27/01/2026

Data Gouvernance

Dashboard & Monitoring

Yousra Bouhanna, Abdeljebbar Abid, Elias Ait Hassou
MASTER 2 BUSINESS INTELLIGENCE & ANALYTICS – LYON2

Table des matières

1. Contexte et objectifs

Le jeu de données utilisé correspond à un contexte hospitalier fictif, avec plusieurs tables décrivant les patients, le personnel, les consultations, l'occupation des services et les plannings du staff. Ces données sont exploitées dans le cadre d'un projet de qualité des données, afin d'évaluer la fiabilité de l'information disponible pour le pilotage de l'activité.

L'objectif spécifique de cette partie du projet est de concevoir et mettre en œuvre un dashboard de suivi de la qualité des données, permettant de monitorer les principaux piliers de qualité (complétude, exactitude, validité, cohérence, unicité, actualité) ainsi que des KPI associés.

2. Architecture technique et pipeline

2.1 Vue d'ensemble du pipeline

La couche de visualisation s'inscrit dans un pipeline global de qualité des données structuré en quatre étapes principales : stockage → profilage/exploration → validation → visualisation→ gouvernance.

1. **Stockage des données sources**
   Les fichiers CSV (patients, staff, consultations…) sont d'abord importés dans une base relationnelle. Cette étape fournit un socle unique et structuré sur lequel s'appuient le profilage, la validation et la visualisation.

2. **Profilage et exploration**
   Les données chargées sont ensuite explorées et profilées afin de comprendre leurs distributions, leurs valeurs manquantes et leurs anomalies, ce qui permet

de définir des règles de qualité pertinentes par pilier (complétude, exactitude, etc.).

3. **Validation avec Great Expectations**
   Les règles de qualité de données sont mises en place et exécutées dans la phase de validation. Pour chaque règle, des métriques de validation sont calculées, puis exportées dans deux fichiers de sortie :

   - **validation_history :** historique des validations, qui conserve, pour chaque run, table, colonne, pilier et règle, les indicateurs en question.
   - **superset_validation_metrics :** état courant des métriques de qualité, agrégé par table, pilier, règle et colonne.

4. **Visualisation dans Apache Superset**
   Ces deux sorties sont chargées dans PostgreSQL puis exposées comme datasets dans Apache Superset. validation_history est utilisée pour analyser l'historique détaillé des validations, tandis que superset_validation_metrics reflète l'état actuel de la qualité des données et alimente principalement les indicateurs de synthèse du dashboard. En complément, d'autres KPI sont définis directement dans Superset via des requêtes SQL afin d'enrichir l'analyse au-delà des métriques brutes produites par Great Expectations

2.2 Docker Compose : environnement reproductible

Le fichier docker-compose.yml décrit l'ensemble des services nécessaires au projet; pour la partie visualisation, deux services sont essentiels :

- Le service Postgres instancie la base dq_db avec l'utilisateur dq_user et stocke les données ainsi que les résultats de validation dans un conteneur dédié.
- Le service Superset est configuré pour se connecter à cette base Postgres, héberger l'interface web de dataviz et charger les datasets utilisés par les dashboards de qualité.

Enfin, l'enseignant peut démarrer tout l'environnement en une seule commande **docker compose up -d**, ce qui assure une mise en route rapide et reproductible sur toute machine équipée de Docker.

2.3 Scripts SQL et tables de résultats

Les scripts SQL du répertoire db-init (**01_schema_confluence, 02_load_sql**) créent puis chargent les tables sources à partir des fichiers CSV (patients, staff, consultations, services_weekly, staff_schedule, etc.). Cela permet de reconstituer en une seule étape le socle de données nécessaire dans PostgreSQL avant de construire les vues de visualisation.

