Fragmentación con PostgreSQL

Profesor Heider Sanchez

El objetivo del laboratorio es familiarizarnos con las características de fragmentación horizontal de base de datos que nos provee PostgreSQL. PostgreSQL a partir de la versión 10 soporta la partición básica de tablas usando la técnica de particionamiento por rango (PARTITION BY RANGE) y para consultas puntuales (PARTITION BY LIST). Desde la versión 11 se han agregado nuevas características de particionamiento como el PARTITION BY HASH. En versiones anteriores se debería emular la fragmentación usando herencia de tablas con INHERITANCE (ver manual de PostgreSQL).

P0. Pasos para crear una fragmentación con PARTITION BY LIST

1- Crear la tabla Employees indicando que será fragmentado por el atributo department_id.

```
CREATE TABLE employees
   ( employee_id    INTEGER
   , first_name     VARCHAR(20)
   , last_name      VARCHAR(25) NOT NULL
   , email          VARCHAR(25) NOT NULL
   , phone_number   VARCHAR(20)
   , hire_date      TIMESTAMP  NOT NULL
   , job_id         VARCHAR(10) NOT NULL
   , salary         NUMERIC(8,2)
   , commission_pct NUMERIC(2,2)
   , manager_id     INTEGER
   , department_id  INTEGER
   , CONSTRAINT     emp_salary_min  CHECK (salary > 0)
   ) PARTITION BY LIST (department_id);
```

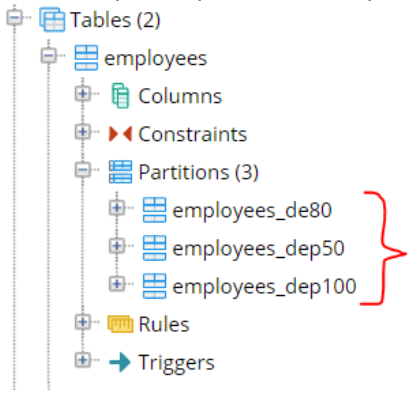
2- Crear la fragmentación:

```
CREATE TABLE employees_dep100 PARTITION OF employees FOR VALUES IN (100);

CREATE TABLE employees_dep50 PARTITION OF employees  FOR VALUES IN (50);

CREATE TABLE employees_dep80 PARTITION OF employees  FOR VALUES IN (80);
```

3- Verificar que las particiones hayan sido creadas



4- Probar la fragmentación insertando los siguientes datos:

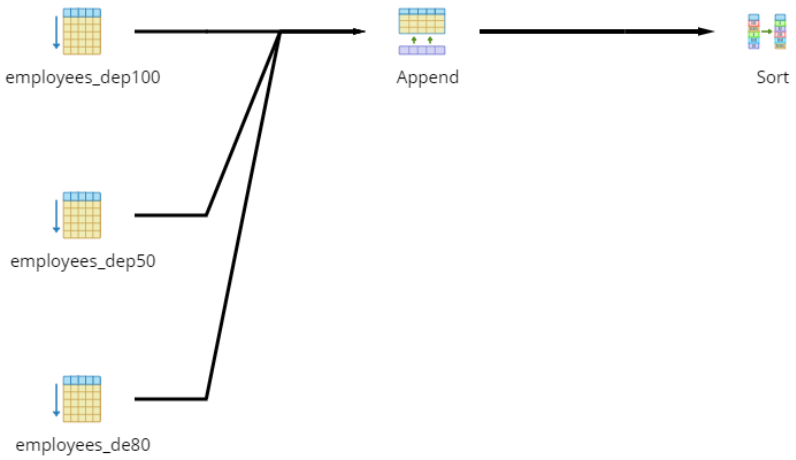
```
INSERT INTO employees VALUES ( 1, 'Jhon', 'King', 'JKING', '515.123.4567', TO_DATE('17-JUN-1987', 'dd-MON-yyyy'), 'AD_PRES', 24000, NULL, NULL, 100);

INSERT INTO employees VALUES  (2, 'Karen', 'Fripp', 'KFRIPP', '515.123.4567', TO_DATE('25-JUN-1987', 'dd-MON-yyyy'), 'PU_CLERK', 21000, NULL, NULL, 50);

INSERT INTO employees VALUES  (3, 'Steve', 'Jobs', 'SJOBS', '515.123.4567', TO_DATE('02-OCT-1975', 'dd-MON-yyyy'), 'HR_REP', 81000, NULL, NULL, 80);
```

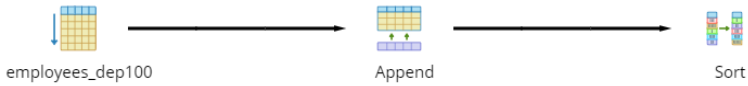
5- Traer los datos de todos los empleados insertados:

```
SELECT * FROM employees ORDER BY employee_id;
```



6- Su verdadero efecto es cuando aplicamos consultas puntuales por department_id:

```
SELECT * FROM employees WHERE department_id = 100 ORDER BY employee_id;
```



7- También podemos traer los datos directamente de la partición:

```
SELECT * FROM employees_dep100;
```

8- Para eliminar una partición la mejor forma sería:

```
ALTER TABLE employees DETACH PARTITION employees_dep100;
```

P1. Ejecutar los siguientes enunciados y analizar su resultado

1- Crear otra partición para el mismo departament_id = 80.
2- Crear una partición para juntar los empleados de los departamentos 20, 30 y 60.
3- Crear una partición para los departamentos que no son 50, 80 y 100. ¿Cuál sería la solución?
4- Eliminar las particiones 80 y 100 y volverlos a crear pero que los datos sean almacenados en "otro disco duro". Para ello se debe crear un nuevo TABLESPACE.

P2. Tiempo de respuesta

- ✓ Cree otra tabla employees pero sin partición (employees_1)
- ✓ Cargue los datos de la carpeta compartida en ambas tablas: employees y employees1
- ✓ Analice los resultados que se obtienen al ejecutar la misma consulta por departamento_id en ambas tablas. Use el comando Explain Analyze y coloque los costos en una tabla comparativa.

	department_id ==50	department_id ==80	department_id ==100
employees			
employees1			

P3. Fragmentación con PARTITION BY RANGE

1- En otro esquema, crear la tabla Employees indicando que será fragmentado por rango sobre el año de la fecha de contrato:

PARTITION BY RANGE (date_part('year', hire_date))

2- Realizar la fragmentación con respecto al año (vector: [1995, 1998]).
3- Crear un índice para el atributo hire_date en cada partición.
4- Cargue los datos de la carpeta compartida.
5- Elabore el cuadro de comparación de costos al ejecutar tres consultas.

P4. Fragmentación con dos atributos.

1. Considerar un segundo predicado de consulta sobre el atributo salary (vector: [30k,70k]).
2. Hay dos opciones de fragmentación con dos atributos:
   2.1. Sobre la fragmentación anterior realizar una sub fragmentación sobre cada partición con el atributo salary.
   2.2. Crear otra tabla Employees indicando ambos atributos en la partición:

   PARTITION BY RANGE (date_part('year', hire_date)), salary)
3. Mostrar el gráfico de Explain Analyze para tres consultas (similar al P0.5).

Entregable: en formato PDF el P2, P3 y P4.

[1]. Guia oficial de PostgreSQL, https://www.postgresql.org/docs/10/ddl-partitioning.html