Informe - Laboratorio 9

1. Optimización de consultas por departamento

- Creación de tablas

Luego de crear las tablas employees y salaries, creamos la tabla employees1, la cual está particionada por list(dept_no).

```
CREATE TABLE employees1 (
emp_no int,
birth_date date,
first_name varchar(14),
last_name varchar(16),
gender character(1),
hire_date date,
dept_no varchar(5),
from_date date
) PARTITION BY LIST(dept_no);
```

- Distribución de datos:

```
select dept_no, count(*) as count
from employees
group by dept_no
order by dept_no;
```

	dept_no	count
1	d001	20211
2	d002	17346
3	d003	17786
4	d004	73485
5	d005	85707
6	d006	20117
7	d007	52245
8	d008	21126
9	d009	23580

Ahora, con el objetivo de acelerar las consultas por departamento, debemos crear tres fragmentos lo más balanceados posibles. Para ello, aplicamos un algoritmo greedy:

- Sea A = lista de departamentos ordenados de forma decreciente por número de empleados, P, Q, R los tres fragmentos resultantes, insertar las tuplas correspondientes al departamento A[0] en X : |X| = min{|P|, |Q|, |R|}.

Luego, tenemos tres fragmentos balanceados:

- Fragmento 1: d005, d006 (105824 empleados)

- Fragmento 2: d003, d004, d008 (112397 empleados)
- Fragmento 3: d001, d002, d007, d009 (113382 empleados)

```
CREATE TABLE employees1_d005_d006
     PARTITION OF employees1 FOR VALUES IN ('d005', 'd006');

CREATE TABLE employees1_d004_d008_d003
     PARTITION OF employees1 FOR VALUES IN ('d004', 'd008', 'd003');

CREATE TABLE employees1_d007_d009_d001_d002
     PARTITION OF employees1 FOR VALUES IN ('d007', 'd009', 'd001',
'd002');
```

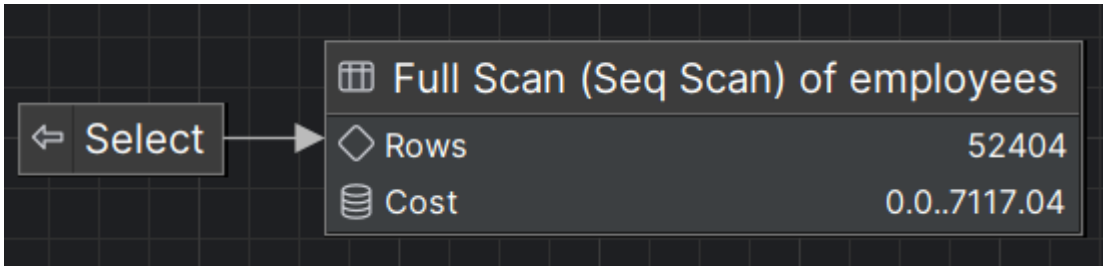
- Comparación de rendimiento:

```
vacuum full employees;
vacuum full employees_1;

explain analyse
select *
from employees
where dept_no = 'd005';

explain analyse
select *
from employees_p1
where dept_no = 'd005';
```

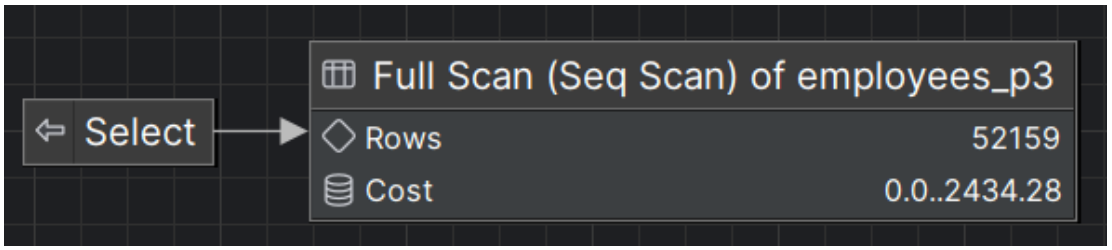
Consulta	Sin Fragmentación	Con Fragmentación	Mejora(%)
d005	70.639 ms	23.946 ms	66.100%
d004	72.721 ms	25.389 ms	65.087%
d007	63.709 ms	24.331 ms	61.809%

- Plan de trabajo:

Sin Fragmentación:



Con Fragmentación:



1. **Fragmentación con partition by range**

- Distribución de los datos:

```
select date_part('year', hire_date) as year, count(*) as count
from employees
group by year
order by year;
```

	year	count
1	1985	39080
2	1986	40005
3	1987	36930
4	1988	34705
5	1989	31348
6	1990	28328
7	1991	24934
8	1992	22539
9	1993	19667
10	1994	16463
11	1995	13413
12	1996	10568
13	1997	7375
14	1998	4562
15	1999	1671
16	2000	15

```
select '1987 minor equal' as range, count(*) as count
from employees
where date_part('year', hire_date) <= 1987
union all
select '1988-1991' as range, count(*) as count
from employees
where date_part('year', hire_date) between 1988 and 1991
union all
select '1992 greater equal' as range, count(*) as count
from employees
```

```
where date_part('year', hire_date) >= 1992
order by range;
```

	range	count
1	1987 minor equal	116015
2	1988-1991	119315
3	1992 greater equal	96273

- Creación de tabla y particiones

```
create table if not exists employees_2
(
    emp_no     int,
    birth_date date,
    first_name varchar(14),
    last_name  varchar(16),
    gender     character(1),
    hire_date  date,
    dept_no    varchar(5),
    from_date  date
) partition by range (date_part('year', hire_date));

create table employees2_p1 partition of employees_2
    for values from (minvalue) to (1987);

create table employees2_p2 partition of employees_2
    for values from (1988) to (1991);

create table employees2_p3 partition of employees_2
    for values from (1992) to (maxvalue);
```

- Carga de datos:

```
select 'p1' as partition, count(*) as count
from employees2_p1
union all
select 'p2' as partition, count(*) as count
from employees2_p2
union all
select 'p3' as partition, count(*) as count
from employees2_p3
union all
select 'union' as partition, sum(total.count)
from (select count(*) as count
      from employees2_p1
      union all
      select count(*) as count
      from employees2_p2
      union all
```

```
      select count(*) as count
      from employees2_p3) as total
union all
select 'original' as partition, count(*) as count
from employees;
```

	partition	count
1	p1	116015
2	p2	119315
3	p3	96273
4	union	331603
5	original	331603

- Pruebas