3. Règles de qualité et indicateurs

3.1 Dimensions de qualité

- **Complétude :** champs obligatoires renseignés (pas trop de valeurs nulles/manquantes).
- **Exactitude :** valeurs au bon format ou norme (téléphone, email, code postal).
- **Validité :** valeurs dans des domaines autorisés (genres, services, plages d'âge…).
- **Cohérence :** relations logiques entre champs (âge vs date de naissance, départ après arrivée, FK valides).
- **Unicité :** absence de doublons sur une clé ou combinaison de colonnes.
- **Actualité :** données suffisamment récentes.

3.2 Règles de qualité implémentées

Ces règles ont été définies dans la phase de validation avec Great Expectation :

- **Complétude :** *_id_not_null, telephone_70pct_filled.
- **Exactitude :** telephone_format_FR, email_format_rfc5322, code_postal_fr_form.
- **Validité:** age_range_18_75, genre_in_allowed_values, service_in_allowed_value.
- **Cohérence :** age_date_naissance_coherent, departure_after_arrival, *_fk_valid.
- **Unicité :** *_unique, combinaisons de colonnes (weekstaffidcombinationunique, patientconsultationunique, etc.).
- **Actualité :** arrival_date_since_2020.

3.3 Calcul des métriques

Dans validation_history.csv, chaque ligne correspond à l'exécution d'une règle : on y stocke checkspassed, checksfailed, successrate.

- **Calcul des compteurs par règle**
    - checks_passed = nombre de lignes qui respectent la règle.
    - checks_failed = nombre de lignes qui ne respectent pas la règle.

On en déduit les formules suivantes :

- $success\_rate = \frac{checks\_passed}{total\_expectations} \times 100$
- $error\_rate = \frac{checks\_failed}{total\_expectations} \times 100$

- **Agrégations des métriques**
    - Par pilier :
        - Moyenne simple des taux de succès sur l'ensemble des règles d'un pilier donné :

$$AVG(success\_rate)$$

        - Ou approche « pondérée » par le volume de données :

$$\frac{\sum checks\_passed}{\sum checks\_passed + \sum checks\_failed} \times 100$$

    - Par table :
        - Même logique, mais en filtrant par tablename puis en agrégeant checks_passed et checks_failed de toutes les règles de cette table.
    - Par règle
        - Il suffit de lire directement les colonnes checkspassed, checksfailed, totalexpectations, successr ate pour la paire (tablename, rulename), sans entrer dans le détail fonctionnel de la règle à ce stade.

4. Construction du dashboard Superset

4.1 Datasets et connexion à la base

On définit dans Superset une connexion PostgreSQL en utilisant l'URI suivante :

postgresql://dq_user:dq_pass@postgres:5432/dq_db

- postgresql : dialecte SQLAlchemy pour Postgres, utilisé par Superset.
- dq_user / dq_pass : utilisateur applicatif et mot de passe dédiés à la data quality, avec des droits en lecture sur le schéma cible.
- postgres : nom d'hôte du service Postgres (dans notre cas, le service Docker postgres résolu via le réseau docker-compose).

- 5432 : port standard de PostgreSQL.
- dq_db : base contenant les tables sources (patients, staff, consultations, etc.) et les tables de résultats de validation consommées par Superset.

Dans Superset, cette URI est saisie dans la page Settings → Databases → + Database, ce qui permet ensuite de déclarer les datasets sur les tables de dq_db sans ressaisir les credentials.

4.2 Organisation en trois onglets

Tab 1 – Overview

Cette première vue donne un score global de qualité sur le dernier run, puis le détail par pilier et par colonne.

- **Big Number "Overall Data Quality Score"**
  Score global calculé comme le taux de succès sur l'ensemble des règles
- **Gauge filtrable par pilier**
  Jauge affichant le score de qualité pour un pilier (Complétude, Exactitude, etc.), recalculé avec la même formule mais en filtrant sur pilier.
- **Bar chart "Data Quality by Dimension"**
  Histogramme montrant le success_rate moyen pour chaque pilier, ce qui permet d'identifier rapidement les dimensions les plus faibles.
- **Radar "Radar Pillars"**
  Ce graphique radar affiche, sur un seul visuel, le score de qualité de chacun des piliers pour le run sélectionné. Il n'est pas filtrable par pilier afin de conserver en permanence une vue d'ensemble de toutes les dimensions de qualité en même temps.

