

Outil de Contrôle ODB vs DTime

Outil de Contrôle ODB vs DTime	1
1. Contexte et Objectifs	2
2. Description du projet	3
3. Réalisation	4
4. Résultats	9
5. Conclusion	10

1. Contexte et Objectifs

Dans le cadre de la mise en place d'une nouvelle application de gestion des ressources humaines nommé DTime, de multiples tests ont été effectués sur les différents systèmes présents au sein de Disneyland Paris afin de confirmer la véracité des données reçues dans ces systèmes.

Mon équipe étant en charge de ODB, une base de données dont les sources des machines de ventes, des restaurants mais recevant également les données concernant les heures de travail, il a fallu effectuer des tests permettant de vérifier si les données reçues dans ODB sont similaires aux données envoyées par DTime. Ce contrôle est mis en place car l'interaction entre DTime et ODB n'est pas direct, les données passent par différents systèmes nommés ETL (Extract Transform Load) qui sont des systèmes recevant des données, qui les transforment puis les chargent dans les cibles correspondantes car chaque base de données ne stocke pas les données au même format.

Mon équipe n'est cependant pas en charge des tests à effectuer, elle n'est qu'administrateur de ODB. Les tests sont effectués par les métiers, dans le cas présent, un analyste financier.

Il m'a ainsi été demandé de développer cet outil afin de faciliter le travail de cette personne. De nombreux scénarios étaient à tester. Un scénario de tests se composait d'une date ou d'une période de date et d'un ou de plusieurs employés identifié(s) par un ID.

2. Description du projet

Durée du projet : Le projet a débuté le 14 décembre 2024, et s'est conclu le 6 janvier 2025, soit environ 3 semaines.

Membres de l'équipe :

- Zarath Mougamadou – Financial Planning & Reporting Apprentice
- Virginie Le Renard – Project Leader Financial Systems (Tutrice)
- Remi Auguin – Financial Analyst

Technologies utilisées :

- Excel
- VBA
- PowerQuery
- Teams
- Chat GPT

3. Réalisation

La première étape de ce projet était de comprendre le contexte dans lequel il s'inscrivait ainsi que le besoin du métier. Pour cela, une session de présentation a été organisée par ma tutrice afin de m'expliquer le principe et les fonctionnalités de DTime. A la suite de cela, elle m'a proposé d'effectuer ce projet sur PowerQuery car j'avais auparavant eu l'occasion de prendre en main ce logiciel dans le cadre du développement d'un tableau de bord sur PowerBI.

La seconde étape était pour moi de recueillir plusieurs jeux de données afin de pouvoir tester l'outil durant son développement. N'ayant pas les accès de DTime, ma tutrice s'est chargée de générer des rapports à ma demande.

Exemple de rapport DTime :

[illegible]

De mon côté, je m’occupais de générer des rapports sur une application nommée SAP Business Objects dans lequel se trouvait les données de ODB.

Exemple de rapport BO :

REGULAR détail

Business Center Code	Business Center Description	Date Daily	Actual Regular Hours
		17/11/2024	
		18/11/2024	
		19/11/2024	
		20/11/2024	
		21/11/2024	
		22/11/2024	
		24/11/2024	
		25/11/2024	
		26/11/2024	
		27/11/2024	
		28/11/2024	
		29/11/2024	

La troisième étape était de développer l’outil.

PowerQuery fonctionnant par sources, c’est-à-dire qu’il faut lui charger des données, j’ai tout d’abord créé un fichier Excel comportant un rapport DTime et un rapport BO.

Il m’a fallu retraiter les données afin de pouvoir les comparer. Par défaut, les deux rapports ne possèdent aucune valeur pouvant permettre une correspondance entre les deux rapports. J’ai donc mis en place une clé du format “Date - CCFR” des deux côtés. CCFR correspond aux centres de coûts français. Disney étant une société américaine, il est également utilisé des centres de coûts américains. Il y avait également quelques retraitements à effectuer sur le format des données, notamment des dates car sur BO, les dates sont au format français, c’est-à-dire “JJ/MM/AAAA” alors que sur le rapport DTime, il s’agissait du format anglais “MM-JJ-AAAA”.

