

FLIGHT MANAGEMENT SYSTEM (FMS)

JULIO CÉSAR LOZANO LOZANO

TUTOR: JESÚS ARIEL GONZALEZ BONILLA

CORPORACIÓN UNIVERSITARIA DEL HUILA CORHUILA

MODELADO Y GESTIÓN DE BASE DE DATOS

OCTUBRE 2025

Estructura del Modelo

Modulo Geolocation

Entidad	Atributos	Notas del estudiante
continent	id, name	Contiene los continentes donde opera el sistema.
country	id, name, continent_id	Cada país pertenece a un continente.
state	id, name, country_id	Cada estado pertenece a un país.
city	id, name, state_id	Cada ciudad pertenece a un estado.

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY de cada entidad es **Id**.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q deben definirse con NOT NULL son name, continente_id, country_id, state_id.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

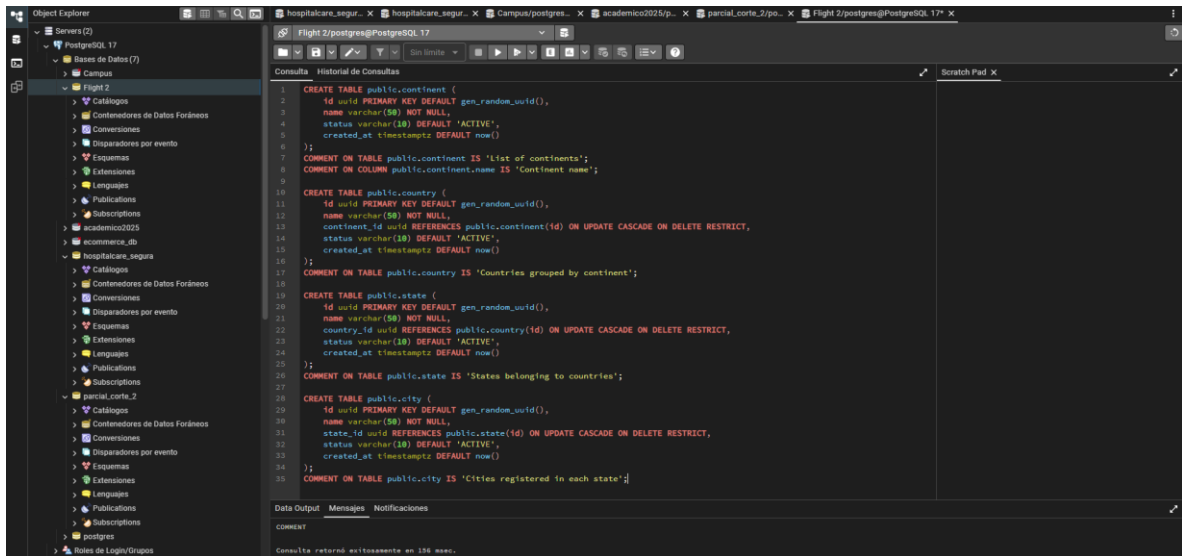
Rta: 1-N (continent→country→state-city)

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: FOREIGN KEY = fk_country_continent, fk_state_country, fk_city_state