Des filtres natifs **Pillar, Table** et **Date** sont placés à gauche du dashboard et pilotent simultanément tous ces graphiques, ce qui permet de voir l'overview pour un run donné et/ou un seul pilier de qualité.

Tab 2 – Trends over Time

Cet onglet suit l'évolution de la qualité d'un run à l'autre.

- **Line chart "Overall Quality Trend"**
  Courbe du score global de qualité en fonction de daterun.
- **Line chart "Quality Trend by Dimension"**
  Courbes par pilier (une série par dimension) utilisant pilier sur la légende pour suivre quelles dimensions se dégradent ou s'améliorent.

- **Heatmap date × pilier**
  Carte de chaleur où les cellules représentent le success_rate par date de run et pilier, pratique pour repérer visuellement les jours « rouges ».
- **Box plot “Rule Score Distribution by Dimension”**
  Boxplot des success_rate par pilier, pour montrer la dispersion des scores de règles à l’intérieur de chaque dimension (pilier homogène vs très hétérogène).

Les mêmes filtres **Pillar** et **Date** permettent de zoomer sur une période courte ou sur un sous-ensemble de dimensions.

Tab 3 – Detailed Analysis

Cette vue sert à analyser en détail où se situent les erreurs (par pilier, table et règle).

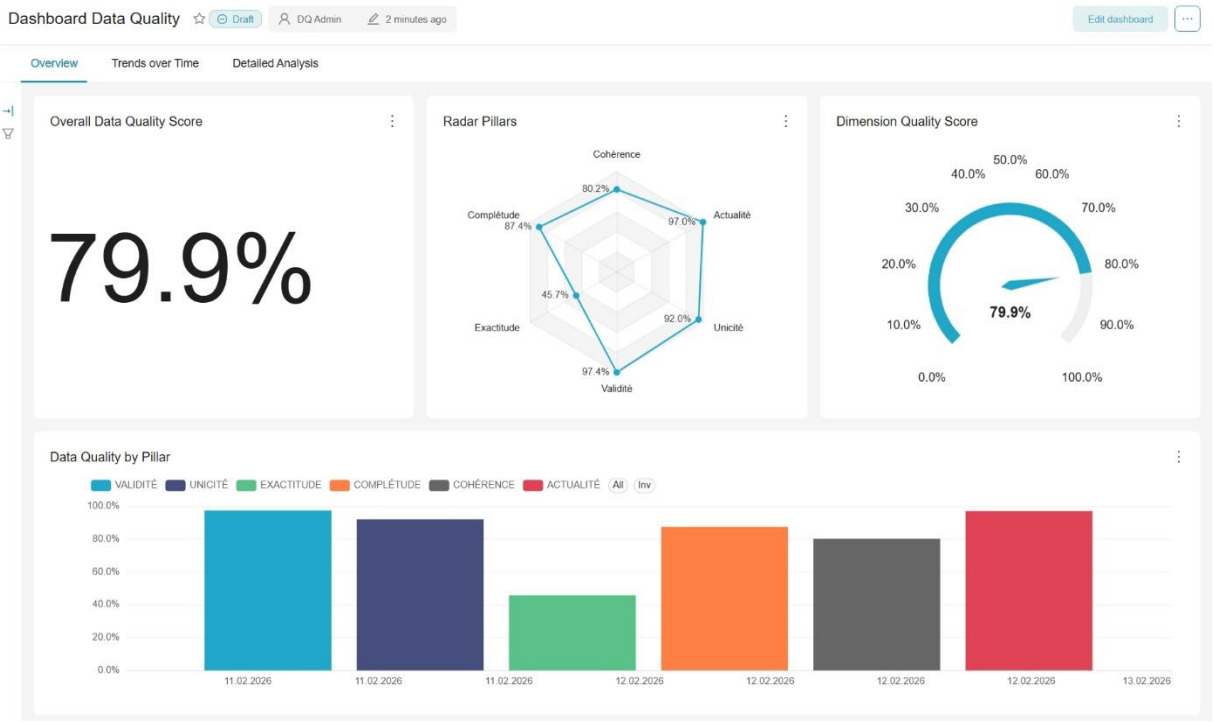
- **Pie chart “Répartition des erreurs par pilier (pondérée)”**
  Part de chaque pilier dans le volume total d’erreurs, calculée à partir de checks_failed agrégés par pilier.
- **Table “Répartition des erreurs par table (pondérée)”**
  Agrégation des checks_failed par tablename
- **Sankey “Error Flow from Tables to Dimensions”**
  Diagramme de Sankey qui montre les flux d’erreurs de la table source vers le pilier de qualité, les largeurs de liens étant proportionnelles à checks_failed.

