**Analítica de Grandes Datos**

**Departamento de Ciencias de la Computación y la Decisión**

**Facultad de Minas**

**Universidad Nacional de Colombia**

Trabajo Nociones de Arquitectura de la Información

Versión: 2021.04.24 15:17

*Observación: Cada vez que agregue nuevos elementos al documento, o que modifique algún componente del informe, revise la coherencia y consistencia con los otros elementos que hacen parte del mismo.*

**Responsables**

|  |
| --- |
| Nombre Completo – Documento de Identificación |
| 1. Gina Marcela Palacios Vivas - 1129364799 |
| 2. César Augusto Ardila Franco - 1144098500 |
| 3. Camilo Ernesto Ardila Franco - 1017196076 |
| **REPO EN GITHUB: https://github.com/Ceardilaf/Trabajo-AGD-Arquitectura.git** |

*Realiza este trabajo considerando los datos que generan los sistemas transaccionales e información no estructurada de tu dominio (si trabajas por ejemplo para TCC tu dominio es la mensajería; también puedes explorar en la página* [*https://www.kaggle.com/datasets*](https://www.kaggle.com/datasets) *o* [*https://arxiv.org/*](https://arxiv.org/)*). Considera tener acceso a esta información, de al menos 10 MB (puede ser uno o varios archivos de texto), y* ***tener al menos cuatro clases conceptuales. Este documento también debe almacenarse en el REPO. Plazo Máximo de Entrega 23 de Mayo, NO SE recibirá por correo electrónico, envío por*** [***https://forms.gle/h7ty3yZykaUq5m7y6***](https://forms.gle/h7ty3yZykaUq5m7y6)

1. **Comprensión del negocio**
   1. **Descripción del contexto del negocio.**

*Describa en máximo 250 palabras el contexto en el cual se generan los datos y cuál es el proceso que los genera.*

**RESPUESTA:**

Estos datos fueron recopilados en dos plantas de energía solar independientes en la India durante un período de 34 días. Cada planta tiene un conjunto de datos de generación de energía y un conjunto de datos de lecturas de sensores ambientales. Los conjuntos de datos de generación de energía se recopilan a nivel del inversor, dispositivo necesario para convertir la Corriente Directa (DC) en alterna (AC), y a su vez cada inversor tiene varias líneas de paneles solares conectados.

En la planta 1 se cuenta con un total de 21 inversores y en la planta 2 con un total de 22 inversores, cada uno identificado con un único ID. En cada uno de los inversores se toman datos de corriente y voltaje DC a la entrada y de corriente y voltaje AC a la salida, permitiendo calcular de manera indirecta la potencia DC y la potencia AC que está ingresando y saliendo del inversor respectivamente. Los datos generados se registran cada 15 minutos.

Cada planta cuenta con una sola estación meteorológica que recopila datos de temperatura ambiente, temperatura de los paneles y radiación solar en los mismos tiempos que se muestrean los datos de potencia, esto evidencia la manera en la que las variables meteorológicas afectan la generación eléctrica de las plantas.

* 1. **Identificación del problema:**

*Delimite en máximo 150 palabras la problemática, así como identificar los requisitos, supuestos, restricciones y beneficios de la solución de este.*

**RESPUESTA:**

Algunos de los principales factores que afectan la generación de energía en una planta solar de manera negativa son las altas temperaturas de las celdas solares y la suciedad de los paneles, lo cual no permite que la radiación se aproveche adecuadamente y disminuya la eficiencia de la planta.

Adicionalmente, la generación de la planta puede disminuir debido a otros factores como baja radiación o daños en los equipos, razón por la cual es necesario conocer las variables meteorológicas y el estado de los inversores con el fin de poder diferenciar si el motivo de la pérdida de eficiencia es suciedad en los paneles u otro factor.

Analizar estas variables nos ayudaría a definir cuándo la eficiencia de la granja se ve afectada por falta de limpieza de las celdas o por fallas en los equipos utilizados, y así sugerir un calendario de limpieza para maximizar la energía producida.

