

DESCRIPTIF DES REQUETES

BILANTE

La plus part des requêtes sous mentionnées se servent de deux vues :
hours_per_station_per_line_view.sql et stations_for_correspondances_view.sql

- **Requête de création de la base de données**
(railway_managment_database_creation_script.sql)
Suivant le modèle réalisé
- **Quels trajets reliant deux gares minimisent les correspondances ou le temps d'attente ? :(correspondances.sql)**
A l'aide des informations stockées dans les vues, on fait une jointure qui donnent pour chaque gare où passent plus d'une ligne, les différents horaires
- **Quels trajets alternatifs sont disponibles en cas d'incident sur une ligne spécifique ?(alternative_lines.sql)**
Nous utilisons des CTE tel que Passage_Incident avec les paramètres spécifiques au trajet et avec plusieurs jointures et la vue des horaires on arrive à retrouver pour chaque gare de la ligne quelles autres lignes permettent de les rejoindre
- **Existe-t-il des gares sans trafic sur une période donnée ?(ghosted_stations.sql)**
Cette requête identifie les lignes ferroviaires n'ayant effectué aucun trajet sur une période donnée (du 1er au 31 janvier 2024) et affiche les gares qu'elles desservent. Elle procède en deux étapes : d'abord, elle sélectionne les lignes sans trajets enregistrés sur la période (LignesSansTrajet), puis elle récupère les gares associées à ces lignes via la table Traverse, en les triant par ligne et par gare.
- **Existe-t-il des conflits d'horaires entre des trajets utilisant les mêmes quais dans une gare ?(traffic_assessment.sql)**
Cette requête détecte les gares susceptibles d'être surchargées en analysant les arrivées de trains sur des périodes glissantes de 60 minutes. Elle commence par extraire les heures d'arrivée des trains par gare (Arrivals), puis compte combien de trains arrivent dans l'heure suivant chaque arrivée (CountIntervals). Enfin, elle identifie les gares où le nombre maximal

d'arrivées en une heure dépasse le nombre de quais disponibles, signalant ainsi un risque de congestion.

- **Empêcher l'affectation d'un train en maintenance à un trajet(train_state_trigger.sql)**

Nous avons ce trigger qui empêche l'insertion d'un train dont le status de maintenance est actif d'être assigné à un trajet

- **Actualiser le statut d'un train en maintenance (update_train_status_after_maintenance_function.sql)**

Nous avons ici une fonction qui, combiné à un scheduler permet de passer toutes les entrées de la table Maintenance et pour les trains dont la date de maintenance est passée, on actualise le statut du train concerné dans la table Train. Pour le scheduler, utiliser pgAgent sur Windows et pgCron sur Unix/Linux

Amine

- **Quels trains nécessitent une maintenance basée sur leurs heures cumulées de trajet ou des incidents signalés ? (le fichier MaintenanceNeeded.sql)**

Cette requête SQL permet d'identifier les trains nécessitant une maintenance en fonction de leur temps d'utilisation et du nombre d'incidents rencontrés. Elle commence par une CTE (MaintenanceNeeded) qui calcule pour chaque train le total d'heures d'utilisation (somme des durées des trajets) et le nombre d'incidents (incidents liés à des retards) . Ensuite, la requête principale attribue une priorité aux trains selon ces critères : une maintenance est urgente si le train a accumulé plus de 50 heures d'utilisation ou a subi au moins deux incidents, préventive entre 20 et 50 heures, et non nécessaire en dessous de ce seuil. Enfin, seuls les trains nécessitant une maintenance sont affichés, triés par heures d'utilisation décroissantes. Les nombres d'heures d'utilisation pour les seuils ont été attribués aléatoirement.

- **Quels trajets présentent le plus souvent des retards importants ? (le fichier RetardsImportants.sql)**

Cette requête crée une vue retards_importants qui identifie les trajets ayant un retard moyen supérieur à 30 minutes. Elle commence par regrouper les données

de la table Retard par idTrajet, puis calcule la moyenne des retards pour chaque trajet. Grâce à la clause HAVING, seuls les trajets dont la moyenne dépasse 30 minutes sont conservés. Enfin, une requête sur cette vue permet d'afficher les résultats en triant les trajets du plus retardé au moins retardé, facilitant ainsi l'analyse des trajets les plus impactés par les retards.

