Desafío - Manipulación de datos y transaccionalidad en las operaciones

1. Creamos la base de Datos en la terminal de Pgadmin con la herramienta `PSQL Tool`
2. Ingresamos el comando para la creación de bases de dato desde la terminal:

| postgres=# CREATE DATABASE "desafio2-Sebastian-Vera-123";  CREATE DATABASE |
| --- |

1. Nos conectamos a la base de datos utilizando el comando de psql `\c`, de la siguiente manera:

| postgres=# \c desafio2-Sebastian-Vera-123; Ahora está conectado a la base de datos «desafio2-Sebastian-Vera-123» con el usuario «postgr es».  desafio2-Sebastian-Vera-123=# |
| --- |

1. Una vez conectado a la base de datos procedemos a crear la base de datos “inscritos” e ingresar los valores tal como se dispone en la documentación del desafío:

| desafio2-Sebastian-Vera-123=# CREATE TABLE IF NOT EXISTS INSCRITOS(cantidad INT, fecha DATE, fuente); CREATE TABLE desafio2-Sebastian-Vera-123=# INSERT INTO INSCRITOS(cantidad, fecha, fuente) desafio2-Sebastian-Vera-123-# VALUES ( 44, '01/01/2021', 'Blog' ); INSERT 0 1 - se agrega el resto de valores ... |
| --- |

1. Finalmente, antes de realizar las querys verificamos que los registros se crearan correctamente:

| desafio2-Sebastian-Vera-123=# SELECT \* FROM inscritos;  cantidad | fecha | fuente ----------+------------+--------  44 | 2021-01-01 | Blog  56 | 2021-01-01 | Página  39 | 2021-02-01 | Blog  81 | 2021-02-01 | Página  12 | 2021-03-01 | Blog  91 | 2021-03-01 | Página  48 | 2021-04-01 | Blog  45 | 2021-04-01 | Página  55 | 2021-05-01 | Blog  33 | 2021-05-01 | Página  18 | 2021-06-01 | Blog  12 | 2021-06-01 | Página  34 | 2021-07-01 | Blog  24 | 2021-07-01 | Página  83 | 2021-08-01 | Blog  99 | 2021-08-01 | Página (16 filas) |
| --- |

Querys

1. Cantidad total de registros:

En la tabla "inscritos" hay un total de 16 registros.

| desafio2-Sebastian-Vera-123=# SELECT COUNT(\*) FROM inscritos;  count -------  16 (1 fila) |
| --- |

2. Cantidad total de inscritos:

En la tabla "inscritos" hay un total de 774 inscritos.

| desafio2-Sebastian-Vera-123=# SELECT SUM(cantidad) as total\_inscritos FROM ins critos;  total\_inscritos -----------------  774 (1 fila) |
| --- |

En esta query estamos sumando la columna “cantidad” con todos los valores resumidos en un alias “total\_insccritos”.

3. Registro o registros de mayor antigüedad:

El registro o registros de mayor antigüedad en la tabla "inscritos" son:

| Cantidad | Fecha | Fuente |
| --- | --- | --- |
| 44 | 2021-01-01 | Fuente |
| 56 | 2021-01-01 | Página |

| desafio2-Sebastian-Vera-123=# SELECT \* FROM inscritos WHERE fecha = (SELECT MIN(fecha) FROM inscritos);  cantidad | fecha | fuente ----------+------------+--------  *44 | 2021-01-01 | Blog*  *56 | 2021-01-01 | Página* (2 filas) |
| --- |

Lo que hacemos en esta consulta es buscar todas las filas en la tabla "inscritos" que tengan la fecha mínima encontrada en la subconsulta, lo que significa que se seleccionarán los registros de mayor antigüedad ya que la fecha más antigua será menor a la más actual.

4. Cantidad de inscritos por día:

La cantidad de inscritos por día en la tabla "inscritos" a partir de la fecha actual es:

| desafio2-Sebastian-Vera-123=# SELECT fecha, SUM(cantidad) as total\_inscritos\_p or\_dia FROM inscritos GROUP BY fecha ORDER BY fecha ASC;  fecha | total\_inscritos\_por\_dia ------------+-------------------------  2021-01-01 | 100  2021-02-01 | 120  2021-03-01 | 103  2021-04-01 | 93  2021-05-01 | 88  2021-06-01 | 30  2021-07-01 | 58  2021-08-01 | 182 (8 filas) |
| --- |

En esta búsqueda agrupamos los resultados por la columna "fecha" y utilizamos la función de agregación SUM() para sumar la columna "cantidad" de cada grupo de fechas y ordenarlos de manera ascendente para una mejor visualización.

5. Día con mayor cantidad de inscritos:

El día con mayor cantidad de inscritos en la tabla "inscritos" es:

| desafio2-Sebastian-Vera-123=# SELECT fecha, SUM(cantidad) as total\_inscritos FROM inscritos GROUP BY fecha ORDER BY total\_inscritos DESCLIMIT 1;   fecha | total\_inscritos ------------+-----------------  2021-08-01 | 182 (1 fila) |
| --- |

por ultimo, buscamos todas las filas en la tabla "inscritos", agrupamos los resultados por la columna "fecha" y utilizamos la función SUM() para sumar la columna "cantidad" de cada grupo de fechas. Luego, ordena los resultados por el total de inscritos en orden descendente y limitamos el resultado utilizando LIMIT 1.