant direct n'est pas ce qu'on croit

7 265 e-mails sur 7 398 (`98,2 %`) encodent l'`id_client` dans leur adresse (`contact2596@…` pour le client 2596) — **l'e-mail est la clé primaire recopiée**. L'effacer au nom de la protection détruirait la jointure avant d'avoir protégé quoi que ce soit.

> **Définition.** Un **quasi-identifiant** est une combinaison de colonnes apparemment anonymes qui désigne une personne dans une part mesurable des cas. Sa part se mesure : 18 171 sur 20 809 combinaisons sont uniques ici. L'anonymisation ne peut pas se définir par la liste des colonnes effacées, elle se définit par cette part, et doit le dire (« après effacement, X % des individus restent désignés par les colonnes conservées »).
>
> **Définition.** Un **identifiant direct** est une colonne qui désigne une personne sans intermédiaire : le nom, le numéro de téléphone, l'adresse e-mail. Sur le socle, la part de lignes touchées est totale pour le nom et le téléphone (23 912 sur 23 912) et de 30,9 % pour l'e-mail (7 398). L'effacement d'un identifiant direct est la première étape, jamais la dernière : les quasi-identifiants font le reste du travail.

5.7 Anonymiser sans casser : l'ordre compte

Deux erreurs, mesurées sur le socle, montrent que l'anonymisation est une **séquence** et non une opération :

- **La troncature qui détruit sans protéger.** Garder les trois premiers chiffres de tous les téléphones du socle ne laisse qu'**une** valeur distincte (tout le monde est `226`) : on a détruit la colonne, pas protégé la personne. À six chiffres, il en reste 90 — de quoi localiser un département, sans doute pas un client. La troncature est un réglage, et comme tout réglage, il se chiffrage : « il reste X valeurs distinctes » écrit à côté du « Y avant ».
- **L'effacement qui casse la jointure.** L'e-mail encodant l'identifiant (7 265 cas) ne peut pas être effacé avant d'avoir été **remplacé par un pseudonyme stable** — sinon les 18 200 lignes de `remises_manuelles.xlsx`, jointes sur `id_client`, se retrouvent sans clé le jour où l'on veut vérifier qui a signé quoi.

L'ordre s'écrit donc : (1) **figer les clés** — décider la clé de jointure et, si besoin, la remplacer par un pseudonyme stable avant tout effacement ; (2) **effacer les identifiants directs** ; (3) **mesurer la part de ré-identification** sur les colonnes conservées (ici : 87,3 % avec quatre colonnes) et abaisser jusqu'à un seuil accepté ou documenter le seuil retenu ; (4) **écrire la décision dans le journal** — « anonymisation le [date], colones X effacées, Y pseudo-anonymisées, part de ré-identification résiduelle Z %, clé stable : `id_client` ». Sans la quatrième ligne, les trois premières sont perdues au prochain export.

> **Attention.** Une pseudo-anonymisation est une **promesse de réversibilité** : le pseudonyme se réattribue, donc la table de correspondance a une valeur — elle est le document le plus sensible du dossier, plus que les données elles-mêmes. La protéger (périmètre de lecture, journal d'accès) fait partie de l'anonymisation, pas du travail suivant.

Livrer une boîte à données — trois zones, un journal, un contrôle qui attend

La documentation n'est pas la récompense du travail : c'est la condition de son rejeu.

brut/	10 fichiers	l'arrivée, telle qu reçue	lecture seule — jamais réécrite
reference/	12 fichiers	la vérité : propres, dictionnaires, verites_terrain	écrit par décision, jamais à la volée
projection/	2 fichiers	les échantillons distribués aux modules	dérivé : recréable, avec son ATTENDU

Le journal se referme par un compteur

ventes : 243 360 → 240 000 (3 360, règle du C03)
montants texte : 3 421 → 0 (convergence à 0 franc)
clients : 23 912 → 23 500 (412, clé nom + chiffres)
catégories : 16 modalités dont 1 vide → 7
avant - après = retirés : sinon, c'est une note, pas une ligne

Le contrôle final attend des résidus, pas des zéros

12 points : 0 vides, 0 texte, 0 domaine, 0 actualité
unicité métier : 80 groupes (160 lignes) — attendus
distincts par l'heure 80/80, client 77, prix 72 ; indistincts : 0
total : 15 419 985 157 FCFA = verites_terrain, écart 0
le résidu attendu s'écrit avant le contrôle

Protéger, c'est chiffrer : 4 fichiers de personnes, un risque mesuré

nom 23 912 · téléphone 23 912 · e-mail 7 398 · ville 22 001 · 22 vendeurs · 6 responsables · 3 rôles
sans nom/tél/e-mail, 87,3 % des clients restent désignés par 4 colonnes anonymes ; l'e-mail encode l'id dans 7 265 cas

6. Exemple concret : le journal du socle, complété

Voici la table de l'§ 5.4 avec les « — » remplis — c'est exactement le livrable attendu au mini-projet :

Date	Fichier	Opération	Règle	Avant	Après	Auteur
2026-09-18	`ventes_brutes.csv` → `ventes_propres.csv`	dédoublonnage	plus petit `id_vente` par (ticket, produit, quantité)	243 360	240 000	analyste (rejeu C03)
2026-09-18	idem	reconvertion `montant_ttc`	`SUBSTITUE` « FCFA » puis espaces, contrôle de convergence à 0 (C05)	3 421 textes	0 texte	idem
2026-09-18	`clients.csv` → `clients_propres.csv`	retrait quasi-doublons	nom normalisé + chiffres du téléphone, 0 faux positif	23 912	23 500	idem
2026-09-18	`produits.csv` → `dictionnaire_produits.csv`	harmonisation catégories	table de correspondance, 0 code inconnu	16 modalités (1 vide)	7	idem
2026-09-18	`clients.csv` → `clients_anonymes.csv`	pseudo-anonymisation	clé `id_client` figée, nom/tél/e-mail remplacés, part résiduelle 87,3 % à documenter	23 912	23 912	idem

Notez la dernière ligne : l'anonymisation **ne change aucun compteur de lignes** (23 912 → 23 912), et c'est normal — elle change les valeurs, pas les lignes. Un journal qui ne sait pas distinguer ces deux effets écrit mal les uns et les autres.

7. Démonstration pas à pas : la boîte à données en huit étapes

Tout se fait dans l'explorateur, le tableur et deux fichiers texte. À chaque étape, un chiffre écrit.

