

Fragmentación con PostgreSQL

Profesor: Heider Sanchez

Estudiantes:

- Ian González
- Benjamín Soto

Repositorio de Github: db2-parallel

P0. Crear tabla particionada

```
1 drop table if exists estudiante;
2 create table estudiante (
3     DNI char(4) not null,
4     Nombre varchar not null,
5     Ciudad varchar(20) not null,
6     Grupo char(1),
7     Promedio float,
8     Edad smallint,
9     Sexo char(1)
10 )
11 partition by list (Ciudad);
12
13 drop table if exists Aux;
14 create table Aux (
15     DNI char(4) not null,
16     Nombre varchar not null,
17     Ciudad varchar(20) not null,
18     Grupo char(1),
19     Promedio float,
20     Edad smallint,
21     Sexo char(1)
22 );
23
24 create table estudiante_Lima partition of estudiante for values in
25     ('Lima');
26 create table estudiante_Callao partition of estudiante for values
27     in ('Callao');
28
29 insert into estudiante (DNI, Nombre, Ciudad, Grupo, Promedio,
30     Edad, Sexo) values
```

```
28 ('0001', 'Selene Aguirre', 'Lima', 'A', 8.5, 17, 'F'),
29 ('0002', 'Martin Porres', 'Lima', 'C', 9, 23, 'M'),
30 ('0003', 'Miriam Gutierrez', 'Callao', 'A', 7, 21, 'F'),
31 ('0004', 'Benito Lopez', 'Callao', 'B', 10, 19, 'M');
```

P1. Creación dinámica de nuevos fragmentos

```
1 CREATE OR REPLACE FUNCTION update_estudiante()
2 RETURNS TRIGGER AS $$
3 DECLARE
4     nombre varchar;
5 BEGIN
6     IF NOT EXISTS (SELECT Ciudad FROM estudiante WHERE Ciudad =
7         quote_literal(NEW.Ciudad))
8     THEN
9         nombre := 'estudiante_' || NEW.Ciudad;
10        RAISE NOTICE '%', nombre;
11        EXECUTE 'create table ' || nombre || ' partition of
12            estudiante for values in (' ||
13            quote_literal(NEW.Ciudad) || ');';
14    END IF;
15    RAISE NOTICE '%', quote_literal(NEW);
16    EXECUTE 'insert into estudiante select (\$1).*' using NEW;
17    RETURN NULL;
18 END;
19 $$ LANGUAGE plpgsql;
20
21 CREATE OR REPLACE TRIGGER trigger_update_estudiante
22 BEFORE INSERT ON Aux
23 FOR EACH ROW
24 EXECUTE PROCEDURE update_estudiante();
25
26 insert into Aux (DNI, Nombre, Ciudad, Grupo, Promedio, Edad, Sexo)
27 values
28 ('0005', 'Selene Aguirre', 'Ayacucho', 'A', 8.5, 17, 'F'),
29 ('0006', 'Martin Porres', 'Junin', 'C', 9, 23, 'M'),
30 ('0007', 'Miriam Gutierrez', 'Arequipa', 'A', 7, 21, 'F'),
31 ('0008', 'Benito Lopez', 'Tacna', 'B', 10, 19, 'M');
32
33 select * from estudiante;
```

