Primer Avance del Trabajo Final: PostgreSQL para Ciencia de Datos

Profesor: Yoseph Ayala Valencia

---

1. Objetivo del Avance

El propósito de este primer avance es que los alumnos demuestren su capacidad para:

- Seleccionar y justificar un dataset de Kaggle para un análisis de ciencia de datos.
- Plantear una pregunta de investigación o definir una problemática de negocio que se abordará con el dataset seleccionado.
- Diseñar el esquema lógico de la base de datos a implementar en PostgreSQL.
- Integrar Python con PostgreSQL para la carga, manipulación y consulta de datos.
- Aplicar los conocimientos vistos en clase, desde la creación y gestión de bases de datos hasta la realización de consultas SQL básicas, el uso de filtros, condicionales, joins, subconsultas, funciones de agregación, agrupamiento y funciones de ventana.

---

2. Requerimientos del Trabajo

2.1 Selección y Planteamiento del Problema

- **Elección del Dataset:**
  - Seleccionar y descargar un dataset de Kaggle.
  - Describir brevemente el origen del dataset, la temática y la razón por la cual es interesante para el análisis de datos.
- **Pregunta de Investigación / Problemática de Negocio:**
  - Formular una pregunta de investigación o definir una problemática de negocio que se pretende resolver o analizar con el dataset.
  - *Ejemplo:* "¿Cómo influyen las variables X, Y y Z en la predicción del valor de venta de un producto?"

2.2 Diseño del Esquema de la Base de Datos

- **Modelo Lógico:**
  - Presentar un diagrama detallado del modelo lógico de la base de datos, definiendo las tablas, campos, tipos de datos y relaciones entre tablas. Se recomienda utilizar dbdiagram (visto en la primera clase)
- **Justificación del Diseño:**
  - Explicar brevemente las decisiones tomadas para estructurar la base de datos, en función de la pregunta de investigación o problemática planteada.

2.3 Implementación en PostgreSQL

a) Creación de la Base de Datos y Tablas **(Hacerlo desde Postgresql)**

- Crear la base de datos y las tablas necesarias siguiendo el modelo lógico propuesto.

b) Carga de Datos desde Python **(Hacerlo desde Python)**

Los alumnos tendrán dos opciones para cargar la data de Kaggle en PostgreSQL:

1. **Opción A: Uso de la API de Kaggle directamente en Python**
    - Conectar con la API de Kaggle usando Python para descargar el dataset de forma programática.
    - Procesar y preparar la data (puede venir en formato JSON, CSV, etc.) y luego cargarla en PostgreSQL.
    - Incluir fragmentos de código que muestren:
        - La conexión y descarga mediante la API de Kaggle.
        - La conexión a PostgreSQL (por ejemplo, usando la librería `psycopg2`).
        - La inserción de datos en la base de datos.
2. **Opción B: Descarga manual de los archivos CSV/Excel desde Kaggle**
    - Descargar manualmente los archivos CSV/Excel del dataset elegido.
    - Usar Python (por ejemplo, mediante `pandas`) para leer los archivos CSV/Excel.
    - Realizar cualquier transformación o limpieza de datos necesaria.
    - Conectar a PostgreSQL y subir los datos a la base de datos.
    - Incluir fragmentos de código que muestren:
        - La lectura del archivo CSV en Python.
        - La conexión a PostgreSQL y la inserción de datos.

2.4 Consultas y Análisis Básicos (Hacerlo desde Postgresql)

- **Consultas SQL Básicas:**
    - Realizar consultas usando `SELECT` y `WHERE` junto con operadores de comparación para extraer información relevante.
- **Uso de Filtros y Condicionales:**
    - Aplicar operadores matemáticos, de texto, de fecha en las consultas.
    - Incluir al menos un ejemplo del uso de la cláusula `CASE WHEN` para generar columnas condicionales.
- **Integración de Joins y Subconsultas:**
    - Incluir consultas que involucre `INNER JOIN` (u otro tipo de join) y/o subconsultas para relacionar la información.
- **Funciones de Agregación y Agrupamiento:**
    - Desarrollar consultas que utilicen funciones de agregación (como `SUM`, `COUNT`, `AVG`, `MIN`, `MAX`) junto con la cláusula `GROUP BY` y, si corresponde, `HAVING`.
    - Incluir al menos un ejemplo en el que se empleen funciones de ventana (usando la cláusula `OVER`).

2.5 Documentación y Presentación del Avance

- **Informe Técnico:**
  - Elaborar un documento (en PDF) que incluya:
    - **Introducción:** Breve descripción del proyecto, el dataset seleccionado y la problemática o pregunta de investigación.
    - **Esquema de la Base de Datos:** Diagrama o descripción detallada del modelo lógico.
    - **Desarrollo:**
      - Detalles de la implementación en PostgreSQL (código SQL para la creación de la base de datos, tablas, y ejemplos de manipulación de datos).
      - Fragmentos de código en Python que evidencien la conexión y carga de datos en PostgreSQL.
      - Ejemplos de consultas realizadas y los resultados obtenidos (pueden incluirse capturas de pantalla o logs de salida).
    - **Conclusiones:** Reflexiones sobre los desafíos encontrados en esta fase
- **Código Fuente:**
  - Incluir todos los scripts SQL y Python utilizados al momento de subir el trabajo en la plataforma.

---

3. Criterios de Evaluación

La evaluación se basará en la siguiente rúbrica:

Criterio	Peso	Descripción
Selección y Justificación del Dataset	15%	- Elección adecuada y justificada del dataset. - Claridad en la descripción de la fuente y relevancia para la problemática propuesta.
Pregunta de Investigación / Problemática	15%	- Formulación clara y precisa de la pregunta o problemática. - Relación directa con el dataset seleccionado.
Diseño del Esquema de la Base de Datos	20%	- Presentación de un modelo lógico coherente y bien estructurado. - Justificación de las decisiones de diseño (tablas, relaciones, etc.).
Implementación en PostgreSQL	20%	- Creación correcta de la base de datos y tablas utilizando SQL. - Ejemplos funcionales de INSERT, UPDATE y DELETE. - Evidencia de carga de datos desde Python (opción A o B).
Consultas y Uso de Funcionalidades SQL	20%	- Uso correcto de consultas básicas (SELECT, WHERE). - Aplicación de filtros, condicionales, funciones de agregación, agrupamiento y, en caso de haber, joins y subconsultas.

Criterio	Peso	Descripción
Documentación y Presentación	10%	- Claridad y coherencia en la redacción del informe. - Inclusión de fragmentos de código, resultados y capturas de pantalla que respalden lo desarrollado. - Organización general del trabajo.

---

4. Indicaciones Generales

- **Equipos:**
  - El trabajo es de integrantes de **4 personas**
- **Formato y Entrega:**
  - La entrega del primer avance deberá incluir un informe en PDF y los archivos de código (scripts SQL y Python).
  - La fecha límite para la entrega es el **15 de febrero**.
- **Originalidad y Buenas Prácticas:**
  - Se valorará la originalidad en la elección del dataset y en el planteamiento de la problemática.
  - Se espera que el código esté bien comentado y estructurado, siguiendo buenas prácticas de programación.