**Fragmentación con PostgreSQL**

Profesor Heider Sanchez

Se le pide crear las siguientes tablas “employees” y “salaries” y cargar los datos del dataset “data2.zip”. Talvez requiera realizar algunas modificaciones para que la data pueda ser cargado al PostgreSQL correctamente.

**P1. Fragmentación con PARTITION BY LIST**

* Crear la tabla employees1 indicando que será fragmentado por el atributo dept\_no.
* Cargue los datos en la tabla employees1
* Analice los resultados que se obtienen al ejecutar una misma consulta en el atributo dept\_no en ambas tablas.
* Use el comando Explain Analyze y coloque los tiempos en una tabla comparativa.

|  |  |  |  |
| --- | --- | --- | --- |
|  | dept\_no = d005 | dept\_no = d004 | dept\_no = d007 |
| employees | 45.968 ms | 30.057 ms | 30.179 ms |
| employees1 | 14.598 ms | 15.086 ms | 14.603 ms |

***Usar*** *SET enable\_partition\_pruning = on;*

**P2. Fragmentación con PARTITION BY RANGE**

* Crear la tabla employees2 indicando que será fragmentado por rango sobre el año de la fecha de contrato:

PARTITION BY RANGE (date\_part('year', hire\_date))

* Realizar la fragmentación con respecto al año (vector: [1988, 1994]).
* Cargue los datos en la tabla employees2.
* Elabore el cuadro de comparación de costos para tres consultas por rango diferentes sobre el atributo hire\_date. Cuidar que la query acceda a una sola partición.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Query 1 | Query 2 | Query 3 |
| employees |  |  |  |
| employees2 |  |  |  |

* Aplique el índice btree sobre el atributo hire\_date en ambas tablas. Note usted que el índice aplicado sobre la tabla particionada es un índice distribuido. Vuelva a ejecutar las consultas y anote los tiempos.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Query 1 | Query 2 | Query 3 |
| employees |  |  |  |
| employees2 |  |  |  |

**P3. Fragmentación con dos atributos**.

* Crear la tabla employees3 para ser particionada en dos atributos.
* Agregar la columna “salary” a la tabla employees para guardar el último salario de cada empleado. Obtener el valor del salario desde la tabla “salaries”.
* Considerar un segundo predicado de consulta sobre el atributo salary
  + - Proponer un vector de particionamiento lo más equitativo posible.
* Hay dos opciones de fragmentación con dos atributos:
  + Opción 1: Primero fragmentar la tabla en el atributo hire\_date. Luego aplicar una sub fragmentación sobre cada partición en el atributo salary.
  + Opción 2: Crear la tabla employees3 indicando ambos atributos en la partición:

PARTITION BY RANGE (date\_part('year', hire\_date), salary)

* Mostrar el plan de ejecución con Explain Analyze para tres consultas que incluya ambos atributos, hire\_data y salary.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Query 1 | Query 2 | Query 3 |
| employees | 136.051 ms | 48.648 ms | 47.939 ms |
| employees3 | 0.110 ms | 0.06 ms | 0.092 ms |

El gráfico para todas las consultas en employees sería el siguiente:

