Universidad Nacional Autónoma de México.

Facultad de ingeniería.

Bases de Datos

Serie 3. Consultas básicas y funciones de agregación

Fecha de realización: 22 de mayo 2025.

Fecha de Entrega: 23 de mayo 2025.

Tavera Castillo David Emmanuel

1. Editar el archivo script bd2.sql teniendo en cuenta las siguientes consideraciones:

```
ALTER TABLE cliente ADD CONSTRAINT pk_cliente PRIMARY KEY (id_cliente);
ALTER TABLE cliente ALTER COLUMN estado SET DEFAULT 'CDMX';

ALTER TABLE articulo ADD CONSTRAINT pk_articulo PRIMARY KEY (num_articulo);
ALTER TABLE articulo ADD CONSTRAINT ch_precio CHECK (precio>=0);

ALTER TABLE orden ADD CONSTRAINT pk_orden PRIMARY KEY (id_orden);
ALTER TABLE orden ALTER COLUMN fecha SET DEFAULT CURRENT_DATE;
ALTER TABLE orden ADD CONSTRAINT fk_orden_cliente FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente);
```

2. Indicar las ciudades que tienen más de un aeropuerto.

```
     city      | count |              dia              | usuario
---------------+-------+-------------------------------+----------
 Jackson       |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Springfield   |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Albany        |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Columbia      |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 New York      |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Jacksonville  |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Wilmington    |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 San Diego     |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Chicago       |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Houston       |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Charleston    |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Portland      |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Columbus      |     2 | 2025-05-22 20:32:45.550999-06 | postgres
 Rochester     |     2 | 2025-05-22 20:32:45.550999-06 | postgres
(14 rows)


Time: 3.258 ms
```

3. Nombre de las aerolíneas que no terminan en Inc. ni en Co sin usar operadores and y or

```
registro_vuelos=#
registro_vuelos=# SELECT airline, now() AS dia, current_user AS usuario FROM aerolineas
registro_vuelos-# WHERE airline NOT LIKE '%Inc.'
registro_vuelos-# INTERSECT
registro_vuelos-# SELECT airline, now() AS dia, current_user AS usuario FROM aerolineas
registro_vuelos-# WHERE airline NOT LIKE '%Co.';
          airline          |              dia              | usuario
---------------------------+-------------------------------+----------
 Virgin America            | 2025-05-22 20:39:48.732787-06 | postgres
 Spirit Air Lines          | 2025-05-22 20:39:48.732787-06 | postgres
 Atlantic Southeast Airlines | 2025-05-22 20:39:48.732787-06 | postgres
 JetBlue Airways           | 2025-05-22 20:39:48.732787-06 | postgres
(4 rows)
```

4. Indicar los nombres de los aeropuertos que estuvieron implicados en el vuelo que presentó el mayor retraso de llegada

```
registro_vuelos=# SELECT origin_airport, destination_airport,arrivale_delay, now() AS dia, current_user AS usuario
registro_vuelos-# FROM vuelos
registro_vuelos-# WHERE arrivale_delay IS NOT NULL
registro_vuelos-# ORDER BY arrivale_delay DESC
registro_vuelos-# LIMIT 1;
 origin_airport | destination_airport | arrivale_delay |              dia              | usuario
----------------+---------------------+----------------+-------------------------------+----------
 BHM            | DFW                 |           1971 | 2025-05-22 20:43:41.845176-06 | postgres
(1 row)


Time: 1340.786 ms (00:01.341)
```

5. Mostrar aquella categoría (tabla artículo) que tiene el precio mínimo. La información debe estar agrupada (Implica que la consulta no sale con solo selects y wheres).

```
datos_clase=# SELECT categoria, MIN(precio) AS precio, now() AS dia, current_user AS usuario FROM articulo
datos_clase-# GROUP BY categoria, precio
datos_clase-# ORDER BY precio
datos_clase-# LIMIT 1;
 categoria  | precio |              dia              | usuario
------------+--------+-------------------------------+----------
 accesorios |    120 | 2025-05-22 21:35:43.019839-06 | postgres
(1 row)
```