1. **Compter la boîte.** 24 fichiers, 3 zones. Écrire le tableau de l'§ 5.1. Si une zone écrit dans `brut/`, s'arrêter et nommer la main.
2. **Scanner les noms.** Fichiers : convention `minuscules.csv`, 22 sur 24 conformes, 2 à citer. Colonnes : 68 distinctes, 0 hors convention. Colonnes partagées : 12 — pour chacune, une ligne « même nom, quel sens ? » dans le dictionnaire.
3. **Ouvrir chaque fichier et noter** : encodage (5 CSV à BOM), séparateur (`;` pour `objectifs_de_ca.csv` et le tarif), en-tête (ligne 1, sauf le tarif : ligne 4, 3 lignes de bruit), type des clés. Cinq colonnes de notes par fichier : c'est l'embryon du dictionnaire.
4. **Écrire le README** en six sections : scénario, tables, défauts, règles, extrait, absences volontaires. 56 lignes de cible : au-delà, on écrit un livre ; en-dessous de 20, on n'a rien dit.
5. **Écrire le journal** : une ligne par opération du C02-C05, avec le compteur avant écrit **avant** l'opération. Chacune doit se refermer : l'écart avant − après doit égaler le nombre d'objets retirés ou modifiés.
6. **Passer la grille de 12 points** sur chaque fichier propre. Écrire les résidus attendus avant de lancer le contrôle : pour `ventes_propres.csv`, « 80 collisions multi-lignes, toutes distinctes par l'heure ».
7. **Vérifier la convergence** : total du propre = 15 419 985 157 FCFA, écart 0 à `verites_terrain.csv`. Écrire l'écart, pas seulement les deux totaux.
8. **Delimiter le périmètre de protection** : les 4 fichiers de l'§ 5.6, la part de ré-identification mesurée (87,3 %), et la décision (anonymiser / pseudo-anonymiser / restreindre) — avec, dans le journal, la date de la décision, pas seulement son effet.

8. Erreurs fréquentes

Symptôme	Cause réelle	Correction
Jointure « normale » qui produit un fichier absurde	clé d'homonymie : `nom` = client d'un côté, magasin de l'autre	dictionnaire de colonnes ; préfixer les clés (`client_nom`, `magasin_nom`)
La colonne `id_vente` « a disparu » d'un script	BOM : le premier en-tête s'appelle `ï»¿id_vente`	encodage déclaré dans le README ; `utf-8-sig` côté lecteur
« J'ai nettoyé les doublons » — et l'année prochaine, on re-devine la règle	journal sans règle ni compteur	la quintuple complète, compteur écrit avant l'opération
Le contrôle final est « rouge » à cause de 80 collisions	on attendait un zéro universel au lieu du résidu attendu	écrire le résidu attendu avant le contrôle ; vérifier que les 80 sont distinctes
L'anonymisation a « protégé » le fichier — et la jointure ne marche plus	e-mail effacé alors qu'il encodait l'identifiant (7 265 cas)	figer la clé **avant** l'effacement ; pseudonyme stable
« Il ne reste qu'une valeur distincte » après troncature du téléphone	troncature à 3 chiffres : tout le monde est `226`	chiffrer les valeurs distinctes avant/après ; 6 chiffres = 90 ici
Deux fichiers nommés `NOTICE.md` et `*_ATTENDU.csv` au milieu des minuscules	la convention n'était jamais vérifiée	le scan, et le nombre de violations écrit à côté du nombre testé

9. Bonnes pratiques professionnelles

- **Une zone, une règle d'écriture.** `brut` en lecture seule, physiquement si possible (permissions du système de fichiers) ; la comparaison des deux zones n'est possible que tant que la zone de départ n'a pas bougé.

- **Le dictionnaire de colonnes avant la première jointure.** Douze colonnes partagées sur 68, c'est une sur cinq : sans dictionnaire, chaque jointure est un tirage. Une ligne par colonne : fichier, sens, type, exemples.
- **Le README en six sections, 56 lignes de cible.** Scénario, tables, défauts, règles, extrait, absences. La section « ce qui n'est pas dedans » est la plus précieuse : elle coupe les questions avant qu'elles ne posent problème.
- **Le journal se referme par un compteur.** Avant − après = retirés ou modifiés ; sinon la ligne est une note.
- **Le contrôle final liste ses résidus attendus.** 80 groupes multi-lignes sur 240 000 lignes, tous distincts par l'heure — écrits avant le contrôle, pour pouvoir les reconnaître.
- **La protection se chiffre.** Part de ré-identification avant (100 % avec nom), après effacement (87,3 % avec quatre colonnes), et le seuil choisi — ou la décision de le laisser, datée.
- **La table de correspondance d'une pseudo-anonymisation est le document le plus sensible** du dossier : elle ré-attribue les pseudonymes, donc son périmètre de lecture est plus restrictif que celui des données.

10. Exercice guidé — la boîte à données du socle (40 min, /10)

Vous recevez le dossier `01_socle_donnees/data/` sans rien savoir. Vous devez pouvoir le remettre en état de rejeu à votre successeur en 40 minutes.

1. **1 pt** — Comptez les fichiers par zone et le total. Écrivez la règle d'écriture de chaque zone.
2. **1,5 pt** — Scannez les noms : combien de fichiers rompent la convention, lesquels ? Combien de colonnes distinctes, combien portées par deux fichiers ou plus ? Citez les deux pièges de sens avec leur exemple du socle.
3. **1,5 pt** — Relisez `NOTICE.md` : combien de lignes, combien de sections ? Quelles quatre notions n'y sont **pas** mentionnées (journal, version, encodage, anonymisation) ?
4. **2 pt** — Écrivez la ligne de journal du dédoublonnage ventes : opération, règle, avant, après, écart. Montrez que l'écart s'explique (3 360 = 243 360 − 240 000).
5. **1,5 pt** — Passez la grille sur `ventes_propres.csv` : combien de collisions restent, combien de groupes, et par quelle colonne tous les groupes sont-ils distincts ? Écrivez le résidu attendu **avant** de regarder.
6. **1 pt** — Vérifiez la convergence : total du propre, total de `verites_terrain.csv`, écart.
7. **1,5 pt** — Donnez le périmètre des données personnelles : quels fichiers, combien de lignes, et la part de ré-identification si l'on garde ville + type + segment + date de création.

Note /10 : 1 pt par ligne 1-3, 2 pts par ligne 4, 1,5 pt par ligne 5-7, avec 1 point de malus si un compteur est donné sans son dénominateur.

11. Exercices autonomes

Exercice 6.1 (★) — Le scan (15 min). Sur le dossier `data/` : (a) combien de fichiers, par zone ; (b) quels noms de fichiers rompent `minuscules.csv` ; (c) combien de colonnes distinctes, combien non conformes ; (d) listez les 12 colonnes partagées et, pour `nom`, `ville` et `fournisseur`, dites ce que chacune désigne dans chaque fichier.

Exercice 6.2 (★★) — Le journal manquant (20 min). Le socle n'a pas de journal. Écrivez-en les quatre lignes (ventes, montants texte, clients, catégories) avec la quintuple complète. Pour la ligne « montants texte », indiquez le contrôle de convergence qui referme l'opération et sa valeur.