- **Quels techniciens interviennent le plus souvent sur les maintenances ?
(le fichier TechnicienMaintenance.sql)**

Cette requête permet d'identifier les techniciens ayant réalisé le plus d'interventions de maintenance. Elle commence par joindre la table Technicien avec la table Maintenance en associant chaque technicien aux maintenances qu'il a effectuées. Ensuite, elle regroupe les résultats par technicien et calcule le nombre total d'interventions effectuées. De plus, elle utilise la fonction ARRAY_AGG(DISTINCT M.idTrain) pour lister les trains sur lesquels chaque technicien est intervenu. Enfin, les techniciens sont triés par nombre d'interventions décroissant, en plaçant ceux n'ayant effectué aucune maintenance à la fin du classement.

- **Lister les gares avec des équipements de restauration
(le fichier restauration.sql)**

Cette requête permet d'identifier les gares disposant d'installations de restauration et de les classer par nombre de restaurants. Elle commence par joindre la table Gare avec Equipement_de_Gare pour récupérer les équipements associés à chaque gare, puis avec la table Equipement afin de filtrer uniquement ceux dont la description est "Restauration". Elle affiche ensuite le nom de la gare ainsi que le nombre de restaurants présents et classe les résultats par ordre décroissant, mettant en avant les gares les mieux équipées en termes de restauration.

- **Quels trains sont les plus utilisés en termes de kilomètres parcourus ?
(le fichier TrainKilometrage.sql)**

Cette requête permet d'identifier les trains les plus utilisés en fonction des kilomètres parcourus. Elle commence par joindre la table Train avec Trajet pour associer chaque train aux trajets qu'il a effectués. Ensuite, elle relie ces trajets aux horaires (Horaire) puis aux lignes (Ligne) afin de récupérer la longueur totale de chaque ligne parcourue (nous supposons que chaque trajet effectué va au bout de la ligne). La fonction SUM(l.longueur_totale) additionne ces distances pour chaque train, et COALESCE(..., 0) garantit qu'un train sans trajet affiche 0 km. Enfin, les résultats sont regroupés par train et triés par distance parcourue décroissante, mettant ainsi en avant les trains les plus sollicités.

- **Analyse Couverture Sanitaire Gares (le fichier CouvertureSanitaire.sql)**

Cette requête permet de connaître le nombre de toilettes disponibles dans chaque gare. Elle commence par sélectionner les gares depuis la table Gare et les joint à la table Equipement_de_Gare pour récupérer les équipements associés. Le filtre eg.idEquipement = 3 s'assure que seuls les équipements correspondant aux toilettes sont pris en compte. La fonction COALESCE(eg.nombre, 0) remplace les valeurs nulles par 0 pour les gares ne disposant pas de toilettes. Enfin, les résultats sont triés par ordre décroissant du nombre de toilettes, mettant en avant les gares les mieux équipées.

Mohamed Zentar

1. Identifier les horaires les plus fréquentés pour chaque ligne(fichier :horaire plus frequent)

- Trouver les horaires qui ont le plus de trajets pour chaque ligne ferroviaire.

Description :

- La requête utilise des jointures entre les tables Horaire et Trajet.
- On regroupe les résultats par idLigne et heure_depart, puis on compte le nombre de trajets.
- Les résultats sont triés pour afficher les horaires les plus fréquentés par ligne.

2. Quelles sont les gares ayant une saturation de trafic selon les heures de pointe(fichier :sous requete Saturation de trafic)

- Identifier les gares avec le plus grand nombre de trains transitant pendant les heures de pointe (7h à 9h).

Description :

- La requête utilise des jointures entre les tables Gare, Traverse, Horaire, et Trajet.
- Une sous-requête extrait les horaires de départ pendant les heures de pointe.
- On compte les trajets associés à chaque gare et on trie les résultats par nombre décroissant.