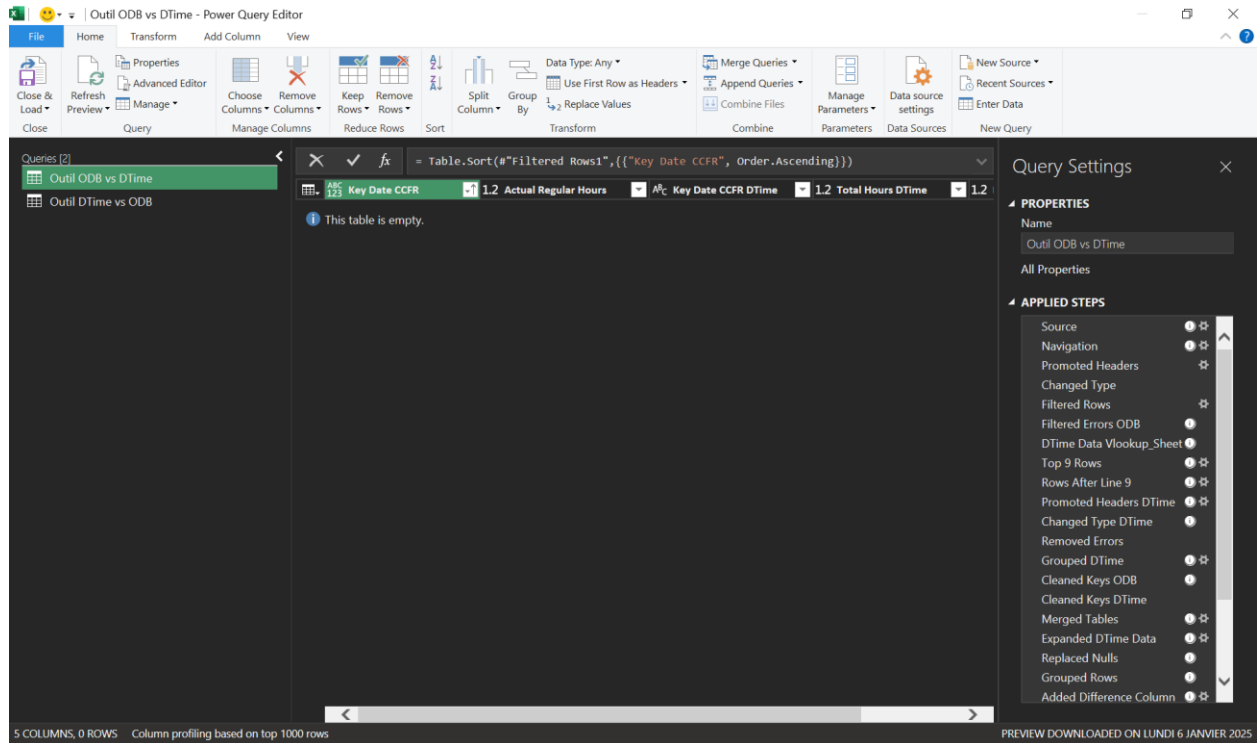
Sur base de ces données, j’ai pu débiter le développement de l’outil sur PowerQuery.

Power Query est un moteur de transformation de données et de préparation des données.

Il est possible d’effectuer un grand nombre de manipulation à travers les menus.

Cependant, Power Query propose la possibilité de coder en utilisant le langage M. Dans le cas de ce projet, je me suis principalement servi de la programmation afin de développer l’outil car il y avait un certain nombre de fonctionnalités avancées indisponibles avec les options des menus, comme la possibilité de créer une colonne dont le résultat dépend d’une autre colonne. En effet, il me fallait extraire une partie des valeurs d’une colonne et selon la valeur extraite, il me fallait retourner une valeur dans une nouvelle colonne.

Interface de PowerQuery :



Fonctionnement de l'outil :

Pour que l'outil fonctionne correctement, il était nécessaire de mettre en place un environnement en local dans la machine. En effet, la source de données étant local, il était essentiel que les fichiers soient au bon endroit car le fonctionnement de l'outil impliquait des chemins de fichiers à respecter.

Il fallait ainsi créer un dossier **Check_DTime_ODB** dans le répertoire **C:** de sa machine. Au sein de ce dernier, il fallait y créer deux dossiers :

- **CastMember**
- **Temporary**

Le dossier **CastMember** traitait les données des employés de l'entreprise.

Le dossier **Temporary** traitait les données des intérimaires, des contrats temporaires etc.

Il a fallu faire une distinction et adapter l'outil car le rapport DTime généré pour les employés et celui pour les autres n'avaient pas le même format. La manipulation des données se basant sur des notions comme le nom des colonnes ou le nombre de lignes, j'ai adapté l'outil afin qu'il traite correctement les données des rapports.

Le principe étant le même, nous allons poursuivre avec le cas du dossier **CastMember**.

- **1 - DTime_ODB_Data_Sources_CM**
- **2 - DTime_ODB_Check_Tool_CM**

Le premier fichier est la source de données. Il comporte les rapports DTime et BO retraités afin d'avoir les données nécessaires à la comparaison.

Le second fichier comporte l'outil.

Voici la procédure qu'il fallait suivre pour utiliser l'outil :

L'utilisateur devait tout d'abord générer les rapports des deux systèmes.

Au sein du fichier **1 - DTime_ODB_Data_Sources_CM** se trouvait un bouton créé en VBA permettant de vider le fichier de toutes les données précédentes.