* 1. **Determinación de objetivos:**

*Describa en máximo 150 palabras las metas a lograr al proponer una solución basada en un modelo de datos o de analítica (cómo y qué tipo de ventaja competitiva se ganará).*

**RESPUESTA:**

El principal objetivo es determinar, con base en las mediciones disponibles, cuándo se debió realizar el lavado de los paneles con el fin de garantizar un funcionamiento eficiente de la planta.

Para esto, se pretende implementar una solución de analítica que permita identificar cómo va disminuyendo la eficiencia de la planta con el paso de los días mediante la comparación de la radiación y la potencia AC generada.

Si se asume un costo equivalente del lavado de la granja en energía, esto permitiría estimar cuándo las pérdidas de generación por suciedad justifican el proceso de lavado de la planta desde un punto de vista financiero. Esto presentaría una ventaja respecto a un cronograma de lavado fijo ya que se reducirían las pérdidas por suciedad y los costos de lavado.

* 1. **Evaluación de la situación actual:**

*Describa en máximo 150 palabras el estado actual antes de implementar la solución de analítica, a fin de tener un punto de comparación que permita medir el grado de éxito de la solución.*

**RESPUESTA:**

Actualmente, muchas de las grandes empresas que se dedican a la operación de granjas solares hacen uso de cronogramas de limpieza fijos que contemplan alrededor de 4 lavados al año dependiendo de las condiciones de suciedad y lluvia de la zona. Sin embargo, es posible que dichas condiciones cambien con el paso del tiempo haciendo que la frecuencia de los lavados pueda ser diferente a lo considerado inicialmente.

En caso de que menos lavados sean necesarios, de seguir con un calendario fijo, se estaría incurriendo en más costos de limpieza de los requeridos. Por otro lado, si las condiciones de suciedad aumentan, se haría necesario aumentar la frecuencia de los lavados con el fin mantener las pérdidas por suciedad dentro de los límites establecidos por la compañía.

* 1. **recolección de datos**

*Describa en máximo 150 palabras los datos a utilizar identificando las fuentes, las técnicas empleadas en su recolección, los problemas encontrados en su obtención y la forma como se resolvieron los mismos. Además, adjunte los datos (archivos de texto, etc.) agréguelos en el github (REPO EN GITHUB) en un solo archivo, por favor comprímalo(s).*

**RESPUESTA:**

Los datos necesarios para llevar a cabo esta actividad fueron tomados de la página: <https://www.kaggle.com/anikannal/solar-power-generation-data?select=Plant_1_Generation_Data.csv>.

Una parte de estos datos (Potencia AC y DC) son recopilados en campo midiendo variables eléctricas directamente sobre cada inversor, mientras que los datos de temperatura ambiente, temperatura del panel y radiación solar son tomados por un único sensor que se instala en un panel de prueba y se asume que todos los paneles de la planta están sometidos a las mismas condiciones climatológicas.

Aunque durante la recolección de los datos se pueden presentar problemas de comunicación, daño de los sensores, entre otros, de la fuente de la que fue tomada la información se garantiza una integridad de los datos del 100%.