Les filtres natifs **Pillar** et **Date** s’appliquent également à cet onglet.

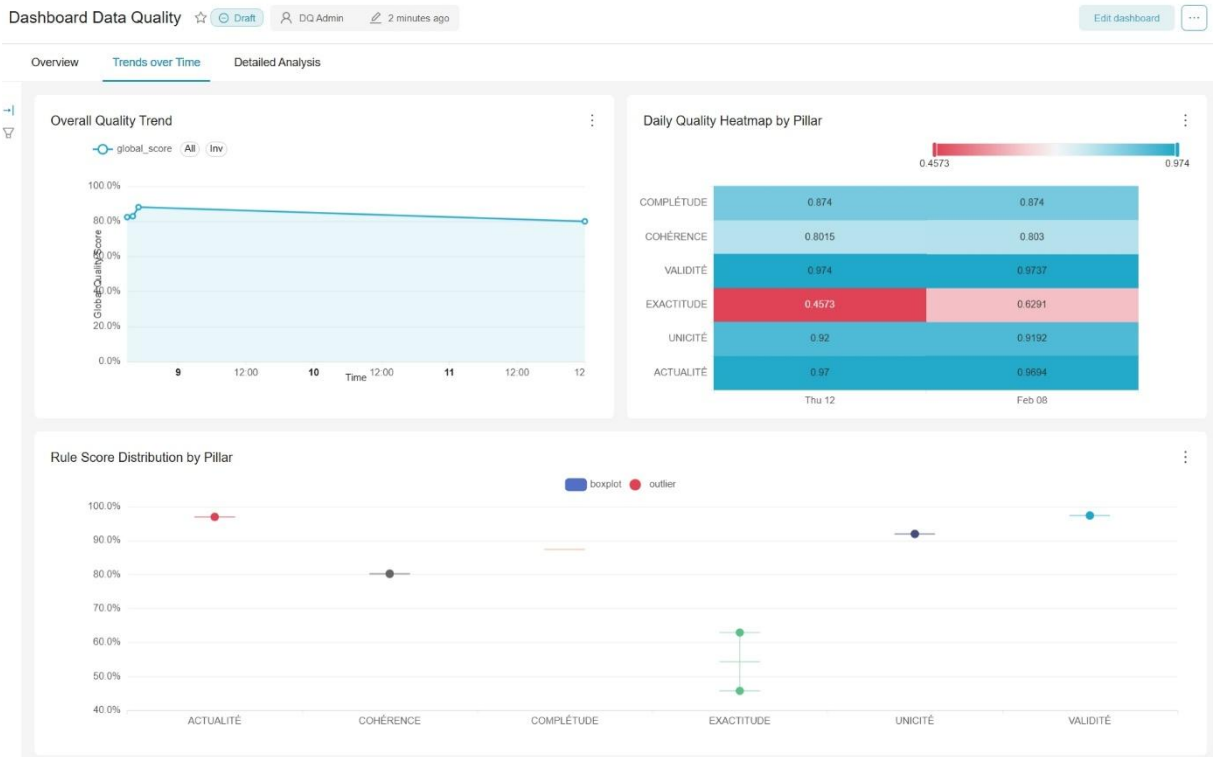
5. Résultats et interprétation

