**Proyecto 2: Infraestructura Visible**

F. Bedoya, C. Salinas, N. Orjuela

**Grupo #20**

**Sección #1**

**Inteligencia de Negocios**

**Departamento de Ingeniería de Sistemas y Computación**

**Universidad de Los Andes**

**Bogotá, Colombia**

1. **Requerimientos analíticos**

Estos requerimientos analíticos fueron escogidos porque

1. **Modelado de Data Marts**
   1. **Modelos dimensionales**

**Diagrama, Esquemático

Descripción generada automáticamente**

Al analizar los datos que se encuentran en los archivos .xslx nos dimos cuenta que estos tienen la misma estructura, por lo que proponemos este modelo para utilizarlo en las dimensiones escogidas, específicamente aquellas que vamos a implementar para este proyecto, es decir, la dimensión de educación.

* 1. **Justificación de los modelos dimensionales**

La granularidad de la tabla de hechos se establece mediante sus dimensiones. Es importante entender el indicador geográficamente y por eso se incluyen las dimensiones de departamento y entidad (mejor entendida como municipio) y la fecha en la que este indicador se tomó. La fecha tiene una granularidad bastante gruesa, solo contando con año y mes, pero es importante considerar que para indicadores de tan grande aspecto y poca variabilidad diaria o semanal, los reportes son comúnmente mensuales y los cambios que se observan se reportan mensual o anualmente.

El hecho se compone del nombre del indicador, la subcategoría del indicador para ayudar a filtrar indicadores parecidos, el resultado numero o cualitativo reportado por el indicador, la unidad del resultado y la fuente de la que fue extraído. De aquí no hay ninguna medida aditiva, además de las que son cualitativas, el único valor numérico es el Dato Numérico para el cual en su gran mayoría son índices o promedios que no tendrían sentido sumar especialmente a través de múltiples localizaciones geográficas.

Es importante recalcar que el manejo de la historia del hecho, necesario para ver la variabilidad anual de los indicadores se maneja en el hecho mismo a través de la dimensión fecha y no se utiliza ninguno de los tipos de manejo histórico desde las dimensiones. Esto se hizo porque ninguna dimensión representa una potencial variabilidad y el nuevo registro o cambio de un indicador se debe ver reflejado sobre sus resultados y fecha.

1. **Entendimiento de los datos, creación del Data Mart y proceso ETL**
   1. **Entendimiento de los datos**

Los datos están muy limpios y son muy trabajables. Lo que mas nos gustaría recalcar es que son consistentes. Un código de departamento o de entidad se mantiene para el mismo departamento o entidad a través de todos los datos. Aun así, son muchos datos y esto tiene un efecto negativo en la infraestructura utilizada para el despliegue. Los datos no presentan ningún nulo por fuera de lo esperado para el negocio, es decir un valor nulo en alguno de los resultados del indicador ya sea cualitativo o cuantitativo por su naturaleza mutuamente exclusiva. De los casi 200,000 datos (de la fuente de datos de educación) que se tienen todos son completos y se rigen con un tipo de dato constante, en su mayoría un String, exceptuando las fechas, códigos y resultados que son datos numéricos.

* 1. **Diseño e implementación del proceso de ETL**

Para la implementación del modelo ETL se obtienen los datos de manera dinámica desde la fuente de TerriData utilizando un Task dentro de Airflow y se descomprime en la máquina de virtual de Docker. Después, otro Task que utiliza el archivo obtenido para generar los CSV de todas las dimensiones y de la tabla de hechos. Es en este mismo Task donde partiendo el archivo principal se manejan los nulos obtenidos y los tipos de datos de las columnas.

Finalmente en un Task por dimensión y tabla de hechos se insertan a la base de datos PostgreSQL.

Graphical user interface, text, application

Description automatically generated

Aquí podemos ver como los Tasks consecutivos se van ejecutando y como la población de las tablas se realiza de manera correcta. Aun así podemos ver un error del DAG cuando se incluye el Task de poblar la tabla de hechos debido a que toma mucho tiempo en la importación (aunque corra después sin errores). Esto hace que los Workers de Airflow tengan errores de sincronización. Para resolver esta problemática se intento cambiar la variable de ambiente que indica el Timeout del DagBag Import pero no se pudo lograr una solución estable en el contenedor de Docker.

1. **Propuesta de una arquitectura de solución**
   1. **Propuesta de una arquitectura de solución**

Queremos mencionar que algunas de las limitaciones que se dieron en el proyecto se debieron a la maquina virtual que se provee y la falta de almacenamiento y memoria principal que esta trae. Debido a esto se recomienda un despliegue en la nube en Microsoft Azure por medio de un contenedor y de su provisión de una base de datos PostgreSQL.

En temas de procesamiento, lo más pertinente en el futuro puede ser usar PySpark, obteniendo los datos dinámicamente a un Bucket S3 o Amazon Redshift y procesando con el mencionado PySpark en Amazon EMR. Esto, aunque posiblemente tenga un costo elevado, ayudaría a reducir problemas de tiempo y falta de recursos y seria una infraestructura extremadamente robusta. A partir de aquí se pueden consumir los datos usando PowerBI conectado a una base de datos remota en la nube.

* 1. **Implementación de los tableros de control**

A

1. **Video**

El video se encuentra en este enlace

1. **Actividades realizadas**

* **Redacción de requerimientos analíticos:** Felipe Bedoya, Camilo Salinas, Nicolás Orjuela
* **Modelado de Data Marts:** Felipe Bedoya, Nicolás Orjuela
* **Entendimiento de las fuentes de datos:** Felipe Bedoya
* **Diseño e implementación del proceso de ETL:** Felipe Bedoya
* **Implementación de los tableros de control:** Camilo Salinas, Nicolás Orjuela

Repartición de puntos (100):

* Felipe Bedoya: 33,3 puntos
* Camilos Salinas: 33,3 puntos
* Nicolás Orjuela: 33,3 puntos