6. Se desea conocer el nombre de aquellas aerolíneas cuyo segundo carácter del iata code termina en X o 9. Debe incluirse una columna que muestre dicha terminación.

```
registro_vuelos=# SELECT       airline,SUBSTR(iata_code, 2, 2) AS terminacion, now() AS dia, current_user AS usuario
registro_vuelos-# FROM aerolineas
registro_vuelos-# WHERE iata_code LIKE '%X' OR iata_code LIKE '%9';
       airline         | terminacion |              dia              | usuario
-----------------------+-------------+-------------------------------+----------
 Frontier Airlines Inc. | 9           | 2025-05-22 20:45:13.140988-06 | postgres
 Virgin America         | X           | 2025-05-22 20:45:13.140988-06 | postgres
 Frontier Airlines Inc. | 9           | 2025-05-22 20:45:13.140988-06 | postgres
 Virgin America         | X           | 2025-05-22 20:45:13.140988-06 | postgres
(4 rows)
```

7. Proporcionar el nombre de los aeropuertos cuya latitud se encuentre entre 40 y 41, y su longitud sea menor que el promedio de la longitud. Nota: el promedio se toma de aquellas observaciones cuya latitud se encuentre entre 40 y 41.

```
registro_vuelos=# SELECT airport, now() AS dia, current_user AS usuario
registro_vuelos-# FROM aeropuertos
registro_vuelos-# WHERE latitude BETWEEN 40 AND 41
registro_vuelos-#   AND longitude < (
registro_vuelos(#       SELECT AVG(longitude)
registro_vuelos(#       FROM aeropuertos
registro_vuelos(#       WHERE latitude BETWEEN 40 AND 41
registro_vuelos(#   );
                  airport                  |              dia              | usuario
-------------------------------------------+-------------------------------+----------
 Arcata Airport                            | 2025-05-22 20:57:01.219704-06 | postgres
 Elko Regional Airport                     | 2025-05-22 20:57:01.219704-06 | postgres
 Central Nebraska Regional Airport         | 2025-05-22 20:57:01.219704-06 | postgres
 Yampa Valley Airportá(Yampa Valley Regional) | 2025-05-22 20:57:01.219704-06 | postgres
 Lincoln Airportá(Lincoln Municipal)       | 2025-05-22 20:57:01.219704-06 | postgres
 Redding Municipal Airport                 | 2025-05-22 20:57:01.219704-06 | postgres
 Salt Lake City International Airport      | 2025-05-22 20:57:01.219704-06 | postgres
 Valdez Airport                            | 2025-05-22 20:57:01.219704-06 | postgres
(8 rows)
```

8. ¿Cuántos aviones por aerolínea y día, fueron cancelados saliendo del aeropuerto de Honolulu?

```
registro_vuelos=# SELECT airline, day,month,year, COUNT (*) AS cancelados, now() AS dia, current_user AS usuario
registro_vuelos-# FROM vuelos v
registro_vuelos-# INNER JOIN aeropuertos a ON v.origin_airport = a.iata_code
registro_vuelos-# WHERE airport Like '%Honolulu%' AND cancelled = '1'
registro_vuelos-# GROUP BY airline, day,month,year;
 airline | day | month | year | cancelados |              dia              | usuario
---------+-----+-------+------+------------+-------------------------------+----------
 AA      |   2 |     6 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |   7 |     3 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |   7 |     7 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |   8 |     8 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |   9 |     5 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |   9 |     7 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |   9 |     8 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  10 |     4 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  11 |     7 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  13 |     6 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  15 |     3 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  15 |     4 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  15 |     8 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  18 |     5 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  20 |     6 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 AA      |  21 |     7 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 UA      |  30 |     5 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 UA      |  30 |     7 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 UA      |  30 |     9 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 UA      |  30 |    12 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 UA      |  31 |     7 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |   1 |     2 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |   3 |     3 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |   5 |     1 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |   6 |     3 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |   8 |     3 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |  17 |     6 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |  18 |     1 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |  23 |     5 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 US      |  24 |     5 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
 VX      |   2 |    12 | 2015 |          1 | 2025-05-22 21:06:11.011457-06 | postgres
(145 rows)

Time: 701.836 ms
```