5.1 Présentation des onglets

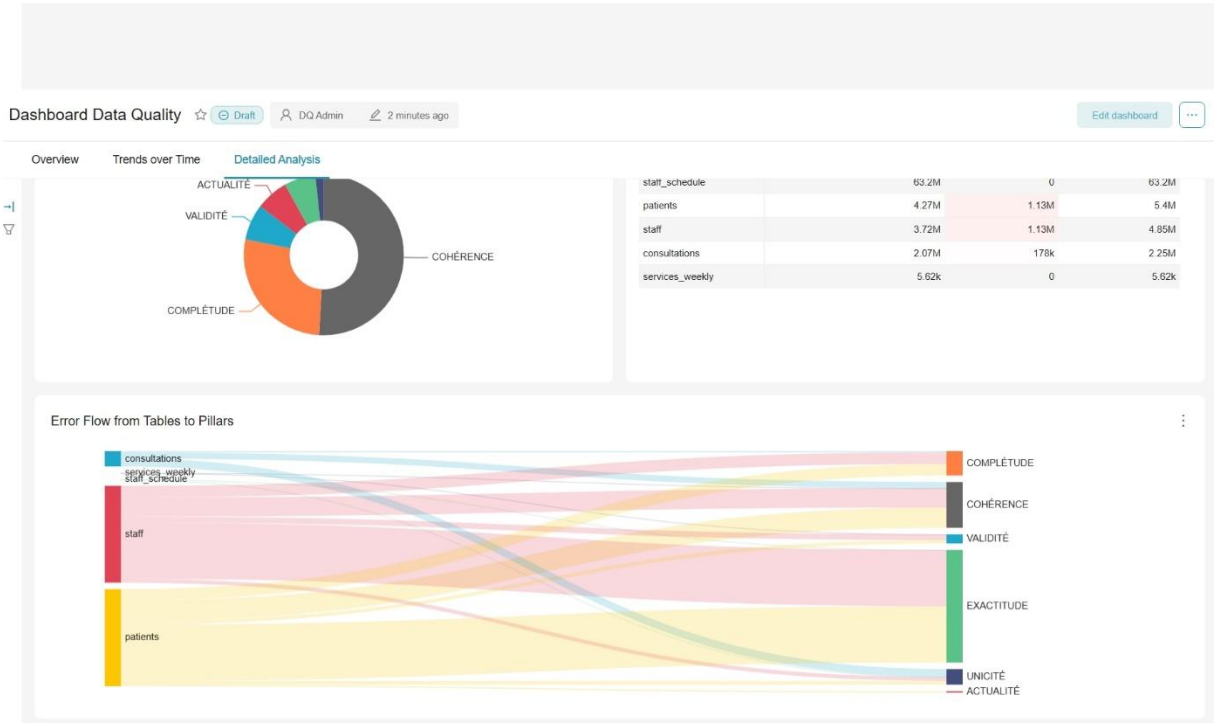
Tab 1 – Overview



Tab 2 – Trends over Time

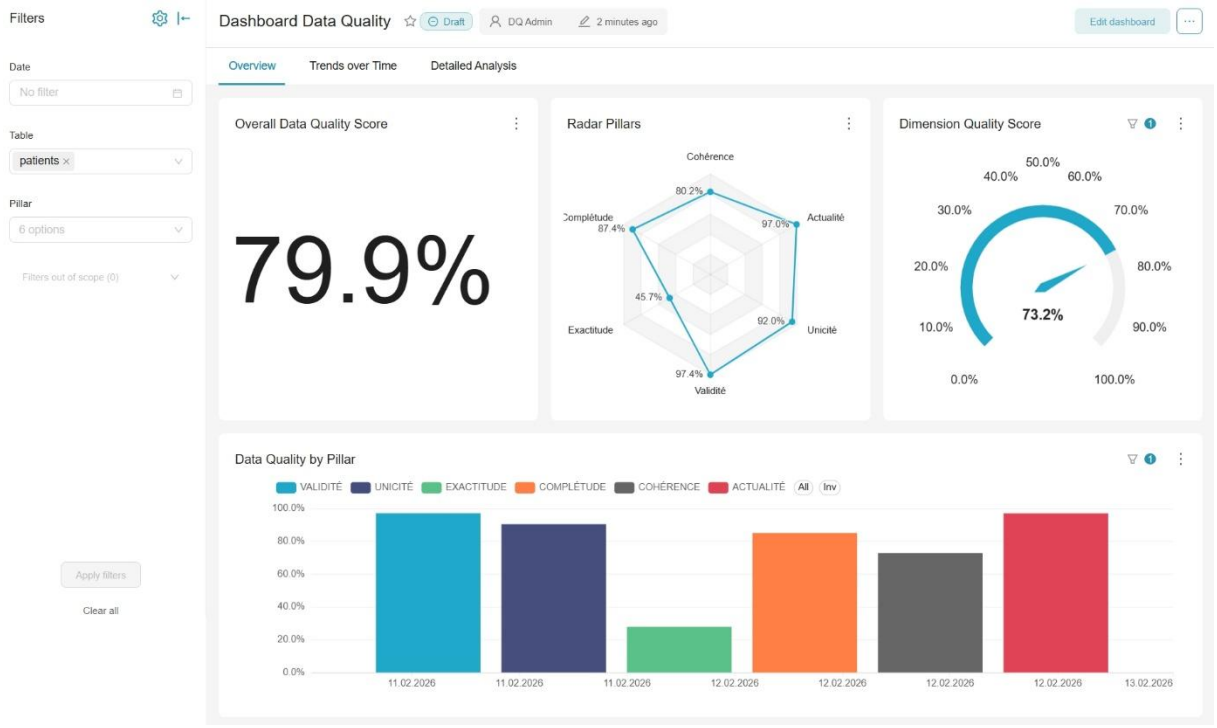


Tab 3 – Detailed Analysis