P2. Asignación

```
1  /*
2  =====
3  |      schemas      |
4  =====
5  */
6  -- set up schema locally
7  create schema local_schema;
8  set search_path to local_schema;
9  show search_path ;
10
11 -- execute for each remote server
12 create database remote_db; -- connect to remote_db afterwards
13 create schema remote_schema;
14
15 -- POSTGRES_FDW IS CASE SENSITIVE
16 drop table if exists estudiante;
17 create table estudiante (
18     DNI char(4) not null,
19     Nombre varchar not null,
20     Ciudad varchar(20) not null,
21     Grupo char(1),
22     Promedio float,
23     Edad smallint,
24     Sexo char(1)
25 )
26 partition by list (Ciudad);
27
28
29 SELECT * FROM pg_catalog.pg_tables where tablename like
30     '%estudiante%';
31
32 -- local
33 create table estudiante_callao partition of estudiante for values
34     in ('Callao');
35 create table estudiante_junin partition of estudiante for values
36     in ('Junin');
37
38 -- execute for remote servers and local server
39 create extension postgres_fdw;
40
41 /*
42 =====
43 |      Remote1      |
44 =====
45 */
```

```
43
44 drop server remote1 cascade;
45 create server remote1 foreign data wrapper postgres_fdw options
46     (host 'host.docker.internal', dbname 'remote_db', port
47         '5433'); -- using docker for local_db too
48
49 create user mapping for current_user
50     server remote1
51     options (user 'postgres', password 'ihavethepower');
52
53 import foreign schema remote_schema
54     from server remote1
55     into local_schema;
56
57 -- estudiante_Lima
58 create foreign table local_schema.estudiante_lima
59     partition of local_schema.estudiante for values in ('Lima')
60     server remote1
61     options (schema_name 'remote_schema', table_name
62         'estudiante_lima');
63
64 -- connect to remote1
65 drop table if exists estudiante_lima;
66 create table estudiante_lima (
67     DNI char(4) not null,
68     Nombre varchar not null,
69     Ciudad varchar(20) not null,
70     Grupo char(1),
71     Promedio float,
72     Edad smallint,
73     sexo char(1)
74 );
75
76 -- estudiante_Tacna
77 create foreign table local_schema.estudiante_tacna
78     partition of local_schema.estudiante for values in ('Tacna')
79     server remote1
80     options (schema_name 'remote_schema', table_name
81         'estudiante_tacna');
82
83 -- connect to remote1
84 drop table if exists estudiante_tacna;
85 create table estudiante_tacna (
86     DNI char(4) not null,
87     Nombre varchar not null,
88     Ciudad varchar(20) not null,
89     Grupo char(1),
```

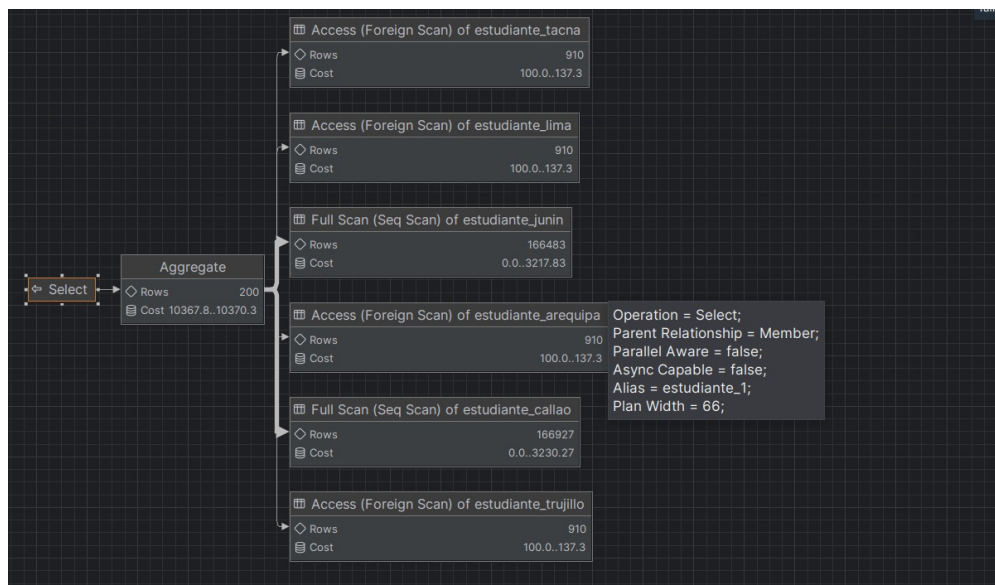
```
87     Promedio float,
88     Edad smallint,
89     Sexo char(1)
90 );
91
92 /*
93 =====
94 |         Remote2         |
95 =====
96 */
97
98 -- configure connection with remote2
99 drop server remote2 cascade;
100 create server remote2 foreign data wrapper postgres_fdw options
101     (host 'host.docker.internal', dbname 'remote_db', port
102       '5434'); -- using docker for local_db too
103
104 create user mapping for current_user
105     server remote2
106     options (user 'postgres', password 'ihavethepower');
107
108 import foreign schema remote_schema
109     from server remote2
110     into local_schema;
111
112 -- estudiante_Arequipa
113 create foreign table local_schema.estudiante_arequipa
114     partition of local_schema.estudiante for values in ('Arequipa')
115     server remote2
116     options (schema_name 'remote_schema', table_name
117             'estudiante_arequipa');
118
119 -- connect to remote2
120 drop table if exists estudiante_arequipa;
121 create table estudiante_arequipa (
122     DNI char(4) not null,
123     Nombre varchar not null,
124     Ciudad varchar(20) not null,
125     Grupo char(1),
126     Promedio float,
127     Edad smallint,
128     sexo char(1)
129 );
130
131 -- estudiante_Trujillo
132 create foreign table local_schema.estudiante_trujillo
133     partition of local_schema.estudiante for values in ('Trujillo')
```

```
132     server remote2
133     options (schema_name 'remote_schema', table_name
134             'estudiante_trujillo');
135
136 -- connect to remote2
137 drop table if exists estudiante_trujillo;
138 create table estudiante_trujillo (
139     DNI char(4) not null,
140     Nombre varchar not null,
141     Ciudad varchar(20) not null,
142     Grupo char(1),
143     Promedio float,
144     Edad smallint,
145     Sexo char(1)
146 );
147
148 /*
149 =====
150 |           Testing           |
151 =====
152 */
153
154 -- checking
155 select * from pg_foreign_server;
156
157 insert into local_schema.estudiante (DNI, Nombre, Ciudad, Grupo,
158     Promedio, Edad, Sexo) values
159     ('0001', 'Selene Aguirre', 'Lima', 'A', 8.5, 17, 'F'),
160     ('0002', 'Martin Porres', 'Lima', 'C', 9, 23, 'M'),
161     ('0003', 'Miriam Gutierrez', 'Callao', 'A', 7, 21, 'F'),
162     ('0004', 'Benito Lopez', 'Callao', 'B', 10, 19, 'M'),
163     ('0006', 'Lisa Porres', 'Junin', 'B', 9, 24, 'M'),
164     ('0008', 'Gabriel Lopez', 'Tacna', 'A', 12, 19, 'M');
165
166 explain analyse
167 select * from estudiante;
```

