Lab 12

Base table

```
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)
);
```

P1. PARTITION BY LIST

```
CREATE TABLE employees1 (
    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);

CREATE TABLE dep50 PARTITION OF employees1
FOR VALUES IN (50);

CREATE TABLE dep80 PARTITION OF employees1
FOR VALUES IN (80);

CREATE TABLE dep100 PARTITION OF employees1
FOR VALUES IN (100);
```

Default:

Employees	50	80	100
Planning time (ms)	0.111	0.583	0.163
Execution time (ms)	0.139	0.185	0.212

With list partition:

Employees1	50	80	100
Planning time (ms)	0.284	0.773	0.732
Execution time (ms)	0.111	0.121	0.110

P2. PARTITION BY RANGE

```
CREATE TABLE employees2 (
    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 RANGE (date_part('year', hire_date));

CREATE TABLE hirelow PARTITION OF employees2
FOR VALUES FROM (MINVALUE) TO (1995);

CREATE TABLE hiremid PARTITION OF employees2
FOR VALUES FROM (1995) TO (1998);

CREATE TABLE hirehigh PARTITION OF employees2
FOR VALUES FROM (1998) TO (MAXVALUE);

CREATE INDEX hirelow_ix ON hirelow (hire_date);

CREATE INDEX hiremid_ix ON hiremid (hire_date);

CREATE INDEX hirehigh_ix ON hirehigh (hire_date);
```

Query 1:

```
SELECT * FROM employees2 WHERE hire_date >= '1995-05-03'::date;
```

Query 2:

```
SELECT * FROM employees2 WHERE hire_date >= '1997-05-03'::date;
```

Query 3:

```
SELECT * FROM employees2 WHERE hire_date >= '2000-05-03'::date;
```

Employees	Q1	Q2	Q3
Planning time (ms)	0.186	0.175	0.161
Execution time (ms)	0.194	0.174	0.169

Employees2	Q1	Q2	Q3
Planning time (ms)	0.755	0.797	0.500
Execution time (ms)	0.295	0.274	0.132

P3. Two Attributes

Script doesn't work quite well.

```
CREATE TABLE employees3 (
    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 RANGE (date_part('year', hire_date), salary);

CREATE TABLE datelow_salarylow PARTITION OF employees3
FOR VALUES FROM (MINVALUE,
MINVALUE) TO (1995, 30000);

CREATE TABLE datelow_salarymid PARTITION OF employees3
FOR VALUES FROM (MINVALUE, 30000) TO (1995, 70000);
```

```
CREATE TABLE datelow_salaryhigh PARTITION OF employees3
FOR VALUES FROM (MINVALUE, 70000) TO (1995,
MAXVALUE);

CREATE TABLE datemid_salarylow PARTITION OF employees3
FOR VALUES FROM (1995,
MINVALUE) TO (1998, 30000);

CREATE TABLE datemid_salarymid PARTITION OF employees3
FOR VALUES FROM (1995, 30000) TO (1998, 70000);

CREATE TABLE datemid_salaryhigh PARTITION OF employees3
FOR VALUES FROM (1995, 70000) TO (1998,
MAXVALUE);

CREATE TABLE datehigh_salarylow PARTITION OF employees3
FOR VALUES FROM (1998,
MINVALUE) TO (MAXVALUE, 30000);

CREATE TABLE datehigh_salarymid PARTITION OF employees3
FOR VALUES FROM (1998, 30000) TO (MAXVALUE, 70000);

CREATE TABLE datehigh_salaryhigh PARTITION OF employees3
FOR VALUES FROM (1998, 70000) TO (MAXVALUE,
MAXVALUE);
```

Students

- Andrea Díaz - 201720031
- Diego Cánez - 201710319