Question 1 : Pourquoi mesurer le taux de rejet plutôt que le nombre brut ?
Le taux de rejet permet de contextualiser les rejets par rapport au volume total traité. Un rejet de 100 lignes représente 1% sur 10 000 lignes mais 50% sur 200 lignes, ce qui n'a pas la même gravité. Le taux permet de détecter les anomalies indépendamment des variations de volume. Il facilite aussi la comparaison entre différentes périodes ou différentes sources de données.

Question 2 : Taux de rejet passant de 3% à 18% en une journée
Plusieurs hypothèses peuvent expliquer cette augmentation brutale. Un changement de version de l'application mobile pourrait générer des données mal formatées. Une mise à jour de la base de données source a pu modifier le schéma ou les contraintes. Un problème technique comme une panne partielle du système de capture pourrait corrompre les données. Une modification du pipeline de validation sans test préalable pourrait rejeter à tort des données valides. Enfin, un événement exceptionnel comme une grève ou des conditions météo extrêmes pourrait générer des données atypiques mais réelles.

Question 3 : Transformation Hop pour calculer des agrégations
La transformation principale est "Group by" qui permet de calculer des agrégations comme AVG, COUNT, MAX, MIN et SUM sur des groupes de données. Elle permet aussi de grouper par plusieurs colonnes et d'appliquer plusieurs fonctions d'agrégation simultanément.

Question 4 : Différence entre COUNT(*) et COUNT(colonne)
COUNT(*) compte toutes les lignes du dataset, y compris celles avec des valeurs NULL dans certaines colonnes. COUNT(colonne) compte uniquement les lignes où cette colonne spécifique a une valeur non NULL. Par exemple, sur 1000 lignes avec 50 valeurs NULL dans la colonne passenger_count, COUNT(*) retournera 1000 tandis que COUNT(passenger_count) retournera 950.

Question 5 : Pourquoi historiser un dashboard de qualité ?
L'historisation permet de détecter les tendances et dégradations progressives de la qualité des données. Elle facilite l'analyse des causes racines en corrélant les problèmes avec des événements passés comme des déploiements ou incidents. Elle permet de mesurer l'efficacité des actions correctives mises en place. Elle sert de preuve pour les audits et la conformité réglementaire. Enfin, elle aide à établir des baselines et des seuils d'alerte pertinents basés sur l'historique réel.

Question 6 : Risque d'utiliser uniquement la moyenne pour total_amount
La moyenne est sensible aux valeurs extrêmes et peut masquer des problèmes importants. Par exemple, 99 courses à 10€ et 1 course à 10 000€ donnent une moyenne de 109€ qui ne représente aucune course réelle. Elle ne détecte pas les distributions bimodales ou asymétriques. Elle cache les outliers qui peuvent être des fraudes ou des erreurs. Il faut donc la compléter avec la médiane, l'écart-type, et les percentiles pour avoir une vision complète de la distribution.

Question 7 : À partir de quel seuil une donnée devient un problème ?
Le seuil dépend du contexte métier et de l'impact business. C'est le métier qui définit les seuils acceptables car il connaît les conséquences business, comme un taux de rejet maximum tolérable de 5% avant impact sur la facturation. Le data engineer apporte son expertise sur la faisabilité technique et les contraintes système. Les seuils peuvent varier selon le criticité des

données, par exemple plus strict pour les données financières que pour les données de localisation. Les SLA et contraintes réglementaires peuvent aussi imposer des seuils.

Question 8 : Différence entre métrique descriptive et métrique d'alerte
Une métrique descriptive fournit une information pour comprendre l'état des données, comme le nombre total de lignes traitées ou la moyenne des montants. Une métrique d'alerte déclenche une action quand un seuil est franchi, comme le taux de rejet supérieur à 5% qui génère une notification. Les métriques descriptives servent à l'analyse et au reporting, tandis que les métriques d'alerte servent au monitoring opérationnel et à la détection d'anomalies en temps réel.

Question 9 : Pourquoi exporter les métriques en SQL plutôt qu'en CSV ?
Une table SQL permet des requêtes complexes et du drill-down pour analyser les tendances historiques. Elle facilite l'intégration avec des outils de BI et de monitoring comme Grafana ou Tableau. Elle garantit la consistance et l'intégrité des données avec des contraintes et transactions. Elle permet la mise à jour et l'enrichissement incrémental sans réécrire tout le fichier. Elle supporte nativement le versioning et l'historisation. Enfin, elle offre de meilleures performances sur les gros volumes avec des index appropriés.

Question 10 : Le dashboard détecte-t-il les données propres mais incohérentes métier ?
Non, pas automatiquement, car un dashboard standard vérifie la qualité technique mais pas la cohérence métier. Par exemple, 100 courses de 2km facturées 500€ chacune sont techniquement valides mais métier absurdes. Il faut définir des règles métier spécifiques comme le ratio distance/prix cohérent ou détecter les valeurs aberrantes avec des percentiles. Le dashboard doit inclure des métriques métier calculées en collaboration avec les experts du domaine. Les contrôles de cohérence croisée entre plusieurs colonnes doivent être explicitement programmés dans le pipeline.