P2. Consultas

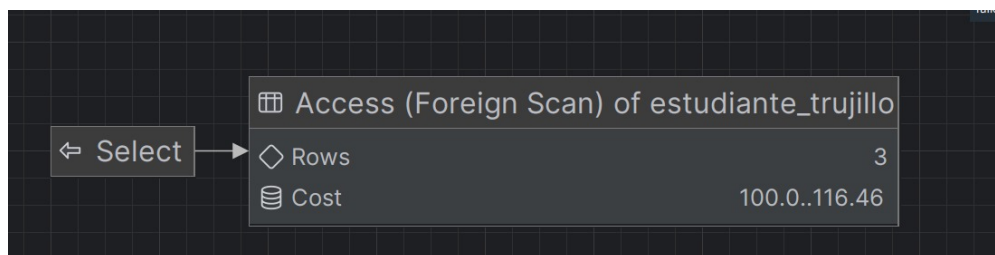
Consulta 1

```
1 SELECT Ciudad, AVG(Promedio) as Promedio_Ciudad
2 FROM estudiante
3 GROUP BY Ciudad;
```



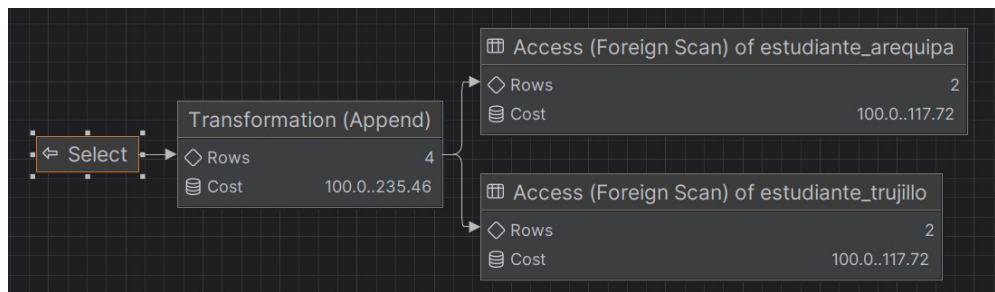
Consulta 2

```
1 SELECT * FROM estudiante
2 WHERE Ciudad = 'Trujillo'
```