9. Hacer un cross join entre la tabla cliente y la tabla aerolíneas. Obviamente ambas tablas forman parte de distintas BDs, debe encontrar la forma de hacerlo.

```
datos_clase=# SELECT c.nombre AS cliente, a.airline AS aerolinea, now() AS dia, current_user AS usuario
datos_clase-# FROM cliente c
datos_clase-# CROSS JOIN aerolineas a;
                  cliente                  |          aerolinea          |              dia              | usuario
-------------------------------------------+-----------------------------+-------------------------------+----------
 Luisa                                     | United Air Lines Inc.       | 2025-05-22 21:45:42.818059-06 | postgres
 mario                                     | United Air Lines Inc.       | 2025-05-22 21:45:42.818059-06 | postgres
 Jaime                                     | United Air Lines Inc.       | 2025-05-22 21:45:42.818059-06 | postgres
 Angela                                    | United Air Lines Inc.       | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | United Air Lines Inc.       | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | United Air Lines Inc.       | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | United Air Lines Inc.       | 2025-05-22 21:45:42.818059-06 | postgres
 Luisa                                     | American Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 mario                                     | American Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Jaime                                     | American Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Angela                                    | American Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | American Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | American Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | American Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luisa                                     | US Airways Inc.             | 2025-05-22 21:45:42.818059-06 | postgres
 mario                                     | US Airways Inc.             | 2025-05-22 21:45:42.818059-06 | postgres
 Jaime                                     | US Airways Inc.             | 2025-05-22 21:45:42.818059-06 | postgres
 Angela                                    | US Airways Inc.             | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | US Airways Inc.             | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | US Airways Inc.             | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | US Airways Inc.             | 2025-05-22 21:45:42.818059-06 | postgres
 Luisa                                     | Frontier Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 mario                                     | Frontier Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Jaime                                     | Frontier Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Angela                                    | Frontier Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | Frontier Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | Frontier Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | Frontier Airlines Inc.      | 2025-05-22 21:45:42.818059-06 | postgres
 Luisa                                     | JetBlue Airways             | 2025-05-22 21:45:42.818059-06 | postgres
 mario                                     | American Eagle Airlines Inc.| 2025-05-22 21:45:42.818059-06 | postgres
 Jaime                                     | American Eagle Airlines Inc.| 2025-05-22 21:45:42.818059-06 | postgres
 Angela                                    | American Eagle Airlines Inc.| 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | American Eagle Airlines Inc.| 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | American Eagle Airlines Inc.| 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | American Eagle Airlines Inc.| 2025-05-22 21:45:42.818059-06 | postgres
 Luisa                                     | Virgin America              | 2025-05-22 21:45:42.818059-06 | postgres
 mario                                     | Virgin America              | 2025-05-22 21:45:42.818059-06 | postgres
 Jaime                                     | Virgin America              | 2025-05-22 21:45:42.818059-06 | postgres
 Angela                                    | Virgin America              | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | Virgin America              | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | Virgin America              | 2025-05-22 21:45:42.818059-06 | postgres
 Luis                                      | Virgin America              | 2025-05-22 21:45:42.818059-06 | postgres
(196 rows)

Time: 1.625 ms
```

10. Cantidad de vuelos cancelados por día

```
registro_vuelos=# SELECT day,month,year, COUNT (*) AS cancelados, now() AS dia, current_user AS usuario
registro_vuelos-# FROM vuelos v
registro_vuelos-# WHERE cancelled = '1'
registro_vuelos-# GROUP BY  day,month,year;
 day | month | year | cancelados |             dia               | usuario
-----+-------+------+------------+-------------------------------+----------
   1 |     1 | 2015 |        466 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     2 | 2015 |       1979 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     3 | 2015 |       1514 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     4 | 2015 |         88 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     5 | 2015 |         56 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     6 | 2015 |        552 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     7 | 2015 |        217 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     8 | 2015 |        164 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |     9 | 2015 |         78 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |    10 | 2015 |         42 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |    11 | 2015 |         86 | 2025-05-22 21:08:42.774858-06 | postgres
   1 |    12 | 2015 |         78 | 2025-05-22 21:08:42.774858-06 | postgres
  30 |    11 | 2015 |         49 | 2025-05-22 21:08:42.774858-06 | postgres
  30 |    12 | 2015 |        281 | 2025-05-22 21:08:42.774858-06 | postgres
  31 |     1 | 2015 |        130 | 2025-05-22 21:08:42.774858-06 | postgres
  31 |     3 | 2015 |         75 | 2025-05-22 21:08:42.774858-06 | postgres
  31 |     5 | 2015 |        384 | 2025-05-22 21:08:42.774858-06 | postgres
  31 |     7 | 2015 |        150 | 2025-05-22 21:08:42.774858-06 | postgres
  31 |     8 | 2015 |        144 | 2025-05-22 21:08:42.774858-06 | postgres
  31 |    10 | 2015 |        178 | 2025-05-22 21:08:42.774858-06 | postgres
  31 |    12 | 2015 |         42 | 2025-05-22 21:08:42.774858-06 | postgres
(365 rows)


Time: 721.703 ms
```