Exercice 6.3 (★★) — Le contrôle qui n'est pas vert (20 min). (a) Combien de collisions ticket + produit + quantité dans `ventes_propres.csv` ? (b) Ces lignes sont-elles des doublons ? Prouvez-le colonne par colonne (heure, client, prix : combien de groupes diffèrent sur chacune). (c) Écrivez le résidu attendu de ce contrôle, formulé pour un lecteur qui n'a jamais vu le C03.

Exercice 6.4 (★★) — La protection chiffrée (25 min). Sur `clients.csv` : (a) combien de combinaisons (ville, type, segment, date de création) désignent un client unique, et quelle part des 23 912 ? (b) Si l'on tronque le téléphone à 3 chiffres, combien de valeurs distinctes restent ? Et à 6 ? (c) Combien d'e-mails encodent l'identifiant client, et quelle en est la conséquence pour l'ordre d'effacement ?

Exercice 6.5 (★★★) — La boîte complète (40 min). Livrez pour le socle : le tableau des zones, le dictionnaire des 12 colonnes partagées, le README en six sections (56 lignes de cible), le journal en quatre lignes, le contrôle final avec ses résidus attendus, et le périmètre de protection avec sa part de ré-identification. Le tout doit se relire sans ouvrir un seul fichier brut.

12. Correction détaillée

Exercice 6.1. (a) 10 + 12 + 2 = 24 fichiers. (b) 2 : `NOTICE.md` et `ventes_magasin5_2025_ATTENDU.csv`. (c) 68 colonnes, 0 non conforme à `minuscules_et_tirets`. (d) `nom` : nom du client (`clients.csv`) / nom du magasin (`magasins.csv`) ; `ville` : ville du client / ville du magasin ; `fournisseur` : fournisseur du produit (`produits.csv`) / fournisseur du coût d'achat (`couts_achat.csv`) — pour celui-là, il faut vérifier dans le dictionnaire que c'est le même objet des deux côtés. Le piège n'est pas la colonne qui « sonne mal », c'est celle qui « sonne juste » : `ville` et `region` sont des jointures parfaitement naturelles à écrire, et fausses.

Exercice 6.2. Les quatre lignes de l'§ 6, avec l'auteur remplacé par votre nom et la date du jour. La ligne « montants texte » se referme par : conversion par double retrait (suffixe puis espaces), 3 421 → 0, contrôle de convergence : somme des valeurs reconverties + somme des 239 939 valeurs déjà numériques = 15 639 751 286 FCFA, égal à la somme `montant_ht + montant_tva` des mêmes lignes, écart 0. Sans ce dernier nombre, la ligne dit qu'on a converti ; avec, elle dit qu'on a converti **sans rien changer**.

Exercice 6.3. (a) 80 groupes, 160 lignes. (b) Non : l'heure diffère dans 80 groupes sur 80, le client dans 77, le prix dans 72, et 0 groupe est indistinct une fois `id_vente` mis de côté. Ce sont des lignes de vente légitimes du même ticket. (c) Résidu attendu : « 80 groupes (160 lignes) en collision sur (ticket, produit, quantité), tous distincts par l'heure ; tout groupe indistinct par toutes les colonnes est un échec du dédoublonnage. » Ce qui rend le contrôle capable de rejeter un mauvais dédoublonnage **sans** rejeter le bon.

Exercice 6.4. (a) 20 809 combinaisons, dont 18 171 uniques — soit 87,3 % des combinaisons désignant un seul client ; en pratique, retirer nom + téléphone + e-mail et garder ces quatre colonnes ne protège que 12,7 % des cas. (b) À 3 chiffres : 1 valeur distincte (tout le monde commence par `226`) — la troncature a tout détruit, personne n'est protégé ; à 6 chiffres : 90. (c) 7 265 e-mails sur 7 398 (`98,2 %`) portent l'`id_client` dans leur adresse : l'e-mail est la clé recopiée, et l'effacer avant d'avoir figé la clé casse la jointure avec les 18 200 lignes de remises.

Exercice 6.5. Attendu : les six pièces de l'§ 7, dont le contrôle final écrit « 12 points, 0 résidu sauf le point 5 : 80 groupes multi-lignes attendus, distincts par l'heure (80/80), 0 groupe indistinct ; le total du propre, 15 419 985 157 FCFA, est égal à `verites_terrain.csv` (écart 0) ». Le piège de l'exercice est la ligne 4 : un dictionnaire qui ne traite que `nom` a raté les 11 autres colonnes partagées — et celles qui tuent les jointures sont justement `ville` et `region`, qui n'inspirent aucun soupçon.

13. Mini-projet M04.P6 — « La boîte à données » (1 h 30)

Livrable : un dossier `boite_ventes/` contenant le socle de ventes **documenté** — c'est-à-dire ce qui se passe si vous partez la semaine prochaine. Huit pièces, dans l'ordre :

1. l'arborescence à trois zones, avec la règle d'écriture de chacune ;
2. le scan de nommage : fichiers et colonnes, avec le nombre testé à côté du nombre de violations ;
3. le dictionnaire des colonnes partagées (les 12), une ligne par colonne ;
4. le README de données en six sections (56 lignes de cible) ;
5. le journal des transformations en quatre lignes, compteur avant écrit en premier ;
6. le contrôle final : grille de 12 points sur le propre, **résidus attendus écrits avant**, total et écart à `verites_terrain.csv` ;
7. le périmètre de protection : 4 fichiers, parts mesurées, décision datée ;
8. le fichier `clients_anonymes.csv` : `id_client` figée, identifiants directs remplacés par des pseudonymes stables, part de ré-identification résiduelle **mesurée et écrite** (attendue : 87,3 % avec les quatre colonnes conservées — ou votre seuil, documenté).

Barème (10 points). Pièces 1-3 complètes et chiffrées (2) · README qui permet la mise en route en moins d'une heure, testée sur quelqu'un qui n'a pas suivi le module (2) · journal dont chaque ligne se referme par son écart (2) · contrôle final avec résidus attendus et écart à la vérité (2) · périmètre de protection mesuré et pseudonymisation réversible et documentée (2). Seuil de validation : 7. Une boîte dont le README ne mentionne pas l'encodage ou le séparateur perd 2 points sur le README, quel que soit le reste : le premier lecteur qui ouvre `objectifs_de_ca.csv` en virgule perdra la demi-heure que le README aurait dû lui rendre.