* 1. **Descripción de datos (diccionario):**

***Diligencia la siguiente tabla, puede agregar otra columna si lo considera necesario.***

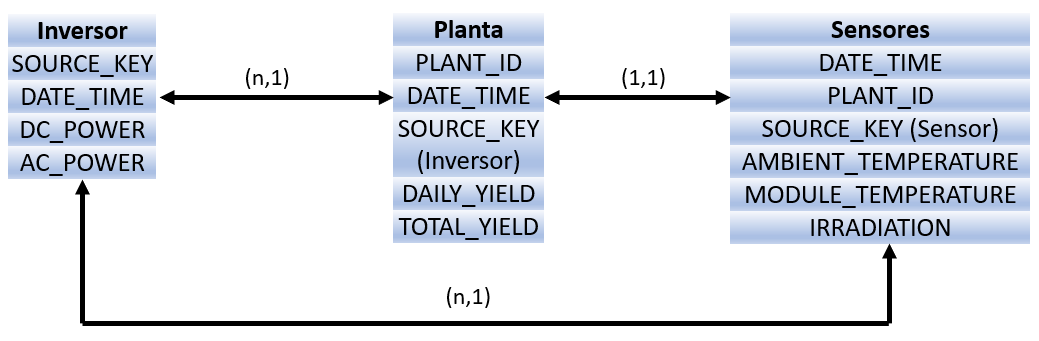
**RESPUESTA:**

|  |  |  |
| --- | --- | --- |
| Nombre del atributo / variable | Formato o Tipo de Dato | Descripción |
| DATE\_TIME | Fecha | Fecha y hora en la que se tomaron las muestras de cada variable. |
| PLANT\_ID | Entero | ID de la planta de generación solar. |
| SOURCE\_KEY | String | ID del inversor / ID del sensor. |
| DC\_POWER | Double | Potencia DC de cada inversor |
| AC\_POWER | Double | Potencia AC de cada inversor |
| DAILY\_YIELD | Double | Generación acumulada (diaria) |
| TOTAL\_YIELD | Double | Generación acumulada (anual) |
| AMBIENT\_TEMPERATURE | Double | Temperatura ambiente de la planta. |
| MODULE\_TEMPERATURE | Double | Temperatura de las celdas. |
| IRRADIATION | Double | Radiación solar. |

