Problema 1)

Un cliente reporta problemas de performance (demoras excesivas) en una funcionalidad de una aplicación Web de mediano porte, de una empresa financiera encargada de manejar movimientos bancarios de clientes previamente registrados.

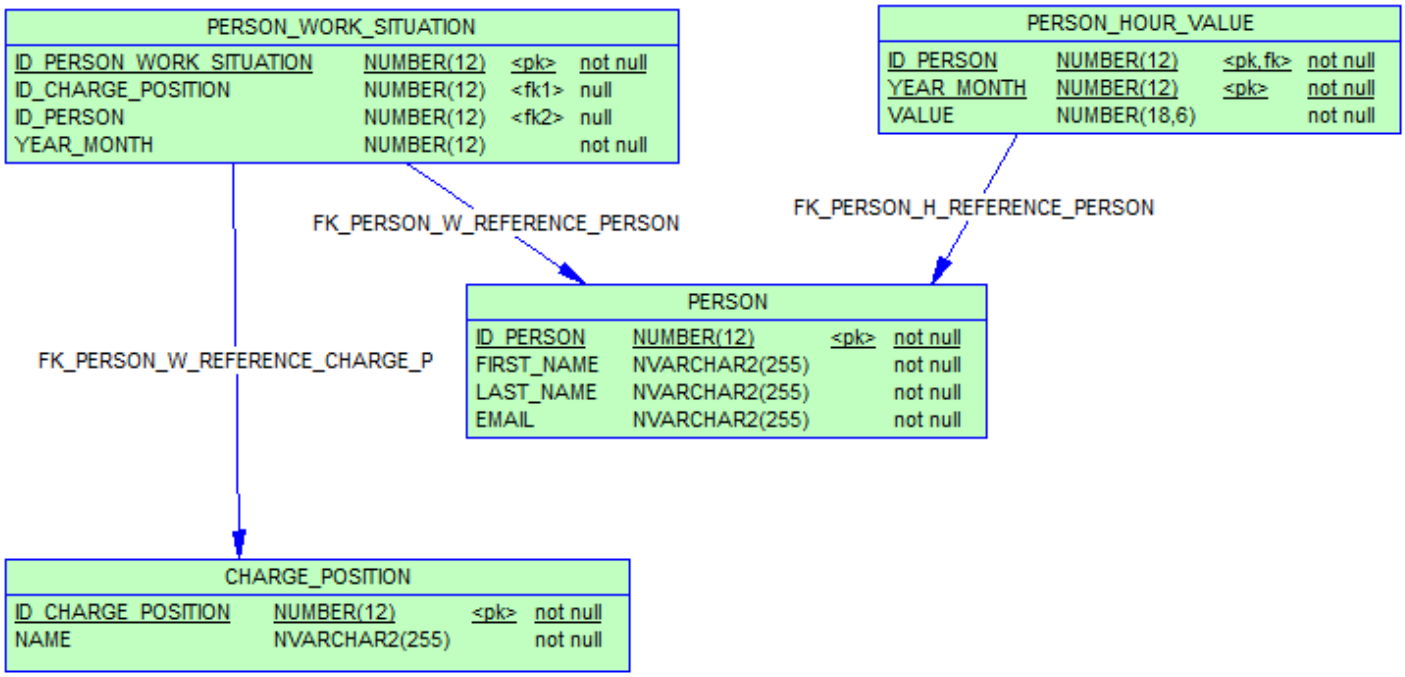
Dicha funcionalidad consta del ingreso de ciertos datos de identidad de un cliente ya registrado, para luego retornar información vinculada a todas sus cuentas con todos sus movimientos. En resumen, una funcionalidad que requiere de gran procesamiento a nivel de datos y aritmética.