14. Résumé du chapitre

Documenter, nommer, protéger, c'est écrire **quatre artefacts** avant de déclarer le travail fini, et un **contrôle** qui sait lire ce qui reste. L'arborescence tient en une règle d'écriture par zone — 24 fichiers en trois zones, dont le brut ne s'écrit jamais, et c'est ce qui a permis au C03 de faire sa forensique. Le nommage se vérifie par scan : 68 colonnes conformes, 2 fichiers non conformes, et 12 colonnes partagées dont deux — `ville` et `region` — tueraient n'importe quelle jointure sans dictionnaire. Le README de données répond à quatre questions en 56 lignes ; le journal des transformations referme chaque opération par un compteur avant → après ; le contrôle final attend 80 résidus et pas zéro — et c'est ce qui le rend capable de rejeter un mauvais dédoublonnage sans rejeter le bon ; la protection des personnes se chiffre avant de se faire : 87,3 % des clients restent désignés par quatre colonnes anonymes, et l'e-mail, qu'on croit identifiant direct, est en réalité la clé primaire recopiée dans 7 265 adresses sur 7 398. Le socle n'avait aucun de ces artefacts : il a fallu les écrire. C'est le chapitre entier en une phrase.

15. À retenir

> **À retenir. ★ Une zone, une règle d'écriture.** Le brut se lit, ne s'écrit jamais : c'est ce qui rend la comparaison des zones une preuve (243 360 → 240 000, 3 360 expliquées). ★ **Même nom, deux sens** est le piège n° 1 des jointures : 12 colonnes partagées sur 68, et `nom` désigne un client d'un côté, un magasin de l'autre. Le dictionnaire tranche, pas le nom. ★ **Le journal se referme par un compteur.** Avant − après = retirés ; sans règle et sans compteur, c'est une note. ★ **Un contrôle final attend des résidus, pas des zéros.** 80 collisions multi-lignes sur 240 000 lignes, toutes distinctes par l'heure (80/80), 0 indistincte — écrit **avant** le contrôle. ★ **La protection se chiffre.** 87,3 % des clients désignés par quatre colonnes anonymes ; l'e-mail encodant l'identifiant dans 7 265 cas sur 7 398 : on fige la clé **avant** d'effacer l'identifiant.

16. Évaluation formative (auto-correction, 8 min)

Q1. Pourquoi le brut est-il en lecture seule ? → parce que la comparaison des zones (brut vs propre) est la preuve du nettoyage ; une zone de départ réécrite n'est plus une preuve, et le journal perd son référentiel.

Q2. Le scan de nommage trouve 2 fichiers non conformes sur 24. Que faut-il écrire, et pourquoi pas seulement « 2 » ? → le nombre testé (24) à côté du nombre de violations (2) : un scan qui ne dit pas ce qu'il a vu est un faux négatif poli — et les 2 noms (`NOTICE.md`, `*_ATTENDU.csv`) nommés, car ce sont des conventions à arbitrer, pas des coquilles.

Q3. Le contrôle final du fichier propre rend 80 collisions. Est-ce un échec ? → non : 80 groupes de lignes légitimes (160 lignes), tous distincts par l'heure (80/80), 0 indistincts hors `id_vente`. Le résidu attendu s'écrit avant le contrôle ; c'est lui qui rend le contrôle capable de distinguer un propre d'un mal dédoublonné.

Q4. On retire nom, téléphone et e-mail des clients. Combien de clients restent désignés si l'on garde
ville + type + segment + date de création ? → 18 171 combinaisons uniques sur 20 809, soit 87,3 % des combinaisons

> l'effacement des identifiants directs ne protège que la minorité des cas. La date de création fait le travail des trois autres colonnes à elle seule (0 combinaison unique sans elle).

Q5. Pourquoi l'e-mail ne peut-il pas être effacé en premier ? → parce qu'il encode l'`id_client` dans 7 265 cas sur 7 398 (`98,2 %`) : c'est la clé primaire recopiée. L'effacer avant d'avoir figé un pseudonyme stable casse la jointure avec les 18 200 lignes de remises sans protéger davantage.

Question ouverte (la plus importante). Votre chef vous demande « est-ce qu'on peut envoyer le fichier clients à l'agence de Lyon ? ». Répondez en quatre phrases, avec les chiffres du socle. Corrigé attendu : (1) le fichier contient 23 912 personnes nommées, dont 7 398 joignables par e-mail et 22 001 localisées en ville ; (2) si l'on anonymise mal, 87,3 % restent ré-identifiables par quatre colonnes anonymes — l'envoi « protégé » ne protège que 12,7 % ; (3) l'e-mail est la clé de jointure (7 265 cas encodent l'identifiant), il ne peut partir qu'après pseudonymisation stable, sinon les 18 200 remises deviennent invérifiables ; (4) décision : soit un extrait pseudonymisé avec part résiduelle mesurée et datée au journal, soit la restriction du périmètre — l'envoi du brut n'est pas une option, il est hors périmètre.

Projet M04.P — « Le dossier de données de la quincaillerie de Koudougou »

Module M04 · projet de fin de module · 6 h · barème /20, seuil de passage 13/20

> **Le cadre réel.** Un gérant de quincaillerie vous a remis un dossier tel qu'il le tient dans son poste : un fichier de commandes de 160 lignes, un fichier de clients, un tarif fournisseur « généré par son logiciel », et une phrase : « je sais que c'est mal tenu, mais les totaux du mois doivent sortir lundi ». Ce dossier contient **sept défauts de natures différentes**, dont **deux qui cassent les totaux** — pas plus, pas moins. Vous rendez la boîte en état de rejeu : un diagnostic chiffré, un fichier réparé dont on sait d'où viennent les nombres, un journal, un README, un contrôle final qui sait ce qui doit rester, et un périmètre de protection pour les clients. C'est la synthèse des six chapitres : chaque livrable en mobilise un, et aucun n'est noté pour lui-même.
>
> **Note de cohérence.** Le dossier est généré de façon déterministe par `python3 tools/dossier_M04.py` (graine 41) : qui que vous soyez, vous touchez les mêmes 160 lignes. `03_exercices/dossier_M04/ATTENDU.json` contient les résultats attendus, **mesurés sur le dossier généré, jamais déduits** : vous n'y recourez qu'après avoir produit vos propres nombres, pour comparer, pas pour copier. Un candidat qui lit l'`ATTENDU` avant de calculer n'a plus rien à rendre. Les chiffres de ce corrigé sont ceux du dossier livré, vérifiables ligne à ligne.