* 1. **Modelo del dominIo**

***Observación****: Incluya el gráfico del modelo del dominio que representa la estructura de datos de su problema.*

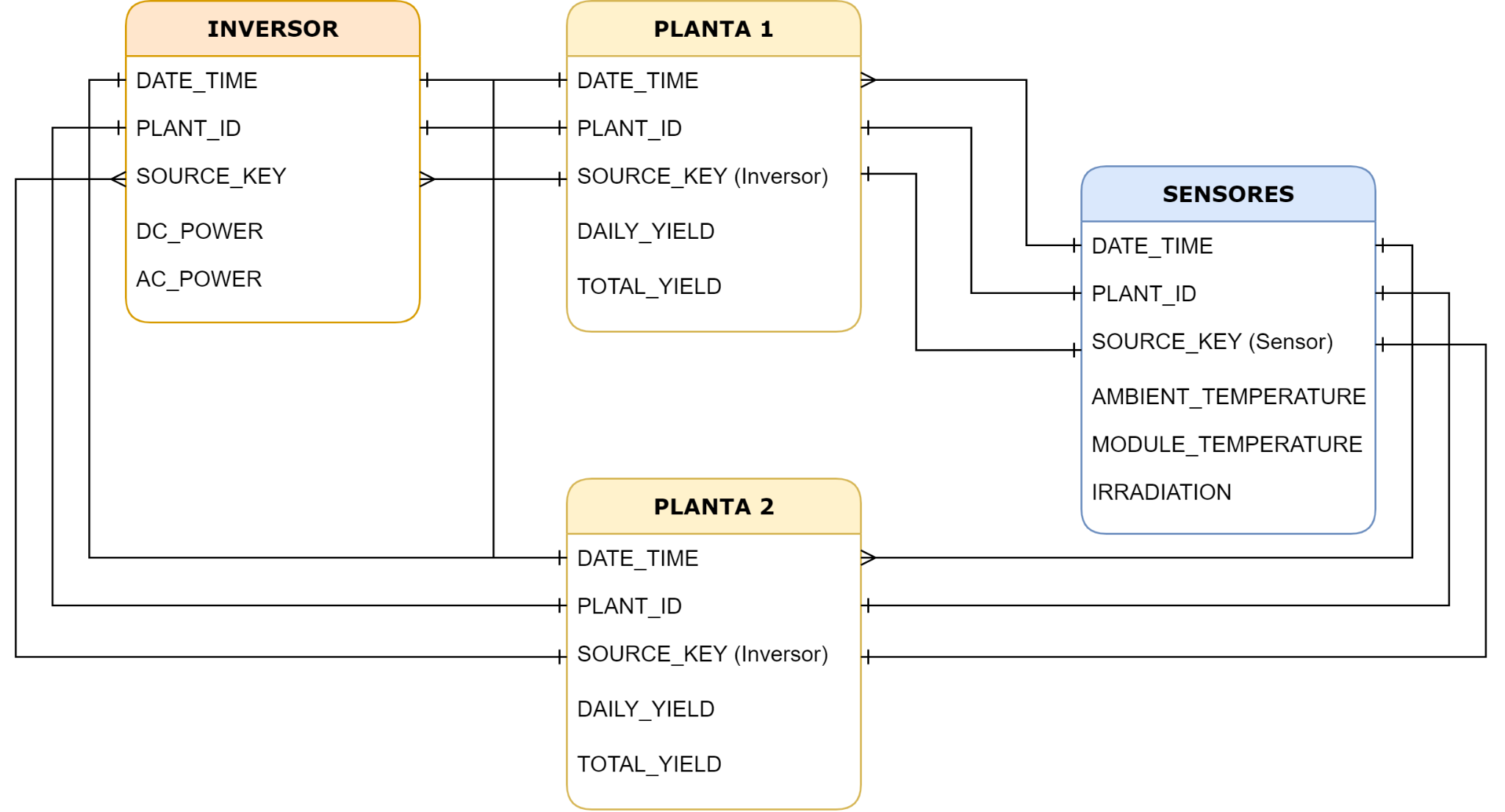
**RESPUESTA:**



1. **Modelo Entidad-Relación**
   1. **Toma de pantalla del modelo E-R**

***Observación****: lo que se pide, puede usar https://draw.io o Microsoft Visio® y modele usando la notación de Barker.*

**RESPUESTA:**

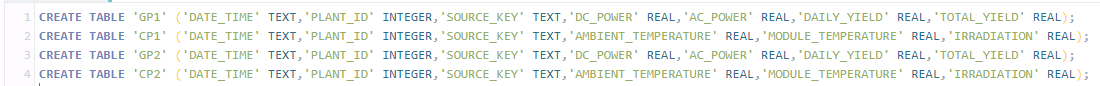
**

* 1. **Sentencia o consulta de creación del tabla(s)**

***Observación****: Escriba el código en el Sistema de Gestión de Bases de Datos Relacionales de su elección (se recomienda SQLite por simplicidad, mediante https://sqlitebrowser.org/) para crear las tablas que corresponda con su conjunto de datos específico. Almacene en el repositorio* **(REPO EN GITHUB)** *el script con el nombre de T1.3.2.Creacion\_Tablas.sql*

***Se recomienda repasar SQL en*** [***https://www.w3schools.com/sql/default.asp***](https://www.w3schools.com/sql/default.asp)

**RESPUESTA:**



* 1. **Sentencias para Insertar datos**

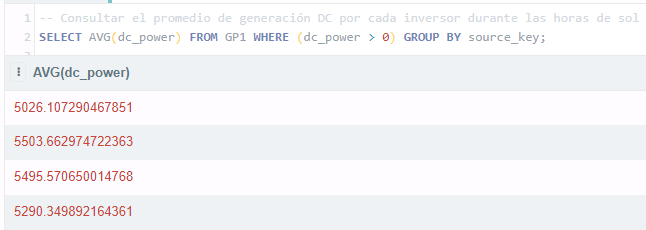
***Observación****: Escriba el código para insertar los datos en cada una de las tablas creadas. Almacene en el repositorio* **(REPO EN GITHUB)** *el script con el nombre de T1.3.3.Insertar\_Datos.sql*