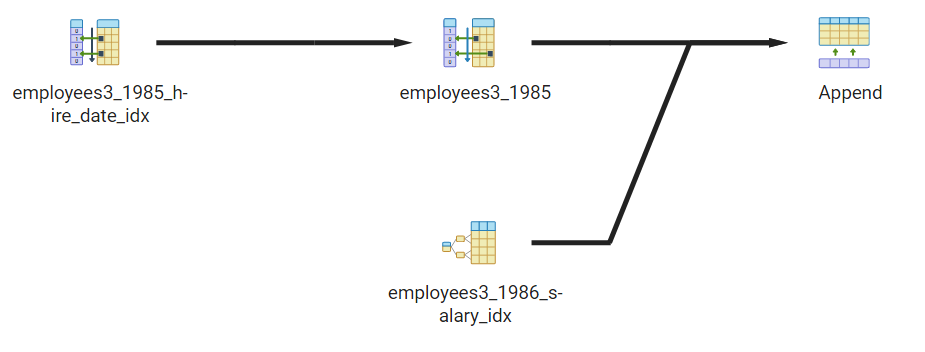


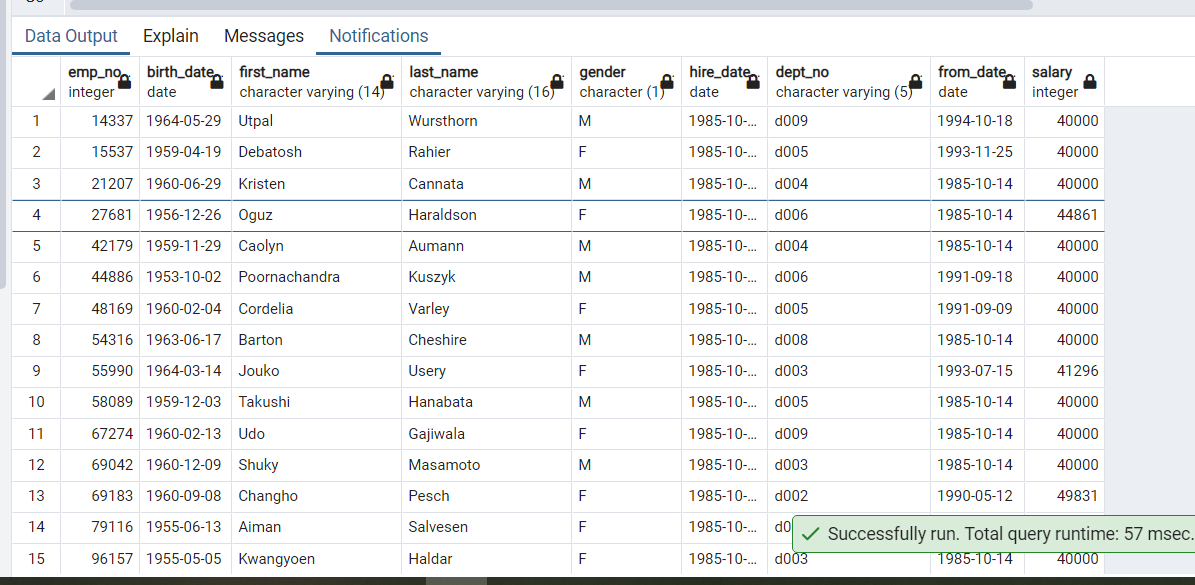
, pues no usa particiones, solo busca en toda la tabla.

En cambio, los gráficos de las consultas en employees3 son:

* Query 1

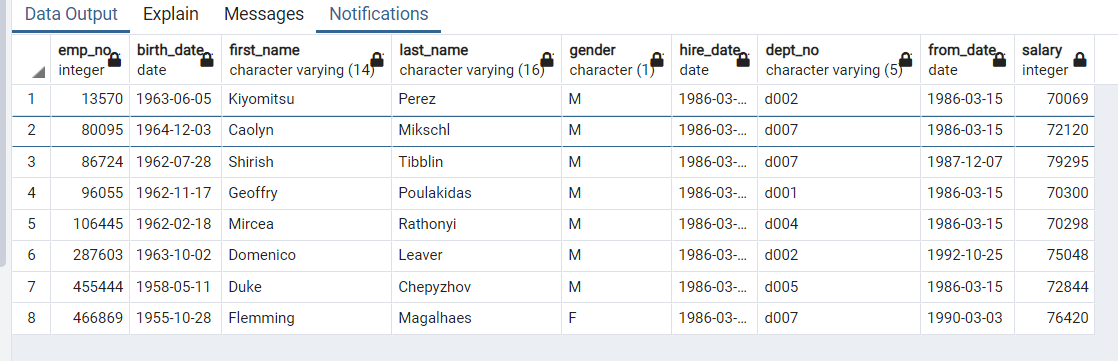
EXPLAIN ANALYZE SELECT \* FROM employees3 WHERE hire\_date = '1985-10-14' AND salary<50000;