---

1. Énoncé

Le matériel

Élément	Chemin	Ce que c'est
Commandes	`03_exercices/dossier_M04/commandes_2026.csv`	160 lignes, 20 colonnes, encodage UTF-8 avec BOM, séparateur virgule
Clients	`03_exercices/dossier_M04/clients_2026.csv`	132 lignes, 11 colonnes, UTF-8 avec BOM
Tarif fournisseur	`03_exercices/dossier_M04/tarif_fournisseur.csv`	10 lignes utiles, séparateur point-virgule, encodage Windows-1252, en-tête décalé
Résultats attendus	`03_exercices/dossier_M04/ATTENDU.json`	à utiliser **après** le calcul, jamais comme source de copie
Outils	un tableur (Excel 2021+ ou LibreOffice Calc) ; Power Query si votre poste l'a	sinon : la voie accessible, et vous l'écrivez

Le dossier contient sept défauts de natures différentes ; deux d'entre eux faussent le total de la somme directe, les cinq autres faussent une analyse, une jointure ou un import — sans jamais changer un seul montant. Trouver les sept, les séparer en deux natures, et prouver la séparation, c'est le premier tiers de la note.

Les six livrables

D1 — La grille du dossier (3 points). La grille des 12 points (C01), passée sur `commandes_2026.csv` et `clients_2026.csv` : pour chaque point non nul, le compteur **avec son dénominateur** (« 17 sur 160 », jamais « 17 »). Vous identifiez les sept défauts, vous les classez en casse un total / fausse une analyse, une jointure ou un import, et vous écrivez la phrase qui prouve la

séparation : quels sont les deux défauts qui faussent la somme directe, et pourquoi les cinq autres ne la faussent pas.

D2 — Le fichier réparé et son total (4 points). `commandes_2026_propres.csv` : 152 lignes, total TTC **8 209 940 FCFA** — à obtenir par votre méthode, pas à viser. Les 8 lignes retirées s'écrivent dans la feuille de travail avec leur motif : 7 copies exactes (retirées sur la règle du contenu) et 1 **resaisie** — une ligne jumelle qui ne diffère de son original que d'un champ (la date) et porte le plus grand `id_vente` ; le piège de l'énoncé, c'est que le fichier compte **9 groupes en collision** : le neuvième est un ticket multi-lignes **légitime** (deux lignes qui diffèrent sur l'heure, le client, le prix et le mode de paiement) et ses 2 lignes doivent survivre. Une règle qui ne garde qu'une ligne par groupe (le plus petit `id_vente`, sans regard) tombe à 151 lignes et 8 105 333 FCFA : vous écrivez ce que ce total a mangé.

D3 — Le journal des transformations (3 points). Cinq lignes au format du C06 (date, fichier, opération, règle, compteur avant → après), couvrant au minimum : la conversion des 17 montants texte (dont 2 négatifs qui sont des retours), le retrait des 8 doublons, la reconstruction des 12 dates hors norme, et le retrait des 2 quasi-doublons clients. Chaque ligne se referme : avant − après = retirés ou modifiés.

D4 — Le README du dossier et le dictionnaire des trois fichiers (3 points). Six sections, 56 lignes de cible (C06) : scénario, tables avec leur grain, règles de gestion (retour = montant négatif ; `id_client = 0` = comptoir, **pas** une donnée à remplir — 18 lignes dans le dossier), défauts constatés, ce qui manque volontairement. Le dictionnaire déclare pour chaque fichier : encodage (les deux CSV du dossier portent un BOM ; le tarif est en Windows-1252 — un import en Latin-1 y insère des caractères de contrôle invisibles), séparateur (`;` pour le tarif), position de l'en-tête (ligne 4 pour le tarif, 3 lignes de bruit avant), et le type de chaque clé. Les 10 prix du tarif sont à virgule décimale : le README dit comment les lire sur une machine à point.

D5 — Le contrôle final avec ses résidus attendus (3 points). La grille de 12 points sur le fichier réparé, avec les résidus attendus **écrits avant** de lancer le contrôle (C06) : 0 montant non numérique, 0 remise hors [0 ; 1] — à condition d'avoir traité les 5 remises en points, sinon 5 résidus —, 0 date hors norme ISO, 0 groupe de doublon, 1 groupe multi-lignes légitime à 2 lignes. La convergence est écrite avec l'écart : votre total 8 209 940 FCFA contre l'`ATTENDU.json`, écart 0 — l'écart est le chiffre, pas le total.

D6 — Le périmètre de protection de `clients_2026.csv` et sa version pseudonymisée (4 points). Vous délimitez les données personnelles (nom, téléphone : 132 lignes ; e-mail : les lignes renseignées ; ville, date de création), vous mesurez ce que l'effacement des identifiants directs ne protège pas, et vous produisez `clients_2026_anonymes.csv` : `id_client` figée et remplacée par un pseudonyme stable, nom/téléphone/e-mail remplacés, **part de ré-identification résiduelle mesurée et écrite**. Le dossier compte 12 noms portés par deux téléphones ou plus (homonymes) : vous écrivez pourquoi « dédoublonner sur le nom » y retirerait des clients réels, et pourquoi la règle qui retire exactement les 2 quasi-doublons (132 → 130) est la clé nom normalisé + chiffres du téléphone — pas le nom seul.

---

2. Ce qui est noté, ce qui est piégé

Piège du dossier	Où il se cache	Ce qu'il enseigne
17 montants en texte (« n nnn FCFA »), dont 2 négatifs	`commandes_2026.csv`, colonne N	la somme directe rate 17 lignes ; les 2 négatifs sont des retours que le contrôle de domaine ne voit qu'après conversion
8 lignes en doublon (7 copies + 1 resaisie)	`commandes_2026.csv`	la resaisie n'est pas une copie exacte : la règle « contenu identique » ne la voit pas, la règle « un seul champ différent » la voit

Piège du dossier	Où il se cache	Ce qu'il enseigne
1 ticket multi-lignes légitime (2 lignes, 8 colonnes différentes)	`commandes_2026.csv`	« une ligne par groupe » mange une vente réelle (8 105 333 au lieu de 8 209 940)
5 remises en points de pourcentage (12/18/25/30/18)	`commandes_2026.csv`, colonne K	la division par 100 est un test à faire, pas une réparation à appliquer
12 dates en deux écritures (JJ/MM/AAAA et AAAA-MM-JJ)	`commandes_2026.csv`, colonne C	le tri chronologique et le pivot mensuel sont faux tant que les deux écritures cohabitent
2 quasi-doublons clients (nom surchargé + téléphone sans espaces) + 12 homonymes	`clients_2026.csv`	la règle « même nom » retire bien plus que les 2 doublons ; la clé juste est nom normalisé + chiffres
3 lignes de bruit, cp1252, `;`, décimales à virgule	`tarif_fournisseur.csv`	l'import par défaut lit 1 colonne, un en-tête faux, et des prix en texte

3. Barème et seuil

- D1 : 3 points — grille complète avec dénominateurs (1) · les sept défauts en deux natures (1) · la phrase de séparation des totaux (1).
- D2 : 4 points — 152 lignes et 8 209 940 FCFA obtenus (1,5) · les 8 lignes retirées motivées ligne à ligne (1) · le groupe légitime survit et l'erreur de la règle naïve est chiffrée (1,5).
- D3 : 3 points — cinq lignes au format complet (1) · chaque ligne se referme par son écart (1) · la conversion des montants se referme par un contrôle de convergence (1).
- D4 : 3 points — six sections lisibles sans ouvrir les fichiers (1) · encodages, séparateurs et en-têtes déclarés (1) · la convention `id_client = 0` écrite comme convention et non comme donnée manquante (1).
- D5 : 3 points — résidus attendus écrits avant le contrôle (1) · grille du réparé avec le groupe légitime nommé (1) · convergence écrite avec l'écart contre l'`ATTENDU` (1).
- D6 : 4 points — périmètre délimité et chiffré (1) · pseudonymisation stable qui ne casse aucune jointure (1) · part de ré-identification résiduelle **mesurée et écrite** (1) · la règle des 2 quasi-doublons opposée à la règle du nom seul, avec les 12 homonymes en argument (1).