**RESPUESTA:**

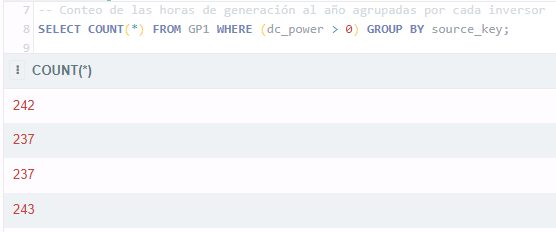
* 1. **Sentencia de consulta**

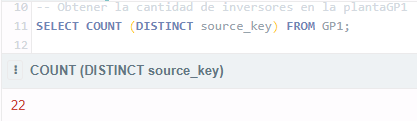
***Observación****: realice la exploración básica de los datos, conteos totales y por categorías, máximos, promedio y mínimos.* Es decir, aplique estadística descriptiva con el fin de conocer las propiedades de los datos y entenderlos lo mejor posible. Use solamente sentencias SQL. Anexe las tomas de pantalla donde evidencie la sentencia SQL y su correspondiente ejecución. Además, *Almacene en el repositorio* **(REPO EN GITHUB)** *el script con el nombre de T1.3.4.Consultar\_Datos.sql*

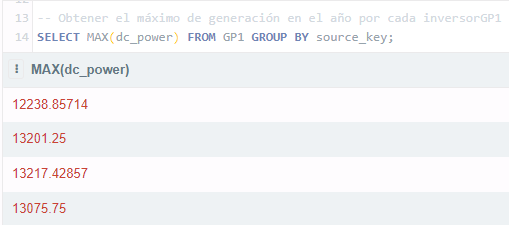
**RESPUESTA:**

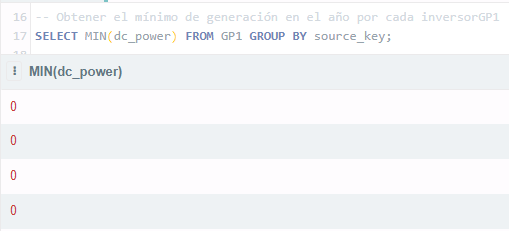












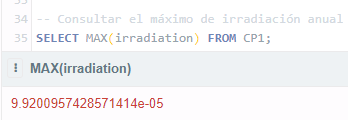












1. **MongoDB**
   1. **Sentencia o consulta de creación del documento(s)**

***Observación****: Escriba el código en MongoDB para crear al menos 20 documentos que correspondan a su conjunto de datos específico. Almacene en el repositorio* **(REPO EN GITHUB)** *el script con el nombre de T1.4.1.Creacion\_Documentos.sql.*

**RESPUESTA:**

db.PGD1.insertMany([

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4135001, "SOURCE\_KEY": "1BY6WEcLGh8j5v7", "DC\_POWER": 37.14285714, "AC\_POWER": 3.585714286, "DAILY\_YIELD": 0, "TOTAL YIELD": 6259559},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4135001, "SOURCE\_KEY": "1IF53ai7Xc0U56Y", "DC\_POWER": 53.5, "AC\_POWER": 5.1625, "DAILY\_YIELD": 0, "TOTAL YIELD": 6183645},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4135001, "SOURCE\_KEY": "3PZuoBAID5Wc2HD", "DC\_POWER": 58, "AC\_POWER": 5.585714286, "DAILY\_YIELD": 0, "TOTAL YIELD": 6987759},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4135001, "SOURCE\_KEY": "7JYdWkrLSPkdwr4", "DC\_POWER": 58.42857143, "AC\_POWER": 5.628571429, "DAILY\_YIELD": 0, "TOTAL YIELD": 7602960},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4135001, "SOURCE\_KEY": "McdE0feGgRqW7Ca", "DC\_POWER": 54.375, "AC\_POWER": 5.25, "DAILY\_YIELD": 0, "TOTAL YIELD": 7158964}