Consulta 3

```
1 SELECT * FROM estudiante
2 WHERE Ciudad in ('Arequipa' , 'Trujillo') AND Promedio > 5.0;
```

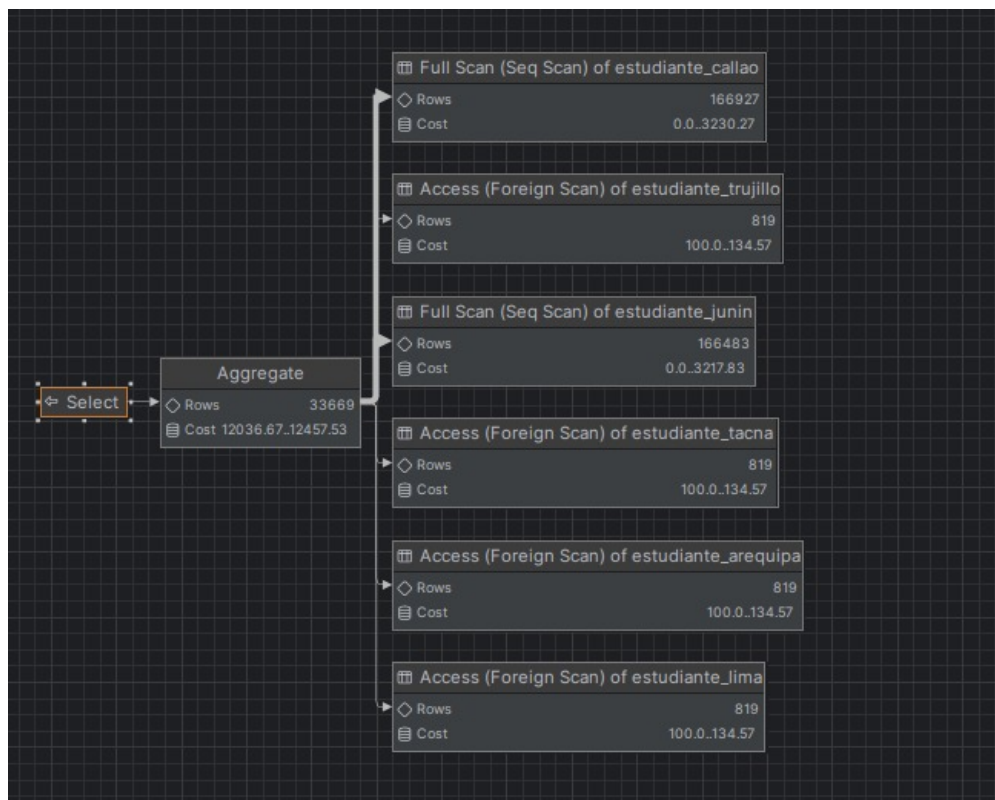


Consulta 4

```

1 SELECT Ciudad, Grupo, AVG(Promedio) AS Promedio_Grupo, MIN(Edad)
   AS Edad_Minima
2 FROM Estudiante
3 GROUP BY Ciudad, Grupo;

```



Consulta 5

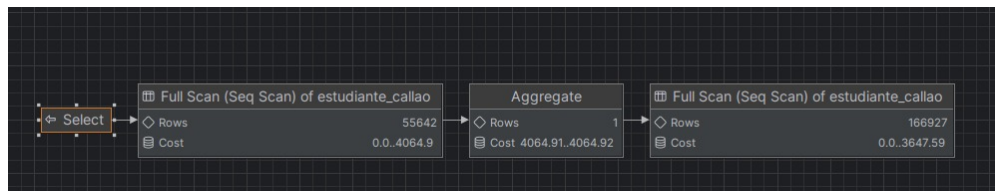
```

1 SELECT DNI, Nombre, Ciudad, Promedio
2 FROM Estudiante AS e WHERE e.Ciudad = 'Callao'
3 AND e.Promedio > (
4     SELECT AVG(Promedio)
5     FROM Estudiante

```



```
6 WHERE Ciudad = 'Callao'
7 );
```



Consulta 6

```
1 explain analyse
2 select * from estudiante;
```