3. Quel est le taux moyen de retard des trains sur une période donnée(fichier :taux moyen de retard)

- Calculer la durée moyenne des retards pour tous les trajets d'une période donnée.

Description :

- On utilise la table Retard avec la jointure sur Trajet pour filtrer par période.
- La durée moyenne de retard est calculée en utilisant AVG().

4. Quels types d'incidents sont les plus fréquents et où se produisent-ils le plus(fichier :type incident le plus frequent)

- Identifier les types d'incidents les plus fréquents et éventuellement les gares associées.

Description :

- On utilise la table Type_Incident pour récupérer la description des types d'incidents.
- Les incidents sont reliés aux trajets via la table Retard, et aux gares via des jointures supplémentaires.
- La requête compte les occurrences des incidents pour chaque type.

5. Identification des correspondances optimales entre différents trajets :(fichier :correspondance optimales)

Cette requête SQL vise à identifier les correspondances optimales entre différents trajets en calculant le temps d'attente entre l'arrivée d'un premier trajet et le départ d'un second. Elle effectue plusieurs jointures pour relier les trajets à leurs horaires respectifs, en associant d'abord la table Trajet à la table Horaire via l'identifiant idHoraire. Une auto-jointure sur la table Trajet permet ensuite de comparer deux trajets distincts partageant potentiellement une correspondance. La condition WHERE h2.heure_depart > h1.heure_arrivee garantit que le second trajet part après l'arrivée du premier. Enfin, les résultats sont triés en ordre croissant selon l'écart de temps entre les deux trajets et limités aux dix correspondances les plus courtes

Beteille Clément

- **Quels trains peuvent être réaffectés pour couvrir une panne sur une autre ligne?**

(fichier : free_to_use_train.sql)

On sélectionne les identifiants des trains qui ne sont pas présents dans une sous-requête contenant les identifiants des trains :

- Assignés à un trajet après une date précisée
- En maintenance
- **Pas en service (erreur dans la vidéo ; il faut vérifier : where t.en_service = FALSE)**

- **Quels incidents ont le plus d'impact sur la ponctualité du réseau ferroviaire?**

(fichier : impact_of_each_incident.sql)

On sélectionne les identifiants des incidents avec le compte de leurs occurrences dans la table Retard et la somme total de leurs temps de retard. LEFT JOIN de TypeIncident sur Retard permet d'avoir la totalité des types d'incidents ; COALESCE permet d'avoir comme valeur par défaut '00:00:00' pour les types d'incidents sans aucun retard.

On regroupe par les identifiants des types d'incidents, et on met dans l'ordre décroissant de leur nombre de retards puis de leur durée totale de retard.

- **Existe-t-il des incohérences comme un train affecté à un trajet alors qu'il est en maintenance?**

(fichier : used_trains_during_maintenance.sql)

On effectue une jointure entre trajet et maintenance sur l'id du train et on vérifie si la maintenance est prévue le même jour que le trajet

- **Quelle est la durée moyenne des trajets par ligne ferroviaire?**

(fichier : average_length_per_line.sql)

Une jointure entre les tables Trajet et Horaire sur idHoraire, puis une seconde entre les tables Horaire et Ligne sur idLigne permettent d'obtenir pour chaque ligne leurs durée moyenne.

On fait une jointure gauche entre Trajet et Retard sur idTrajet pour ajouter au calcul de la moyenne la durée potentielle d'un retard(COALESCE pour éviter les valeurs null).

On regroupe ensuite par les identifiants de ligne et on met dans l'ordre croissant de leur durée.

- **Quels sont les trains les plus fiables (ceux ayant le moins d'incidents ou de retards)?**

(fichier : total_delay_per_train.sql)

On effectue une jointure gauche entre Train et Trajet sur idTrain, puis entre Trajet et Retard sur idTrajet afin d'obtenir pour chaque train le nombre total de retards et la durée totale de retard (COALESCE pour éviter les valeurs null).

On regroupe par idTrain, et on met dans l'ordre croissant (pour connaître les trains les plus fiables) de leur nombre de retards puis de leur durée totale de retard.

- **Fichier d'insertion**

(fichier : insertion.sql)

Insertion dans les tables des valeurs.