Seuil de validation : 13/20. Un livrable dont un compteur manque son dénominateur perd 0,5 point par compteur, sans plafond : la précision du dénominateur est la matière du module, pas une coquille.

4. Correction des points d'attention

- **D1, séparation des totaux.** La somme directe sur la colonne N ne rate que les lignes non numériques : 17. Les doublons la faussent en double comptant 8 lignes. Les 5 remises, les 12 dates, les 2 quasi-doublons et l'encodage du tarif ne changent aucun montant : ils faussent une analyse (remises), un calendrier (dates), une jointure (clients) et un import (tarif). La phrase attendue : « Deux défauts cassent le total — 17 montants en texte non sommés et 8 lignes comptées deux fois ; les cinq autres laissent la somme intacte et cassent chacune une autre lecture du fichier. »
- **D2, le groupe légitime.** Le groupe formé du ticket `T01-230101-000140`, produit 74, quantité 3 : deux lignes qui diffèrent sur l'heure, le vendeur, le client, le prix, les trois montants et le mode de paiement. C'est une double vente réelle du même ticket, pas un doublon. La règle naïve « une ligne par groupe, plus petit `id_vente` » la réduit à 1 ligne : 151 lignes, 8 105 333 FCFA — l'écart 104 607 FCFA est le montant de la ligne mangée.
- **D2, la resaisie.** Le groupe du ticket `T01-230103-000122`, produit 103, quantité 3 : deux lignes identiques sauf la date. Ce n'est pas une copie exacte (le contrôle de contenu la laisse pas-

ser) ; c'est une ressaisie avec une erreur de date, reconnaissable à son `id_vente` plus grand. La règle de garde la retire sans toucher au groupe légitime, qui diffère sur 8 colonnes.

- **D5, les résidus attendus.** « 0 non numérique, 0 hors domaine, 0 hors norme, 0 groupe de doublon, **1 groupe multi-lignes légitime à 2 lignes** » — le résidu est écrit avant le contrôle ; un contrôle qui ne s'attendrait qu'à des zéros rejetterait le bon fichier.
- **D6, la part résiduelle.** Elle se mesure sur les colonnes conservées (ville, type, segment, date de création, conditions de paiement, plafond) : sur ce dossier, elle se compte en combinaisons désignant un client unique. L'e-mail, quand il est renseigné, encode souvent l'identifiant : le pseudonyme stable doit exister **avant** l'effacement, sinon la jointure avec `commandes_2026.csv` se rompt.
- **Le tarif.** Lu en Windows-1252 avec 3 lignes sautées : 10 lignes, 6 colonnes, 10 prix à virgule décimale. Lu en UTF-8, il casse à l'octet ; lu en Latin-1 sans saut, il donne une colonne et un en-tête faux. Le README déclare les trois réglages ; la formule de lecture des prix est celle qui énonce la virgule, pas celle qui l'hérite de la locale.

Évaluation M04 — « Qualité, préparation et documentation des données »

Module M04 · durée totale 2 h 30 · quatre épreuves · seuils : quiz 11/15, exercices 14/20, étude de cas 12/20

Épreuve	Durée	Barème	Noté	Ce qui est attendu
A · Questions de récupération	15 min	non noté	non	vérifier qu'on a lu les six chapitres avant de se tromper cher
B · Quiz	25 min	/15	oui	quinze questions à réponse unique, quatre blocs
C · Exercices pratiques	50 min	/20 (auto-corrigé)	oui	quatre exercices menés jusqu'au chiffre sur le dossier du projet
D · Étude de cas	60 min	/20, seuil 12	oui	un dossier en quatre pages : « le stock indiqué est-il crédible ? »

> Le score du module se lit ainsi : **B ≥ 11** et **D ≥ 12** valident le module ; le projet M04.P doit atteindre 13. Un candidat qui réussit B et échoue à D repasse uniquement D : savoir ce qu'est une quasi-identification ne dispense pas de dire si le comptage de 60 000 lignes est croyable. Tout nombre de cette évaluation est mesuré sur le socle livré ; les corrections le citent avec leur clé `m04_*/m04p_*` dans `chiffres_cites.json`.

A · Questions de récupération (non notées)

À traiter sans document, en une ou deux phrases chacune. Elles ne sont pas comptées ; elles servent à repérer ce qui n'est pas assimilé avant de perdre des points sur un détail.

1. Citez les cinq mécanismes d'absence du C02 et le traitement qui va avec chacun. Pourquoi `id_client = 0` (43 681 lignes, 2 813 668 030 FCFA) n'est-il aucun des cinq ?
2. Donnez les quatre niveaux où l'on peut chercher un doublon (C03) et, pour chacun, le résultat mesuré sur le fichier de ventes (0 / 0 / 3 440 / 8 453).
3. Citez les quatre familles de règles de contrôle (C04) avec un nombre mesuré chacune — et la famille qui a rendu 24 180 alertes pour 0 erreur.
4. Quelles sont les deux voies indépendantes qui reconvertissent les 3 421 montants texte (C05), et quel nombre commun rendent-elles avant dédoublonnage ?
5. Quelles sont les quatre pièces que le C06 attend d'un dossier livré, et quel résidu **attendu** le contrôle final du fichier propre doit-il accepter (80 groupes) ?
6. Pourquoi les 973 remises à 12, 18, 25, 30 ne changent-elles aucun total, et pourquoi la division par 100 est-elle une réparation qui échoue son propre test ?
7. Quelle est la différence entre un identifiant direct et un quasi-identifiant, avec le chiffre du socle qui rend la distinction coûteuse (18 171 combinaisons uniques sur 20 809) ?

B · Quiz (15 questions · 15 points)

Barème : 1 point par question, une seule bonne réponse. Les réponses justifiées en une ligne rapportent un demi-point de bonus, plafonné à 15.

