Ejercicio Final de Master 2022

Alex Ayuso Herranz
Vanesa Cayetano Alfaro
Aitor Llanos Irazola

accenture

índice

Copyright © 2022 Accenture. All rights reserved.

Memorias de como se ha montado

1. Primero nos reunimos para leer el enunciado del ejercicio, descargar los datasets necesarios y creación del repositorio GitHub.
2. Cada uno diseño una arquitectura para después ponerla en común y elegir cuales herramientas íbamos a usar.
3. En un principio, empezamos usando Azure Databricks pero finalmente usamos Azure Syanpse en su lugar.
4. Se creo un Data Lake Gen 2 donde hemos almacenado 2 datasets en formato .tsv que se nos ha proporcionado en el enunciado del ejercicio, y 2 datasets en formato .csv que nos hemos descargado para completar los datos que necesitábamos.
5. Se creo Azure Synapse para el proceso de ETLs usando Spark y nos reunimos para saber que tablas necesitamos crear.
6. Cada tabla se almaceno en Hive para su posterior uso.
7. En un principio, empezamos usando Mysql pero debido a que fallaba la conexión se cambió por SQL database.
8. Se han creado las distintas tablas en SQL Database
9. Se ha usado Power Bi para la visualización de las tablas y gráficos que pedía el cliente en el ejercicio usando un patrón DAR (Dashboard / Analysis / Reporting).
10. Finalmente, para mejorar la solución se intento la primera aproximación de automatización, creando una ingesta de datos con las pipelines de Synapse, para, en un futuro, poder crear una pipeline que realice todo el trabajo de transformación de forma automática.

Copyright © 2022 Accenture. All rights reserved.

Arquitectura por partes

- ## ¿Por qué Synapse?

  Inicialmente, escogimos Databricks, por ser la herramienta en la que se hizo la práctica en Spark y en la que más tiempo pasamos. Pero aparte de los problemas de integración que nos daba con otras herramientas de almacenamiento decidimos que no era la herramienta correcta. Posteriormente, escogimos Azure Synapse, que es el lugar donde realizamos todo el procesamiento, por tener una gran cantidad de herramientas integradas, Data Lake Gen 2, pipelines, monitor...
  Además, nos era relativamente similar también debido a que tuvimos una práctica relacionada con sus características, y se podía hacer la ejecución de Spark sobre los datos, algo que considerábamos imprescindible, para realizar las transformaciones de los datos.
  Finalmente, otra funcionalidad interesante fue la integración con GitHub que nos permitió mantener un control de versiones en el desarrollo del proyecto.

- ## Ingesta de datos mediante pipelines

  Se realizo una ingesta de datos, con Azure Synapse Pipelines.
  Se obtienen los datos de un Azure Blob Storage, dónde están almacenados, y se copian desde ese dispositivo de almacenamiento al Azure Data Lake Gen 2 que se utiliza posteriormente en la transformación de datos.
  Esta ingesta está planificada para realizarse todos los días a las 10 de la mañana por si se capturasen datos nuevos.

- ## ETLs de transformación de datos

  La transformación de datos se planifico inicialmente para realizarla con Spark. Se ha realizado en los distintos Notebooks en un Apache Spark pool para la creación de las tablas con los datos necesarios para posteriormente ser consumidos en la fase de explotación.

- ## Hive

  Se han realizado guardado de las tablas con spark en formato Hive y almacenadas en el Data Lake Gen 2, y como fichero en formato .csv

- ## SQL

  Se han creado distintas tablas con todos los datos necesarios para su posible consumición en una base de datos SQL Server.

- ## Visualización con Power BI

  Se ha seguido el patrón DAR, para la creación de un Dashboard, Analysis y Reporting para obtener toda la información pedida por el cliente. Se han creado diversas métricas para llevarlas a cabo.

Copyright © 2022 Accenture. All rights reserved.

Problemas que han surgido y su solución

1. Selección de estilo de ejecución, dudando si elegir Databricks por familiaridad al utilizarlo en las prácticas para ejecutar Spark, hacerlo en local como sugería el enunciado u otro elemento.

   SOLUCIÓN: usamos Azure Synapse

2. Al crear Synapse no podíamos darnos permisos, por lo que no podíamos ver los Notebooks de los compañeros y vincular nuestra rama de GitHub a Synapse.

   SOLUCIÓN: contactamos con Sergio para que nos diera los permisos necesarios.

3. Conexión con base de datos Mysql database, inicialmente MySQL desde Apache Spark, ya que fallaba la conexión.

   SOLUCIÓN: Cambiar la base de datos creada a una SQL database.

4. Formateo de las fechas para saber si una fecha de registro está en un intervalo temporal del horóscopo u otro.

   SOLUCIÓN: Definición de una función para determinar el signo del horóscopo según una fecha dada. La función recibe una fecha en formato String y realizando las transformaciones correspondientes de formato, consulta el dataset de horóscopos y devuelve el signo al que pertenece. Esta función fue aplicada al rdd obtenido de nuestro dataframe.

5. La carga de datos de la tabla “userGroups” era tan grande que la base de datos no podía almacenar todos los datos ya que tardaba tanto que la conexión acababa fallando.

   SOLUCIÓN: aumentar los núcleos de la base de datos porque faltaba capacidad computacional para guardar tanta cantidad de datos.

Copyright © 2022 Accenture. All rights reserved.

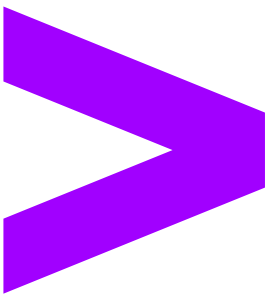
Mejoras o propuestas

- **Automatización**: automatización de la ingesta y ejecución de los procesos automáticamente tras esta para dejar los datos filtrados en las tablas debidas, SQL y HIVE.

- **Costes**: realizar un mejor control de costes de los recursos en la nube ya que pueden acabar siendo voluminosos, y se pueden hacer un control sobre ellos resultando en un menor presupuesto del proyecto

- **Recursos** mejor utilizados: existen recursos que pueden ser aprovisionados de mejor forma, acorde al trabajo realizado, ya porque estén infrautilizados o sobre utilizados y se pueden escalar.

Copyright © 2022 Accenture. All rights reserved.



Copyright © 2022 Accenture
All rights reserved.
Accenture and its logo are trademarks of Accenture.