Le asignan a usted y a su equipo, el deber de encontrar el motivo de dicho problema. Indique qué haría en esta situación teniendo en cuenta que dispone de todo tipo de permisos para trabajar con la aplicación.
Plantee todas sus líneas de razonamiento y sobre todo las preguntas que haría, a modo ejemplo:

'Si dispongo de log, haría tal cosa…'
'Dispongo de acceso a la Base de datos de producción, por lo que…'

Problema 2)

Un Instituto de Estadística lleva registro de toda la situación laboral de las personas del país teniendo en cuenta no únicamente su trabajo actual, sino sus distintos trabajos a nivel del tiempo. A nivel mensual para ser exacto.
Para lo descrito, se tiene el siguiente modelo de Base de datos:



- PERSON
  Tabla en donde se almacena información básica de las personas del país.
- PERSON_WORK_SITUATION
  Tabla en donde se especifica la posición que tiene la persona en un mes determinado.
- CHARGE_POSITION
  Tabla de posiciones.
- PERSON_HOUR_VALUE
  Tabla que especifica el valor hora de una persona en un mes determinado. Siendo el valor hora, el sueldo que cobra la persona, correspondiente a una hora de trabajo.

Se pide:

1. Escribir el script de creación del modelo de Base de Datos (Puede usarse Pseudocódigo)
2. Escribir en SQL las siguientes consultas:
   a. Obtener la cantidad de personas que trabajaron en el cargo ‘Software Engineer’ durante el mes agosto de 2016.
   b. Obtener la suma de todos los valores hora, de aquellas personas que trabajaron en el cargo de ‘Software Engineer’ entre los meses de Marzo de 2017 y Diciembre de 2017 inclusive.

Notas:

- La columna YEAR_MONTH representa el mes con un número de la siguiente manera: YYYYMM en donde YYYY indica año y MM indica mes. Por ejemplo, Enero de 1998 se representaría como 199801
- Tener en cuenta que pueden existir tablas con cantidad de registros en el orden de los 100.000.000 de registros o incluso más.

Problema 3)

Un Sistema procesa línea a línea un archivo de entrada de gran tamaño (En el orden de 1GB).
Nuestro Sistema toma el archivo del *File System* y comienza a procesar cada línea de manera NO sincrónica (en paralelo).

A medida que se va procesando el archivo, cada línea ya procesada llega como mensaje a una operación llamada *processedLine(line : Line),* ubicada en la clase *LineProcessed.*

Se le ha encomendado escribir un nuevo archivo llamado "*FileDone.txt*", en la carpeta "*DestinationFolder*/" con las líneas ya procesadas, es decir, las líneas que llegan al método *processedLine.*

Dado que el procesamiento inicial se hace de manera asíncrona, las líneas que llegan a la operación *processedLine* no llegan en el orden inicial del archivo origen. Usted se debe asegurar que el archivo de salida respete el orden del archivo de entrada.

Usted cuenta con las siguientes operaciones:

Clase Line

- getLineNumber(): long
  Indica el número de línea al que corresponde el mensaje en el archivo de entrada.
- lastLineOnFile(): boolean
  Indica si el mensaje es la última línea del archivo de entrada.
- getLineMessage(): String
  Retorna el mensaje que debe ser escrito en el archivo de salida.

Otras:

- createFile(destFolder: String, nameFile: String): File
  Crea y retorna un archivo con destino destFolder y con nombre nameFile. Dicho archivoqueda disponible para su uso en posteriores llamadas al método *processedLine*
- writeLineMessage(message: String, file: File)
  Escribe en el archivo file, el mensaje indicado en message.
- closeFile(file: File)
  Cierra el archivo.
- isFirstProcessingLine(): Boolean
  Indica si es la primera vez que está procesando una línea de un archivo de entrada.

Se pide:
Escribir el método *processedLine(line : Line)* encargado de escribir el nuevo archivo y respetando el orden original.

Notas:

- El método processedLine no tiene problemas de concurrencia, dado que se llama secuencial.
- Usted va a procesar **un único** archivo.
- Puede agregar los atributos, métodos o clases que crea conveniente y en caso de utilizar alguna estructura defínala.