11. Seleccionar el nombre de los aeropuertos cuya segunda letra del iata code sea K o X, sin usar operadores and, not u or. Puede usar alguna función propia de postgres.

```
registro_vuelos=# SELECT        airline, now() AS dia, current_user AS usuario
registro_vuelos-# FROM aerolineas
registro_vuelos-# WHERE SUBSTR(iata_code, 2, 2) = 'K'
registro_vuelos-# UNION
registro_vuelos-# SELECT        airline, now() AS dia, current_user AS usuario
registro_vuelos-# FROM aerolineas
registro_vuelos-# WHERE SUBSTR(iata_code, 2, 2) = 'X';
     airline      |             dia               | usuario
------------------+-------------------------------+----------
 Spirit Air Lines | 2025-05-22 21:14:27.76794-06 | postgres
 Virgin America   | 2025-05-22 21:14:27.76794-06 | postgres
(2 rows)
```

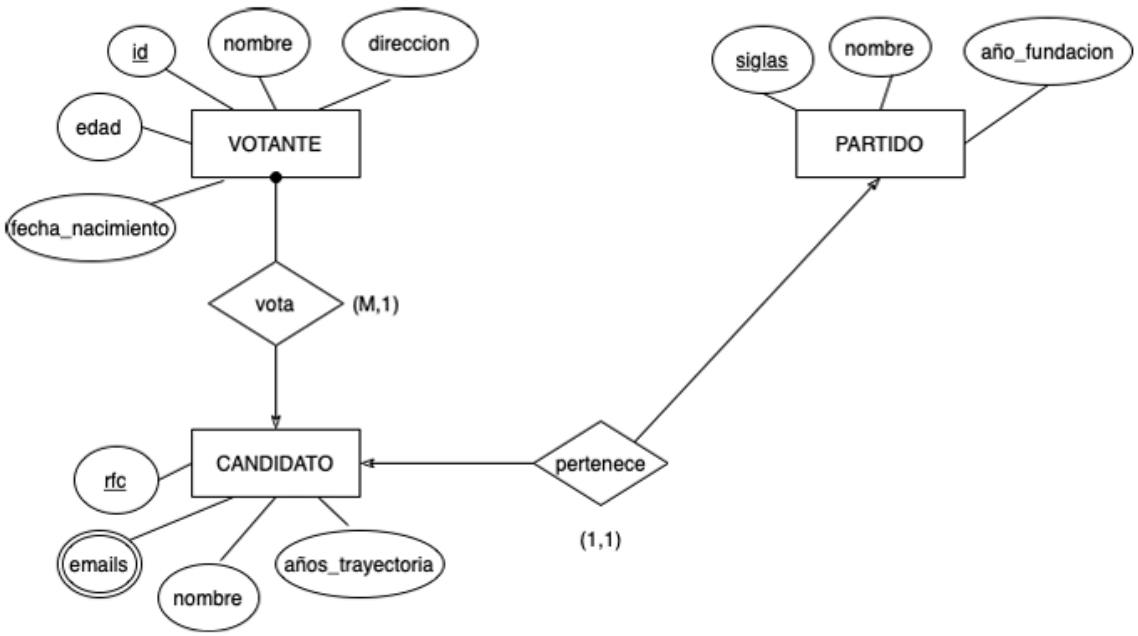
12. Indicar el nombre(s) de la aerolínea cuya distancia de vuelo es la mayor.

```
registro_vuelos=# SELECT a.airline, distance, now() AS dia, current_user AS usuario
registro_vuelos-# FROM vuelos v
registro_vuelos-# INNER JOIN aerolineas a
registro_vuelos-# ON v.airline = a.iata_code
registro_vuelos-# ORDER BY distance DESC
registro_vuelos-# LIMIT 1;
        airline         | distance |              dia              | usuario
------------------------+----------+-------------------------------+----------
 Hawaiian Airlines Inc. |     4983 | 2025-05-22 21:20:02.731314-06 | postgres
(1 row)


Time: 1316.331 ms (00:01.316)
```