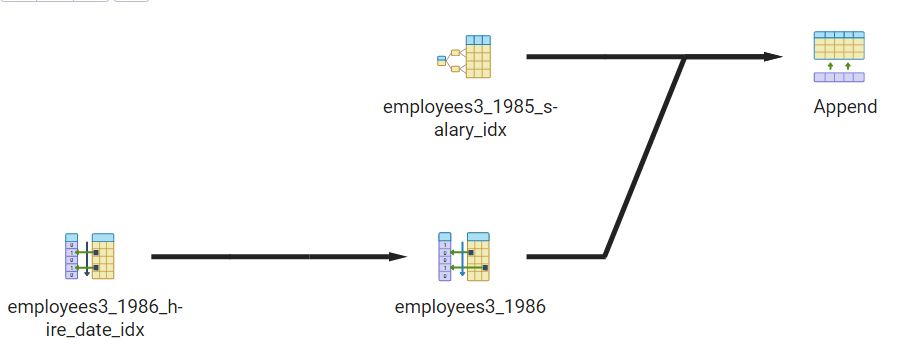




* Query 2

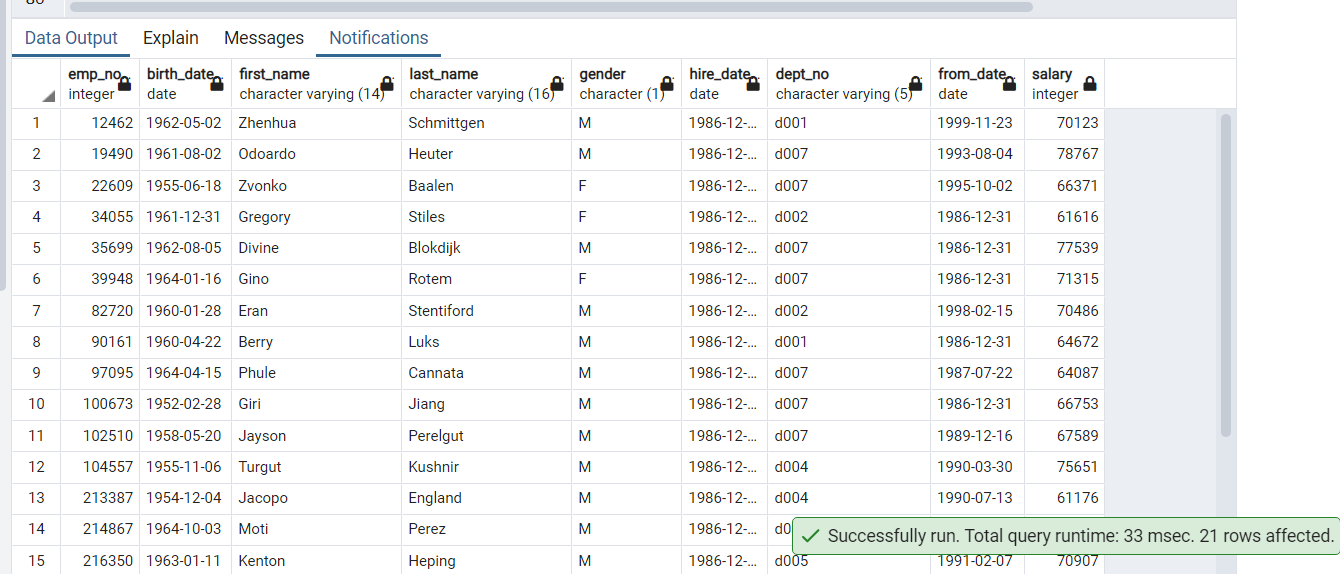
SELECT \* FROM employees3 WHERE hire\_date = '1986-03-15' AND salary>70000;

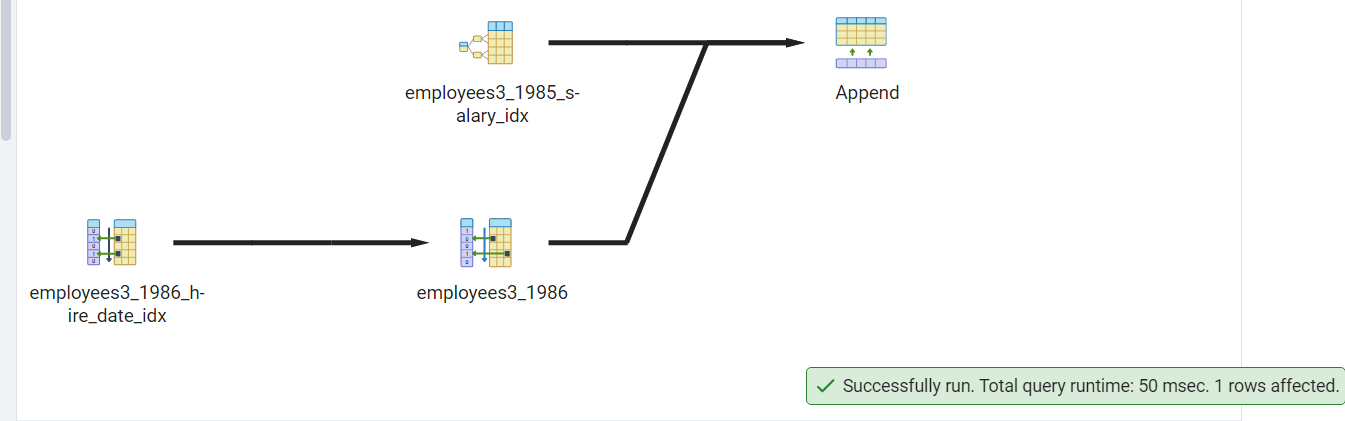




* Query 3

SELECT \* FROM employees3 WHERE hire\_date = '1986-12-31' AND salary BETWEEN 60000 AND 80000;





**Entregable:**  el script y resultado en formato PDF.