]);

db.PGD2.insertMany([

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4136001, "SOURCE\_KEY": "4UPUqMRk7TRMgml", "DC\_POWER": 15.406666666666665, "AC\_POWER": 14.86, "DAILY\_YIELD": 0.79999999999999993, "TOTAL YIELD": 2429011.8000000003},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4136001, "SOURCE\_KEY": "81aHJ1q11NBPMrL", "DC\_POWER": 14.773333333333333, "AC\_POWER": 14.246666666666668, "DAILY\_YIELD": 0.7333333333333335, "TOTAL YIELD": 1215278736.7333333},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4136001, "SOURCE\_KEY": "9kRcWv60rDACzjR", "DC\_POWER": 15.38, "AC\_POWER": 14.84, "DAILY\_YIELD": 1.8666666666666667, "TOTAL YIELD": 2247719578.8666668},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4136001, "SOURCE\_KEY": "Et9kgGMDl729KT4", "DC\_POWER": 14.973333333333334, "AC\_POWER": 14.440000000000001, "DAILY\_YIELD": 1, "TOTAL YIELD": 1704251.0000000002},

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4136001, "SOURCE\_KEY": "IQ2d7wF4YD8zU1Q", "DC\_POWER": 15.16, "AC\_POWER": 14.633333333333338, "DAILY\_YIELD": 0.73333333333333328, "TOTAL YIELD": 19941526.733333334}

]);

db.PWD1.insertMany([

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4135001, "SOURCE\_KEY": "HmiyD2TTLFNqkNe", "AMBIENT\_TEMPERATURE": 24.088446066666666, "MODULE\_TEMPERATURE": 22.206756600000002, "IRRADIATION": 0.0058869571866666665},

{"DATE\_TIME": "15-05-2020 06:15", "PLANT\_ID": 4135001, "SOURCE\_KEY": "HmiyD2TTLFNqkNe", "AMBIENT\_TEMPERATURE": 24.011635266666669, "MODULE\_TEMPERATURE": 22.353458666666668, "IRRADIATION": 0.022281607466666669},

{"DATE\_TIME": "15-05-2020 06:30", "PLANT\_ID": 4135001, "SOURCE\_KEY": "HmiyD2TTLFNqkNe", "AMBIENT\_TEMPERATURE": 23.976731266666665, "MODULE\_TEMPERATURE": 22.893282, "IRRADIATION": 0.049409723866666665},

{"DATE\_TIME": "15-05-2020 06:45", "PLANT\_ID": 4135001, "SOURCE\_KEY": "HmiyD2TTLFNqkNe", "AMBIENT\_TEMPERATURE": 24.21899, "MODULE\_TEMPERATURE": 24.442443933333333, "IRRADIATION": 0.0953944536},

{"DATE\_TIME": "15-05-2020 07:00", "PLANT\_ID": 4135001, "SOURCE\_KEY": "HmiyD2TTLFNqkNe", "AMBIENT\_TEMPERATURE": 24.537398399999997, "MODULE\_TEMPERATURE": 27.185652866666665, "IRRADIATION": 0.1419404433333333}

]);

db.PWD2.insertMany([

{"DATE\_TIME": "15-05-2020 06:00", "PLANT\_ID": 4136001, "SOURCE\_KEY": "iq8k7ZNt4Mwm3w0", "AMBIENT\_TEMPERATURE": 24.74278656666667, "MODULE\_TEMPERATURE": 24.077229833333334, "IRRADIATION": 0.012961978146666667},

{"DATE\_TIME": "15-05-2020 06:15", "PLANT\_ID": 4136001, "SOURCE\_KEY": "iq8k7ZNt4Mwm3w0", "AMBIENT\_TEMPERATURE": 24.763910896551721, "MODULE\_TEMPERATURE": 24.356189448275863, "IRRADIATION": 0.021037668724137933},

{"DATE\_TIME": "15-05-2020 06:30", "PLANT\_ID": 4136001, "SOURCE\_KEY": "iq8k7ZNt4Mwm3w0", "AMBIENT\_TEMPERATURE": 24.7863229, "MODULE\_TEMPERATURE": 24.548886100000004, "IRRADIATION": 0.042435399},

{"DATE\_TIME": "15-05-2020 06:45", "PLANT\_ID": 4136001, "SOURCE\_KEY": "iq8k7ZNt4Mwm3w0", "AMBIENT\_TEMPERATURE": 25.04824510344827, "MODULE\_TEMPERATURE": 25.746661999999997, "IRRADIATION": 0.1034958783448276},