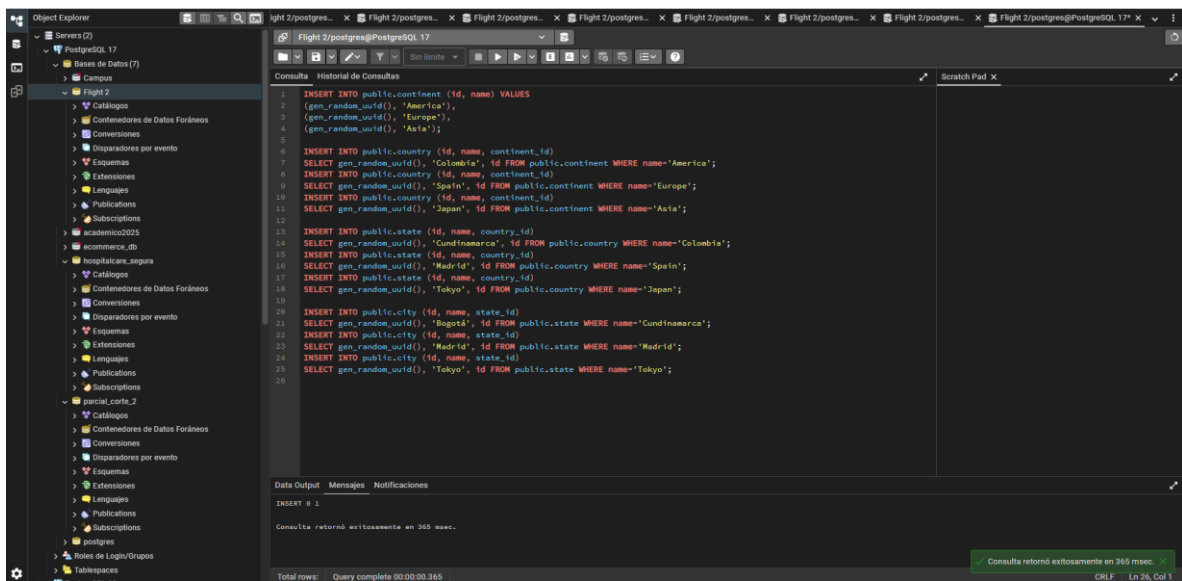
Prácticas DDL

Creación de tablas.



Prácticas DML

Insertar registros en cada Entidad.



Módulo Infrastructure

Entidad	Atributos	Notas del estudiante
airport	id, name, code, city_id	Cada aeropuerto pertenece a una ciudad.
terminal	id, name, airport_id	Cada aeropuerto puede tener varias terminales.

boarding_gate	id, code, terminal_id	Cada terminal tiene múltiples puertas de embarque.
----------------------	-----------------------	--

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **Id**.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q deben definirse con NOT NULL son name, code, city_id, airport_id, terminal_id

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

Rta: 1-N (airport → terminal → boarding_gate).

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: FOREIGN KEY = fk_airport_city, fk_terminal_airport, fk_boardinggate_terminal.

Prácticas DDL

Creación de tablas.

```

1 CREATE TABLE public.airport (
2   id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
3   code varchar(18) NOT NULL UNIQUE,
4   name varchar(180) NOT NULL,
5   address text,
6   city_id uuid REFERENCES public.city(id) ON UPDATE CASCADE ON DELETE RESTRICT,
7   created_at timestampz DEFAULT now(),
8   created_by uuid,
9   status varchar(18) DEFAULT 'ACTIVE'
10 );
11 COMMENT ON TABLE public.airport IS 'Registered airports in the system';
12
13 CREATE TABLE public.terminal (
14   id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
15   name varchar(80) NOT NULL,
16   airport_id uuid REFERENCES public.airport(id) ON UPDATE CASCADE ON DELETE RESTRICT,
17   status varchar(18) DEFAULT 'ACTIVE'
18 );
19 COMMENT ON TABLE public.terminal IS 'Airport terminals';
20
21 CREATE TABLE public.boarding_gate (
22   id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
23   code varchar(18) NOT NULL,
24   terminal_id uuid REFERENCES public.terminal(id) ON UPDATE CASCADE ON DELETE RESTRICT,
25   status varchar(18) DEFAULT 'ACTIVE'
26 );
27 COMMENT ON TABLE public.boarding_gate IS 'Boarding gates in terminals';
28