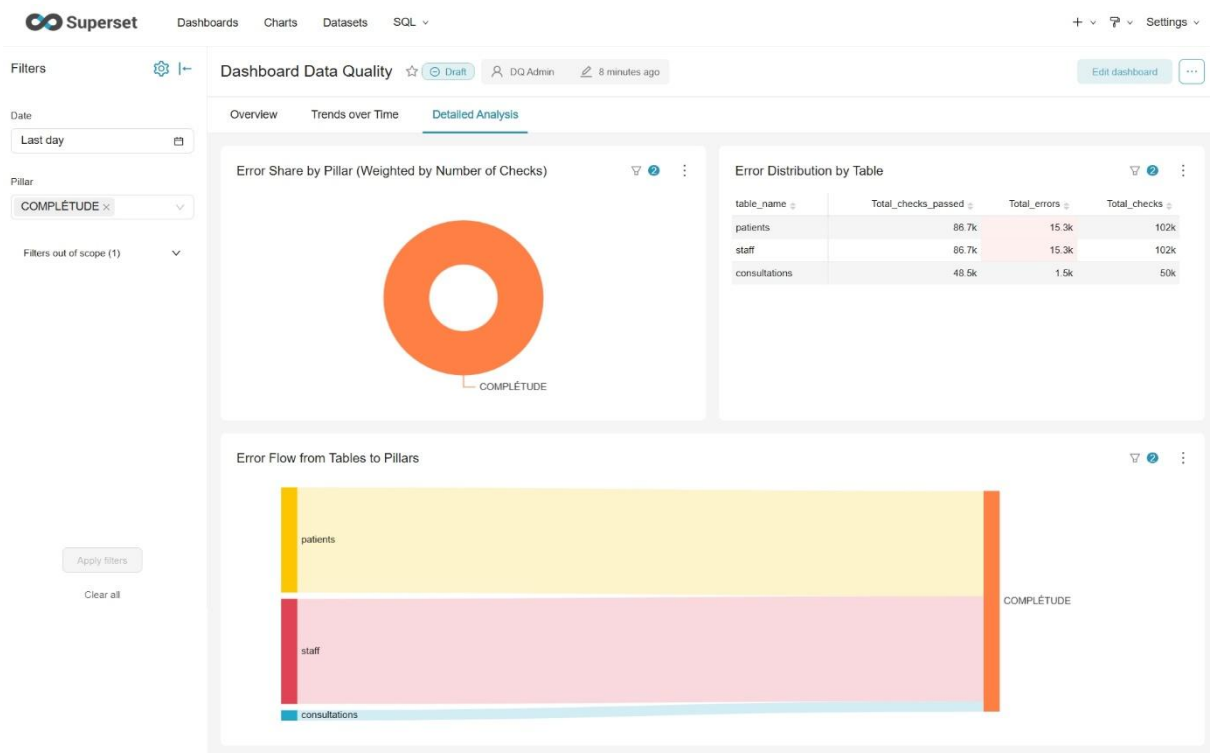
5.2 Effets des filtres

- Filtre Table



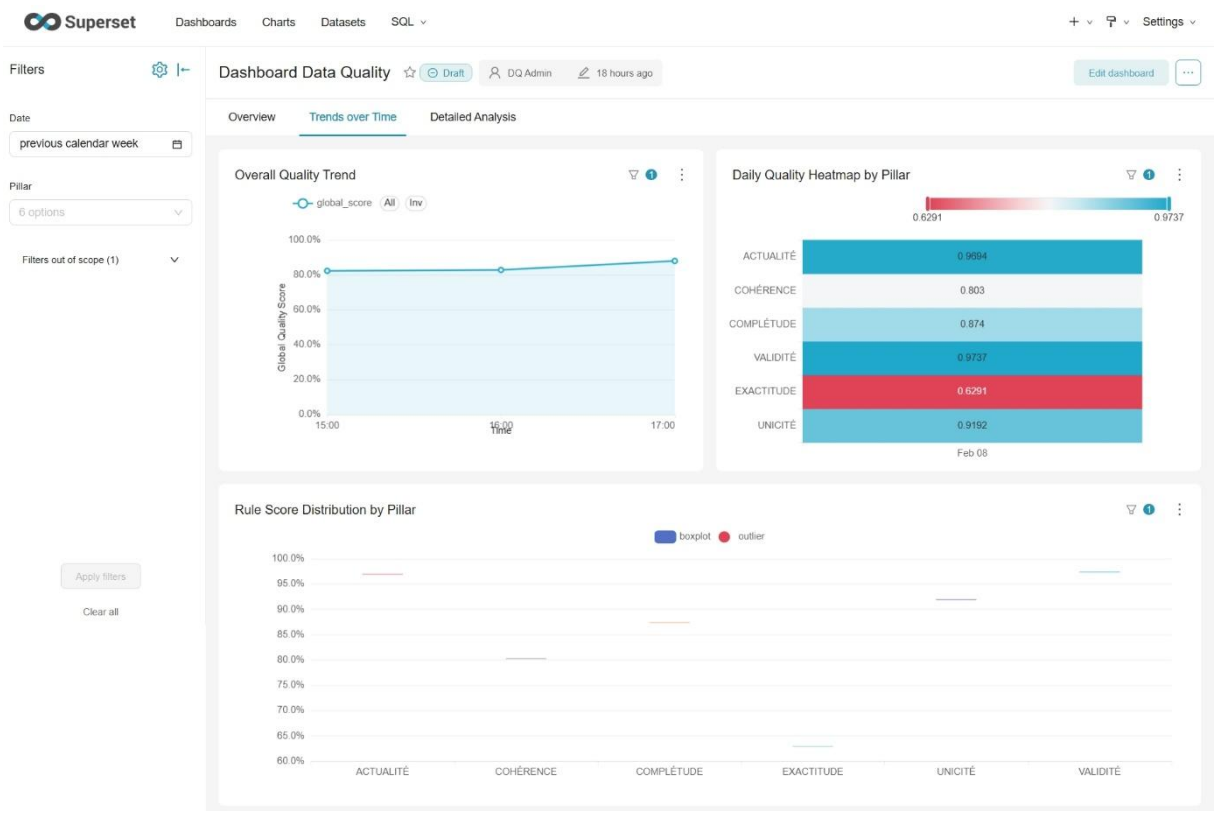
En appliquant un filtre sur la table patients, la jauge et le graphique en barres se recalculent automatiquement pour n'afficher que les scores de qualité de cette table, qui atteignent ici 73,2%.