13. Indicar el nombre del aeropuerto de origen donde se presentó el mayor tiempo de vuelo

```
registro_vuelos=# SELECT a.airport, v.air_time, now() AS dia, current_user AS usuario
registro_vuelos-# FROM vuelos v
registro_vuelos-# INNER JOIN aeropuertos a
registro_vuelos-# ON v.origin_airport = a.iata_code
registro_vuelos-# WHERE air_time IS NOT NULL
registro_vuelos-# ORDER BY air_time DESC
registro_vuelos-# LIMIT 1;
                          airport                           | air_time |             dia              | usuario
------------------------------------------------------------+----------+------------------------------+----------
 John F. Kennedy International Airportá(New York International Airport) |      690 | 2025-05-22 21:28:29.049929-06 | postgres
(1 row)


Time: 1754.920 ms (00:01.755)
```

14. Partiendo del siguiente MER



(a) Generar el mapeo a la representación intermedia de MR

VOTANTE:
{ id int (PK),
nombre varchar(100),
direccion varchar(150),
edad int,
fecha_nacimiento date
rfc varchar(13) (FK) }
PARTIDO:
{ siglas varchar(10) (PK),
nombre varchar(100),
año_fundacion int }

CANDIDATO:
{ rfc varchar(13) (PK),
nombre varchar(100),
años_trayectoria int,
siglas_partido varchar(10) (FK) }

EMAIL:
{[rfc varchar(13) (FK),
emails varchar(150)](PK)}

(B)Generar el DDL de las relaciones obtenidas en el punto anterior. Para las FK debe establecerse la restricción cascade para las operaciones de borrado y actualización.

```
CREATE TABLE VOTANTE (
  id INT PRIMARY KEY,
  nombre VARCHAR(100),
  direccion VARCHAR(150),
  edad INT,
  fecha_nacimiento DATE,
  rfc VARCHAR(13),
  CONSTAINT fk_votante_candidato FOREIGN KEY (rfc)
  REFERENCES candidato(rfc) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE PARTIDO (
  siglas VARCHAR(10) PRIMARY KEY,
  nombre VARCHAR(100),
  año_fundacion INT
);

CREATE TABLE CANDIDATO (
  rfc VARCHAR(13) PRIMARY KEY,
  emails VARCHAR(150),
  nombre VARCHAR(100),
  años_trayectoria INT,
  siglas_partido VARCHAR(10),
  CONSTAINT fk_candidato_partido FOREIGN KEY (siglas_partido)
  REFERENCES PARTIDO(siglas) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE EMAIL (
  rfc VARCHAR(13),
  emails VARCHAR(150),
  CONSTRAINT pk_email PRIMARY KEY (rfc, emails),
  CONSTRAINT fk_email_candidato FOREIGN KEY (rfc)
  REFERENCES CANDIDATO(rfc) ON DELETE CASCADE ON UPDATE CASCADE
);
```

15. Genere una función o procedure que resuelva lo siguiente:
- Se pasará como parámetro el nombre de una categoría
- Se deberá seleccionar el nombre de articulo y precio de todos los registros que pertenezcan a la categoría pasada como parámetro
- Se deberá iterar sobre los n registros que forman parte de la categoría (Se sugiere ampliamente el uso de cursores)
- Además del nombre y precio, se debe incluir para cada registro, el precio mínimo y máximo de la categoría en cuestión
- Se debe imprimir los n registros pertenecientes a la categoría

```
DROP FUNCTION articulos_por_categoria;
CREATE OR REPLACE FUNCTION articulos_por_categoria(cat_nombre VARCHAR)
RETURNS TABLE (
  nombre_articulo VARCHAR,
  precio integer,
  precio_min_categoria integer,
  precio_max_categoria integer
) AS $$
DECLARE
  v_precio_min integer;
  v_precio_max integer;
BEGIN
  SELECT MIN(a.precio), MAX(a.precio)
  INTO v_precio_min, v_precio_max
  FROM articulo a
  WHERE a.categoria = cat_nombre;

  RETURN QUERY
```

```
  SELECT
    a.nombre_articulo,
    a.precio,
    v_precio_min,
    v_precio_max
  FROM articulo a
  WHERE a.categoria = cat_nombre;
END;
$$ LANGUAGE plpgsql;
```

```
datos_clase=# SELECT * FROM articulos_por_categoria('accesorios');
 nombre_articulo | precio | precio_min_categoria | precio_max_categoria
-----------------+--------+----------------------+----------------------
 gorra           |    120 |                  120 |                  600
 lentes          |    400 |                  120 |                  600
 bolsa           |    300 |                  120 |                  600
 mochila         |    570 |                  120 |                  600
 Reloj           |    600 |                  120 |                  600
(5 rows)


Time: 1.405 ms
```