Une fois le fichier vide, il fallait copier les données des deux rapports dans leurs onglets respectifs.

Ensuite, il fallait passer sur le second fichier : l'outil. Il suffisait de rafraîchir le fichier, et l'outil exposait les différences entre les deux systèmes.

Outil :

DTIME ----> ODB - CHECKS CAST MEMBERS

ODB VS DTime				
ODB SOURCE		DTIME SOURCE		TO CHECK
Key Date CCFR	Actual Regular Hours	Key Date CCFR DTime	Total Hours DTime	Difference

DTime vs ODB				
DTIME SOURCE		ODB SOURCE		TO CHECK
Key Date CCFR	Hours	Key Date CCFR ODB	Total Hours ODB	Difference

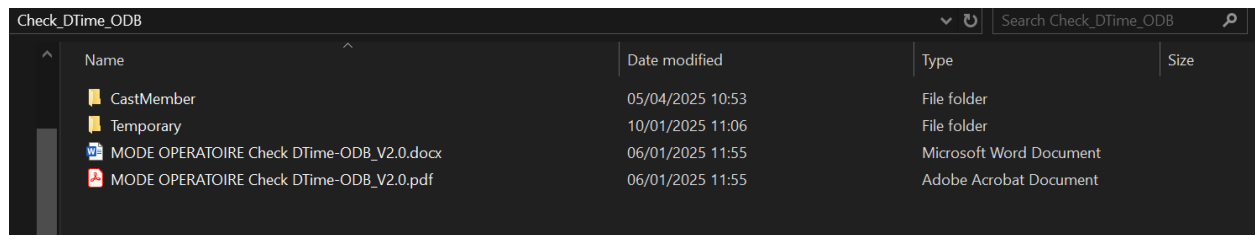
Une fois terminé, une présentation de l’outil a été effectué avec la personne chargée des tests.

Durant les tests, l'utilisateur a rencontré quelques problèmes dû aux modifications apportées dans le rapport DTime. Pour les résoudre, il m’a été envoyé les rapports avec lesquels il a travaillé afin que je puisse tester de mon côté et adapter l’outil en conséquence. De plus, il m’a été demandé de modifier le format des valeurs sur le tableau de l’outil afin que les calculs soient alignés avec le format des rapports puis d’ajouter de la mise en forme afin de mettre en surbrillance les écarts.

Mise à part cela, l’outil a très bien fonctionné dû aux multiples tests effectués de mon côté avant soumission du projet.

4. Résultats

Le livrable proposé au métier comportait un dossier **Check_DTime_ODB**, dans lequel se trouvait un dossier **CastMember** et un dossier **Temporary** dans lesquels se trouvent le fichier source dans lequel placer les données des rapports DTime et ODB et le fichier dans lequel se trouve l'outil, ainsi qu'un mode opératoire expliquant comment utiliser l'outil, étape par étape.



Name	Date modified	Type	Size
CastMember	05/04/2025 10:53	File folder	
Temporary	10/01/2025 11:06	File folder	
MODE OPERATOIRE Check DTime-ODB_V2.0.docx	06/01/2025 11:55	Microsoft Word Document	
MODE OPERATOIRE Check DTime-ODB_V2.0.pdf	06/01/2025 11:55	Adobe Acrobat Document	

Grâce à cet outil, un gain de temps conséquent s'est opéré permettant d'effectuer l'intégralité des tests demandés avant la fin de la période de test. De plus, dû à la simplicité d'utilisation de l'outil, l'analyste financier a pu effectuer les tests sans grandes difficultés. Il suffisait d'extraire les deux rapports, les copier dans le fichier source puis rafraîchir le fichier de l'outil et le test était terminé.

5. Conclusion

En conclusion, le développement de cet outil de comparaison des données entre le système DTime et le système ODB a permis de simplifier la réalisation des tests demandés ainsi que de gagner beaucoup de temps. La personne chargée des tests fût satisfaite de la simplicité et de l'efficacité de l'outil qui lui permettait en quelques minutes d'exposer les éventuels écarts mais surtout d'effectuer l'intégralité des tests avant la fin de la période autorisée. De plus, cet outil a permis à ma tutrice chargée de suivre l'avancée des tests d'avoir un support lors des réunions dans lesquelles il lui était possible de présenter le tableau final dans lequel se trouvait les différences, ce qui simplifiait la compréhension du problème pour les autres acteurs du projet et apportait de la clarté.

Lors du développement du projet, je me suis rendu compte qu'il aurait été préférable de passer par une solution cloud. En effet, plutôt que de créer un environnement dans sa machine, en local, l'utilisation d'un espace de stockage en ligne comme SharePoint aurait pu simplifier la mise en place de l'outil pour l'utilisateur.

Mode opératoire :



MODE OPERATOIRE
Check DTime-ODB_\