{"DATE\_TIME": "15-05-2020 07:00", "PLANT\_ID": 4136001, "SOURCE\_KEY": "iq8k7ZNt4Mwm3w0", "AMBIENT\_TEMPERATURE": 25.6732025, "MODULE\_TEMPERATURE": 28.674707033333334, "IRRADIATION": 0.19945048366666665}

]);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A continuación se presentan dos de los documentos creados:

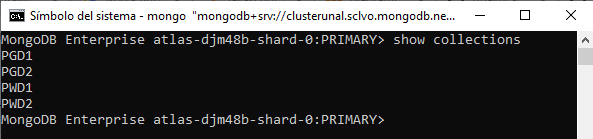


* 1. **Sentencia de consulta**

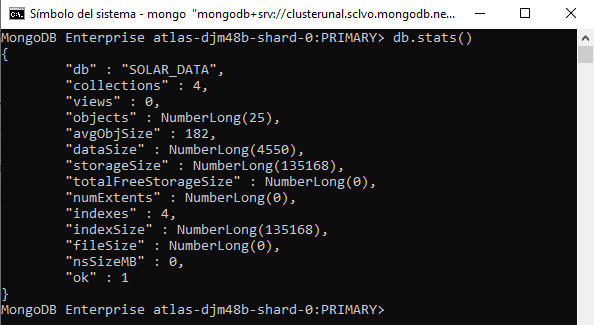
***Observación****: Realice la exploración básica de los datos, conteos totales y por categorías, máximos, promedio y mínimos.* Es decir, aplique estadística descriptiva con el fin de conocer las propiedades de los datos y entenderlos lo mejor posible. Use solamente sentencias SQL. Anexe las tomas de pantalla donde evidencie la sentencia y su correspondiente ejecución. Además, *Almacene en el repositorio* **(REPO EN GITHUB)** *el script con el nombre de T1.4.2.Consultar\_Datos.sql*

**RESPUESTA:**

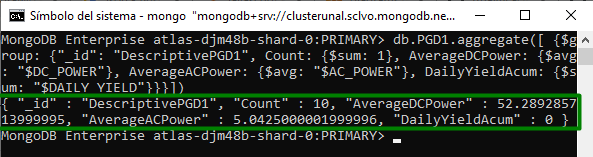
show collections



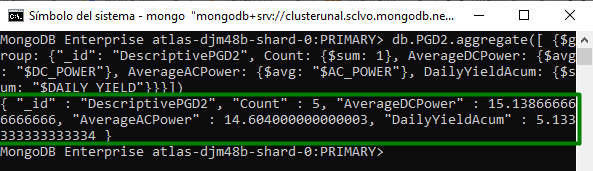
db.stats()



db.PGD1.aggregate([ {$group: {"\_id": "DescriptivePGD1", Count: {$sum: 1}, AverageDCPower: {$avg: "$DC\_POWER"}, AverageACPower: {$avg: "$AC\_POWER"}, DailyYieldAcum: {$sum: "$DAILY\_YIELD"}}}])



db.PGD2.aggregate([ {$group: {"\_id": "DescriptivePGD2", Count: {$sum: 1}, AverageDCPower: {$avg: "$DC\_POWER"}, AverageACPower: {$avg: "$AC\_POWER"}, DailyYieldAcum: {$sum: "$DAILY\_YIELD"}}}])



db.PWD1.aggregate([ {$group:

{"\_id": "DescriptivePWD1", Count: {$sum: 1},

AverageAT: {$avg: "$AMBIENT\_TEMPERATURE"},

AverageMT: {$avg: "$MODULE\_TEMPERATURE"},

AverageIrradiation: {$avg: "$IRRADIATION"},

MaxAT: {$max: "$AMBIENT\_TEMPERATURE"},

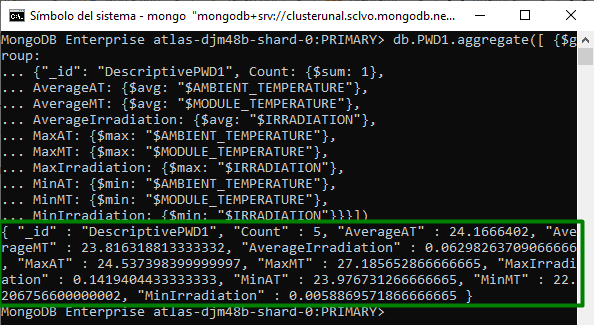
MaxMT: {$max: "$MODULE\_TEMPERATURE"},

MaxIrradiation: {$max: "$IRRADIATION"},

MinAT: {$min: "$AMBIENT\_TEMPERATURE"},

MinMT: {$min: "$MODULE\_TEMPERATURE"},

MinIrradiation: {$min: "$IRRADIATION"}}}])



db.PWD2.aggregate([

{$group: {"\_id": "DescriptivePWD2",

Count: {$sum: 1},

AverageAT: {$avg: "$AMBIENT\_TEMPERATURE"},

AverageMT: {$avg: "$MODULE\_TEMPERATURE"},

AverageIrradiation: {$avg: "$IRRADIATION"},

MaxAT: {$max: "$AMBIENT\_TEMPERATURE"},

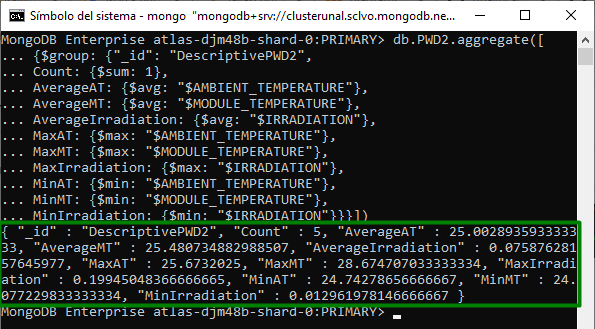
MaxMT: {$max: "$MODULE\_TEMPERATURE"},

MaxIrradiation: {$max: "$IRRADIATION"},

MinAT: {$min: "$AMBIENT\_TEMPERATURE"},

MinMT: {$min: "$MODULE\_TEMPERATURE"},

MinIrradiation: {$min: "$IRRADIATION"}}}])



1. **Análisis de lectura**

***Observación****: Considerando el artículo: “*The Definitive Guide to Graph Databases for the RDBMS Developer*” de Neo4J. Compartido en las carpeta de lecturas recomendadas. Analice y responda cada pregunta en máximo 150 palabras:*

1. ¿Cuáles son las limitaciones, que se pueden inferir de la lectura, para migrar los conjuntos de datos relacionales a NoSQL?

**R/**las limitación para migrar datos relacionales, se dan porque no son suficientes para manejar el volumen, la velocidad y variedad de datos actuales.

1. ¿Cuáles limitaciones adicionales se deben considerar, a parte de las mencionadas en el artículo?

**R/**Otro desafío está relacionado con la manera en que el modelo de base de datos relacional evita redundancia de datos, que es parte del modelos NoSQL. por lo que se convierte en un reto mantener todas las relaciones existentes, pues deben estar representados adecuadamente sin pérdida o distorsión de datos. Finalmente existe un costo asociado con la adaptación de las aplicaciones de software a comunicarse correctamente con el nuevo modelo de base de datos.

1. ¿Cuáles son las razones (criterios) que se deben considerar para migrar un conjunto de datos relacionados a NoSQL?

**R/** Cuando los datos deben ser consistentes sin dar posibilidad al error utilizar una base de datos relacional, SQL.

- Cuando nuestro presupuesto no se puede permitir grandes máquinas y debe destinarse a máquinas de menor rendimiento, NoSQL.

- Cuando las estructuras de datos que manejamos son variables, NoSQL.

- Análisis de grandes cantidades de datos en modo lectura, NoSQL.