- Filtre Pillar



Dans l'onglet "Detailed Analysis", lorsqu'on filtre sur le pilier Complétude, la table met en évidence les tables qui concentrent le plus d'erreurs de complétude, tandis que le diagramme de Sankey montre les flux d'erreurs allant de ces tables vers le pilier Complétude.

- Filtre Date



En utilisant le filtre temporel sur la date avec l'option *Previous calendar week*, l'onglet "Trends over Time" se met à jour pour n'afficher que les scores de qualité de la semaine précédente. Les courbes et heatmaps deviendront plus riches et lisibles à mesure que le nombre d'exécutions du pipeline augmentera.

5.3 Interprétations et Analyses

Lecture globale des scores

Sur le dernier run, le score global de qualité est de 79,9%, en-dessous du seuil de 90% jugé satisfaisant. Les piliers Validité, Unicité et Actualité dépassent 90%, ce qui est rassurant, alors que la Complétude (~87%) et surtout la Cohérence (~80%) restent en retrait. Le pilier Exactitude, avec un score d'environ 45%, constitue clairement le point de faiblesse principal.

Exactitude et Cohérence : deux signaux d'alerte

Le score très bas d'Exactitude traduit de nombreux échecs sur les règles de format (téléphone, email, code postal). Métierment, cela compromet la contactabilité des patients et les analyses géographiques, et justifie une exploration détaillée des sources et des contrôles de saisie.

La Cohérence est meilleure mais reste autour de 80%, avec des erreurs fréquentes sur les contrôles âge/date de naissance et dates d'arrivée/départ. Ces incohérences

peuvent fausser des analyses cliniques ou de parcours de soins et doivent être traitées via des validations plus fortes à la source.

Apport du taux d'erreur pondéré

Le graphique de répartition des erreurs par pilier utilise un taux d'erreur pondéré (Σ checks_failed / Σ checks_failed + checks_passed). Il montre que :

- Exactitude combine des règles très difficiles mais aussi beaucoup de contrôles simples et quasi parfaits, ce qui limite son taux d'erreur pondéré à environ 3%, malgré un score moyen faible.
- Cohérence, avec moins de contrôles mais plusieurs règles souvent en échec, dépasse 20% d'erreurs pondérées et occupe une place visuellement plus importante dans ce graphique.

Priorités métier

En pratique, on peut en déduire que :

- Exactitude doit être la priorité n°1 (normalisation des téléphones/emails/CP, règles de saisie, contrôles applicatifs).
- Cohérence doit être rapidement renforcée sur les âges et les dates de séjour.
- Validité, Unicité et Actualité sont à un niveau acceptable (>90%) pour des usages décisionnels courants, tandis que la Complétude nécessite surtout un effort ciblé sur quelques champs clés comme les numéros de téléphone.

6. Limites et pistes d'amélioration

- **Pas d'alerte automatique**
  Aujourd'hui, il faut venir voir le dashboard pour détecter un problème. L'idée serait de déclencher une alerte (mail/Teams) dès que le score global ou un pilier passe sous un seuil (par exemple 90%).
- **Colonnes problématiques pas assez visibles**
  On voit bien les tables et les règles, mais on n'a pas encore un visuel qui montre immédiatement les colonnes les plus en erreur. Un graphique "Top colonnes à problème" (par pilier) aiderait à cibler directement les champs à corriger.
- **Peu de visibilité sur l'origine des erreurs**
  Le dashboard dit *où* sont les erreurs, mais pas *d'où* elles viennent (service clinique, application source…). Ajouter ces informations dans les datasets permettrait de prioriser les actions par service ou par système.