Bloc 1 — Types et normalisation (Q1 à Q4)

Q1. La colonne `montant_ttc` contient 3 421 cellules du type `n nnn FCFA` parmi 239 939 nombres. Ce que `=SOMME()` fait des 3 421 : a) les somme comme des nombres · b) les ignore, sans erreur · c) renvoie `#VALEUR!` · d) elles sont lues sans leur espace de milliers.

Q2. Normaliser `clients.nom` (élagage, minuscule, ponctuation retirée) fait passer 12 640 valeurs distinctes à 12 342. Conclusion : a) 298 clients ont été supprimés · b) 298 écritures différentes d'un même nom se sont rassemblées, sans que le nombre de clients change · c) 298 doublons exacts ont été fondus · d) la colonne est corrompue.

Q3. Deux écritures d'un même téléphone : `+226 78 62 21 81 98` et `+2267862218198`. La colonne qui sert de clé de jointure doit porter : a) le texte tel quel · b) les chiffres seuls · c) le nom du client · d) la ville.

Q4. `date_vente` est à 100 % en `AAAA-MM-JJ`, et 51 lignes portent une date postérieure au 31/08/2026, date du relevé. L'action correcte : a) une formule de reformatage · b) un contrôle de domaine (date ≤ date du relevé) et une décision documentée · c) la suppression des 51 lignes · d) aucune, le format est correct.

Bloc 2 — Doublons et clés (Q5 à Q8)

Q5. La clé primaire rend 243 360 valeurs distinctes, mais 3 440 lignes sont en collision sur (ticket, produit, quantité). Conclusion : a) la clé primaire est fausse · b) la clé métier voit ce que la clé technique ne voit pas · c) le fichier est entièrement dupliqué · d) c'est une erreur de mesure.

Q6. 8 453 lignes partagent (ticket, produit) alors que la règle de garde n'en retire que 3 360. Explication : a) un choix aléatoire · b) un même ticket peut légitimement porter plusieurs lignes sur un même produit (quantité, heure ou client différents) · c) la clé compte mal · d) les 8 453 sont toutes des doublons.

Q7. La règle « même nom » sur les 23 912 clients fusionnerait 11 570 lignes. Réponse correcte : a) l'appliquer puis vérifier a posteriori · b) la rejeter : elle retire 11 158 lignes de trop, des homonymes réels · c) la limiter aux noms courts · d) l'appliquer aux noms avec espaces.

Q8. Les 412 quasi-doublons clients portent tous deux anomalies d'écriture (nom surchargé, téléphone sans espaces), et 0 ligne conservée ne les porte. Ce qui s'en déduit : a) l'anomalie de format est la règle de dédoublonnage · b) l'anomalie de format est un indice de rejeu, pas une règle : un doublon bien formaté ne serait plus vu · c) le fichier est corrompu · d) il faut supprimer tous les espaces du fichier.

Bloc 3 — Règles et valeurs aberrantes (Q9 à Q12)

Q9. `prix_unitaire_ht` : minimum 575, maximum 45 000 FCFA ; la règle des quartiles en signale 24 180 lignes. Ces 24 180 lignes sont : a) des erreurs de saisie · b) la dispersion légitime du métier (du sac de ciment à la palette d'ossature) · c) du bruit de mesure · d) des doublons.

Q10. La bande [0,95 ; 1,35] autour du prix catalogue rend 0 violation ; l'égalité au catalogue rend 232 981 « anomalies ». Leçon : a) le catalogue est faux · b) une règle qui signale 96 % d'un fichier sain est une règle fausse, pas un fichier sale · c) il faut corriger les 232 981 prix · d) il faut utiliser le z-score.

Q11. 973 lignes ont `taux_remise` à 12, 18, 25 ou 30, mais un `montant_ht` correct et une TVA cohérente. La division par 100 laisse 258 lignes au-dessus du plafond de 0,25. L'action défendable : a)

diviser par 100 · b) reconstituer le taux depuis les montants (étendue 0,000 à 0,110, les 973 dans le domaine) et documenter · c) supprimer les 973 lignes · d) laisser tel quel sans note.

Q12. Le contrôle `qte_physique < seuil_alerte` déclenche sur exactement les mêmes 1 800 lignes que `qte_physique < 0`, sur 60 000. Conclusion : a) le seuil est parfait · b) le seuil ne discrimine rien : le signal réel est le signe ; les deux règles doivent être scindées (intégrité / risque) · c) les 1 800 lignes sont des erreurs à supprimer · d) le seuil est trop bas.

Bloc 4 — Documentation et protection (Q13 à Q15)

Q13. Le fichier propre garde 80 groupes en collision sur (ticket, produit, quantité), tous distincts par l'heure, 0 indistincts. Le fichier est-il correctement dédoublonné ? a) non, à refaire · b) oui : le résidu est attendu, écrit avant le contrôle, et les groupes sont des multi-lignes légitimes · c) c'est un bug du script · d) oui, tant que le résidu reste sous 100 groupes.

Q14. Une ligne de journal des transformations sans règle ni compteur avant → après est : a) une note, pas une ligne de journal · b) une ligne valide, abrégée · c) une exigence d'audit · d) un risque juridique seulement.

Q15. On retire nom, téléphone et e-mail des clients, on garde (ville, type, segment, date de création) : 18 171 combinaisons sur 20 809 désignent un client unique. Conclusion : a) le fichier est anonymisé · b) 87,3 % des combinaisons restent ré-identifiables : effacer l'identifiant direct n'est pas anonymiser · c) seuls `2 638` combinaisons ne désignent personne d'unique · d) il suffit de retirer la ville.

Corrigé du quiz

Q1 **b** — `SOMME` ignore le texte sans erreur : c'est ce qui rend le défaut silencieux (écart du total naïf : 2 111 789 FCFA). Q2 **b** — 12 640 − 12 342 = 298 écritures rassemblées ; le nombre de lignes est inchangé. Q3 **b** — les chiffres seuls : c'est la colonne rare discriminante du C03. Q4 **b** — la norme est respectée, la réalité non ; la décision se documente, elle ne se corrige pas par formule. Q5 **b** — l'unicité technique n'est pas l'unicité métier. Q6 **b** — 8 453 lignes partagent (ticket, produit), mais la clé métier complète (avec la quantité) distingue les multi-lignes légitimes. Q7 **b** — 11 570 − 412 = 11 158 lignes de trop. Q8 **b** — la règle de format décrit un lot passé ; la règle de décision reste la clé. Q9 **b** — 24 180 alertes pour 0 erreur. Q10 **b** — la règle se juge à ce qu'elle attrape, pas au volume d'alertes. Q11 **b** — la reconstitution tient sur les 973 ; la division échoue sur 258. Q12 **b** — deux règles au lieu d'une, un destinataire par niveau. Q13 **b** — le contrôle attend des résidus, pas des zéros universels. Q14 **a** — sans règle ni compteur, la ligne ne referme rien. Q15 **b** — 87,3 % de combinaisons uniques ; la date de création fait le travail des trois autres colonnes.