```

Data Output: Messages: Notificaciones

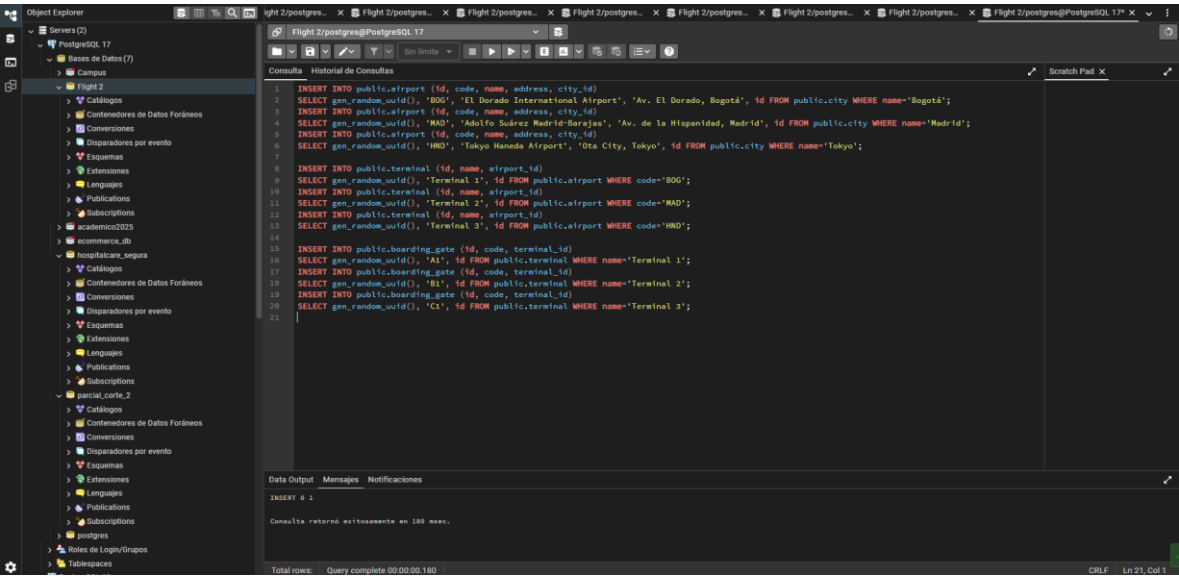
COMMENT

Consulta retornó exitosamente en 155 msec.

Total rows: Query completo 00:00:00.155

Prácticas DML

Insertar registros en cada Entidad.



Módulo Parameterization

Entidad	Atributos	Notas del estudiante
document_type	id, name	Tipos de documento (Passport, ID Card, etc.).
flight_type	id, name	Tipos de vuelo (Domestic, International, Charter).
aircraft_type	id, name, capacity	Tipo y capacidad de la aeronave.
crew_role	id, name	Rol de cada tripulante (Pilot, Flight Attendant, etc.).
ticket_class	id, name	Clase de ticket (Economy, Business, First Class).

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **Id**.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: El atributo q debe definirse con NOT NULL es name.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

Rta: No se presenta ninguna relación con ninguna tabla.

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: En este módulo no se aplican llaves foráneas.

