# PR_06.2

## 💾 Configuración Previa Sugerida

Para la práctica, se necesitarán al menos las siguientes tres tablas:

1. `u_data`: Calificaciones (UserID, MovieID, Rating, Timestamp).
2. `u_user`: Usuarios (UserID, Age, Gender, Occupation, ZipCode).
3. `u_item`: Películas (MovieID, MovieTitle, ReleaseDate, **Genres**...).

**Nota:** Presten especial atención a cómo cargar los datos, ya que los archivos MovieLens 100k usan diferentes delimitadores (barra vertical `|` y tabulador `\t`).

---

## 📝Ejercicios de Apache HiveQL con MovieLens

### I. Gestión de Tablas y Formato de Datos (CREATE TABLE, EXTERNAL, ROW FORMAT)

**Objetivo:** Practicar la definición de esquemas, delimitadores y tipos de tablas.

1. **Carga de Datos Delimitados (Básico):**
   - Crea una **Tabla Externa** llamada `ml_ratings_ext` para el archivo `u.data`.
   - Asegúrate de especificar que los campos están delimitados por el carácter **tabulador** (`\t`) y que las filas terminan en salto de línea.
   - Define las cuatro columnas: `user_id`, `movie_id`, `rating` (INT) y `timestamp` (BIGINT).
2. **Definición de Esquema y Comentarios:**
   - Crea una **Tabla Gestionada** (Managed) llamada `ml_users_managed` para el archivo `u.user`.
   - Especifica el delimitador **barra vertical** (`|`).
   - Añade un `COMMENT` a la tabla y un `COMMENT` a la columna `occupation` explicando su uso.
3. **Simulación de Pérdida de Datos (DROP TABLE):**
   - Ejecuta `DROP TABLE ml_ratings_ext;`. ¿Se eliminaron los datos de HDFS? Explica por qué.

- Ejecuta `DROP TABLE ml_users_managed;`. ¿Se eliminaron los datos de HDFS? Explica por qué.

---

## II. Tipos de Datos Complejos (ARRAY, MAP, STRUCT)

**Objetivo:** Practicar la definición de tipos complejos y el uso de las funciones `size()`, `map_keys()` y el acceso mediante `[]` y `.`.

1. **Uso de STRUCT (Información Personal):**
   - Crea una nueva tabla `ml_user_info` basada en `u_user`.
   - Combina las columnas `age` (INT) y `gender` (STRING) en una sola columna de tipo **STRUCT** llamada `personal_data`.
   - Escribe una consulta para seleccionar el campo `gender` de esta nueva columna y filtra por todas las mujeres (`'F'`).
2. **Uso de ARRAY (Géneros de Película):**
   - El archivo `u.item` tiene 19 columnas binarias para géneros. Simula que ya han sido procesadas en una columna de tipo **ARRAY** llamada `genres_list` (`ARRAY<STRING>`).
   - Escribe una consulta que devuelva el **tamaño** (`size()`) de esta lista de géneros para cada película.
   - Muestra el **segundo** género (índice `1`) de la película con `movie_id = 50`.
3. **Uso de MAP (Puntuación de Usuarios por Ocupación):**
   - Simula que tienes una tabla que resume la puntuación promedio (`FLOAT`) de los usuarios en función de su ocupación (`STRING`), almacenada en un `MAP<STRING, FLOAT>` llamado `avg_rating_by_occupation`.
   - Utiliza la función **`map_keys()`** para listar todas las ocupaciones diferentes que existen en el `MAP`.

---

## III. Consultas Básicas y JOINS

**Objetivo:** Practicar `SELECT`, `WHERE`, `GROUP BY`, `JOIN` y `LEFT SEMI JOIN`.

1. **Agregación Básica:**
   - Calcula el **promedio** de todas las calificaciones (`rating`) que existen en la tabla `ml_ratings_ext`.

2. **Filtrado y Agrupación:**
    - Agrupa los usuarios por `gender` y `occupation` (de `ml_users_managed`).
    - Calcula el número de usuarios que hay en cada combinación (e.g., `'M', 'programmer'`, `'F', 'student'`).
    - Ordena el resultado de forma descendente por el conteo.
3. **INNER JOIN:**
    - Utiliza un `INNER JOIN` entre `ml_ratings_ext` y `ml_users_managed` sobre `user_id`.
    - Calcula la **calificación promedio** por `occupation`.
4. **LEFT OUTER JOIN:**
    - Crea una tabla ficticia `unrated_users` que contenga solo 5 `user_id`s de la tabla `ml_users_managed` que *no* hayan calificado ninguna película.
    - Realiza un `LEFT OUTER JOIN` desde `ml_users_managed` a `ml_ratings_ext`.
    - Filtra el resultado para encontrar aquellos usuarios que **no tienen calificaciones** (donde el `rating` es `NULL` después del *join*).
5. **LEFT SEMI JOIN (Simulación de IN/EXISTS):**
    - Usando `ml_ratings_ext` y `ml_users_managed`, lista todos los **`user_id`** que tienen una ocupación de `'programmer'` y que **han calificado al menos una película**. (Debe usarse `LEFT SEMI JOIN`, no un `INNER JOIN` simple).