C · Exercices pratiques (50 min · /20)

Tout se fait sur `03_exercices/dossier_M04/` (le dossier du projet, 160 lignes ; ATTENDU.json à consulter **après** vos calculs).

E1 — La grille du dossier (5 points). Passez la grille des 12 points sur `commandes_2026.csv`. Note attendue : les quatre compteurs non nuls **avec dénominateurs** (17 montants texte sur 160 ; 5 remises hors domaine sur 160 ; 12 dates hors ISO sur 160 ; 8 lignes en doublon sur 160), plus la mention des 18 lignes au comptoir comme convention et non comme donnée manquante. 1 pt par compteur justifié, 1 pt pour la séparation « total / analyse ».

E2 — Le dédoublonnage du dossier (5 points). Produisez 152 lignes et le total **8 209 940 FCFA**. Note attendue

les 8 lignes retirées motivées (7 copies exactes + 1 resaisie reconnue à son champ unique, la date), le groupe légitime (2 lignes, 8 colonnes différentes) survécut, et le chiffre de la règle naïve (151 lignes, 8 105 333 FCFA, l'écart 104 607 FCFA identifié comme le montant de la ligne mangée). 2 pts pour le résultat, 2 pour la motivation ligne à ligne, 1 pour l'opposition à la règle naïve.

E3 — Conversion et import (5 points). (a) Convertissez les 17 montants texte par le double retrait (suffixe puis espaces) et montrez que les 2 retours à montant texte ne passent le contrôle de domaine (retour ⇒ montant négatif) qu'après conversion : 1,5 pt. (b) Importez `tarif_fournisseur.csv` en déclarant l'encodage Windows-1252, le séparateur point-virgule et les 3 lignes à sauter ; lisez les 10 prix à virgule décimale par une formule qui énonce la virgule ; écrivez le coût du mauvais choix d'encodage (39 octets dont le décodage Latin-1 fait des caractères de contrôle invisibles) : 3,5 pts.

E4 — « Les trois chiffres faux » (5 points). Le corrigé enseignant `rapport_defauts.json` du socle annonce 3 360 montants en texte, 2 640 dates en formats mixtes, 1 880 clients sans ville. Mesurez les trois valeurs réellement livrées — 3 421, 0 et 1 911 — par un compteur unique chacune (3 pts), et écrivez la phrase qui dit pourquoi un corrigé se génère **après** la fabrication du fichier, jamais avant (2 pts). Mention attendue : le cas « 61 catégories hétéroclites annoncées, 44 ou 61 selon la définition » est une question de définition, pas une troisième erreur — confondre les deux est le piège de l'exercice.

D · Étude de cas (60 min · /20, seuil 12) — « Le stock indiqué est-il crédible ? »

Contexte. Le magasinier a remis `stocks_quotidiens.csv` : 60 000 lignes, 7 colonnes (`date`, `id_produit`, `id_magasin`, `qte_theorique`, `qte_physique`, `seuil_alerte`, `valeur_stock_ht`). La direction demande : peut-on se fier au stock indiqué ? Vous rendez un dossier de quatre pages.

Le fichier, mesuré. 154 produits × 6 dépôts = 924 séries, sur 1 339 jours distincts (du 01/01/2023 au 31/08/2026). 0 cellule vide. `qte_theorique` : 0 à 1 795, **aucun négatif**. `qte_physique` : **1 800 négatifs** (3,0 % des lignes), présents sur les 154 produits sans exception. `seuil_alerte` : 0 à 448. Le contrôle `qte_physique < seuil_alerte` déclenche sur exactement les mêmes 1 800 lignes que `qte_physique < 0`. Écart théorique − physique : moyenne 2,77, maximum 532, **118 lignes au-dessus de 100**, 2 au-dessus de 500.

Le livrable, en quatre parties :

1. **Ce que le fichier est** (description et grain) : 3 points.
2. **Le verdict de crédibilité, en deux temps** : comme comptage (1 800 négatifs impossibles sur 154 produits : le comptage n'est pas fiable) et comme signal (les 1 800 alertes sont exactement les 1 800 négatifs : le seuil n'ajoute rien, le signal utile est le signe) : 4 points.
3. **Le calibrage de l'écart** : avec les valeurs mesurées et leurs dénominateurs, délimitez la file de travail — 1 800 négatifs à recompter + 118 écarts au-dessus de 100 à instruire — et dites pourquoi 532 de maximum sur une moyenne de 2,77 change la nature du problème (queue, pas bruit) : 4 points.
4. **Trois règles qui rendraient le prochain fichier crédible**, chacune avec son niveau (bloquant / corriger / surveiller / ignorer tracé) et son destinataire : (a) intégrité — quantité physique négative ⇒ recomptage programmé (niveau 2, magasinier) ; (b) risque — seuil recalibré pour un taux d'alerte traitable (les 118 plus grands écarts d'abord, niveau 3, responsable de dépôt) ; (c) journal de comptage — date, produit, dépôt, théorique, physique, écart, compteur (le journal que ce fichier n'a pas) : 4 points.

Grille de qualité (complément des 4 parties, 5 points). Ce que le fichier ne peut pas dire : 3 points — `qte_theorique` est supposé juste, or le logiciel de gestion n'est pas dans le dossier : la cohérence externe (ventes ↔ mouvements de stock) est impossible à vérifier ici, et le dossier le dit. Reproductibilité : 2 points — chaque chiffre du dossier est reproducible par une formule ou un compteur nommé.

Ce qui est piégé dans l'étude. (1) Répondre « oui, c'est fiable » ou « non, c'est faux » d'un bloc : le verdict doit être en deux temps. (2) Proposer de supprimer les 1 800 lignes négatives : ce sont le signal, pas le bruit. (3) Citer un écart sans son dénominateur. (4) Oublier que le seuil actuel est **redondant** avec le signe — la règle qui compte deux fois le même événement n'est pas une règle, c'est du doublon de contrôle (le mot du C03, réemployé au contrôle).

Lecture des résultats

- **B ≥ 11 et D ≥ 12** : module validé (le projet M04.P doit, de son côté, atteindre 13/20).
- **B < 11** : le socle des définitions n'est pas acquis ; repasser B et la récupération A, le module reste ouvert.
- **D < 12 avec B ≥ 11** : le candidat sait ce qu'est un quasi-identifiant mais ne sait pas dire si un comptage de 60 000 lignes est croyable ; repasser D uniquement.
- Le projet M04.P se note séparément (13/20 de seuil) et pèse sur la validation finale du module.