Prácticas DDL

Creación de tablas.

```

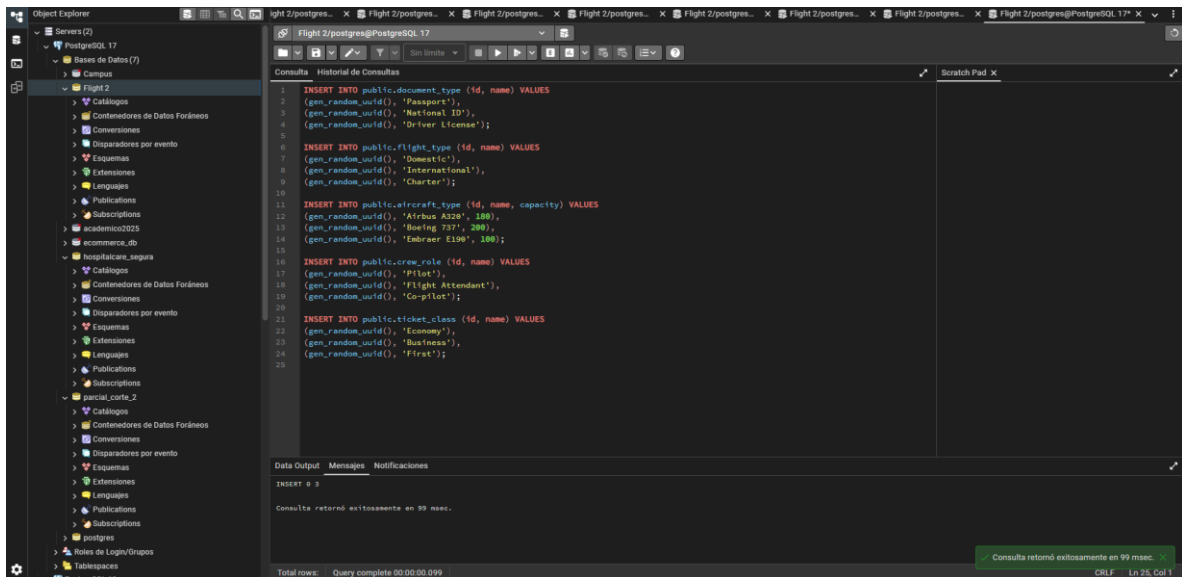
1 CREATE TABLE public.document_type (
2     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
3     name varchar(50) NOT NULL
4 );
5 COMMENT ON TABLE public.document_type IS 'Types of identity documents';
6
7 CREATE TABLE public.flight_type (
8     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
9     name varchar(50) NOT NULL
10 );
11 COMMENT ON TABLE public.flight_type IS 'Types of flights (Domestic, International)';
12
13 CREATE TABLE public.aircraft_type (
14     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
15     name varchar(50) NOT NULL,
16     capacity int,
17     status varchar(10) DEFAULT 'ACTIVE'
18 );
19 COMMENT ON TABLE public.aircraft_type IS 'Aircraft models and capacity';
20
21 CREATE TABLE public.crew_role (
22     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
23     name varchar(50) NOT NULL
24 );
25 COMMENT ON TABLE public.crew_role IS 'Crew roles (Pilot, Attendant)';
26
27 CREATE TABLE public.ticket_class (
28     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
29     name varchar(50) NOT NULL
30 );
31 COMMENT ON TABLE public.ticket_class IS 'Ticket class (Economy, Business)';
32

```

Query complete 00:00:00.001

Prácticas DML

Insertar registros en cada Entidad.



Módulo Security

Entidad	Atributos	Notas del estudiante
role	id, name, description	Define roles del sistema (Admin, User).
permission	id, name, description	Permisos específicos (Create Flight, Delete User).
role_permission	role_id, permission_id	Relación muchos a muchos entre roles y permisos.
module	id, name, description	Módulos funcionales del sistema.
form	id, name, url, description	Formularios o pantallas del sistema.
form_module	form_id, module_id	Relación entre formularios y módulos.
role_form	role_id, form_id	Define qué formularios puede acceder cada rol.
person	id, document_type_id, document_number, first_name, last_name, email, phone	Personas registradas en el sistema.
user_account	id, person_id, username, password, role_id, is_active, created_at	Cuenta de usuario vinculada a persona y rol.

session	id, user_account_id, login_time, logout_time, ip_address	Registro de sesiones de usuarios.
audit_log	id, user_account_id, action, action_date, description	Bitácora de auditoría.
password_reset	id, user_account_id, reset_token, created_at, expires_at, used	Control de recuperación de contraseñas.
user_preference	id, user_account_id, preference_key, preference_value	Preferencias personales (tema, idioma, notificaciones).
security_question	id, question_text	Preguntas de seguridad.
user_security_answer	id, user_account_id, security_question_id, answer	Respuestas de seguridad de cada usuario.

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **Id**, excepto las q son compuestas.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q debe definirse con NOT NULL son username, password, role_id, person_id, question_text, answer.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

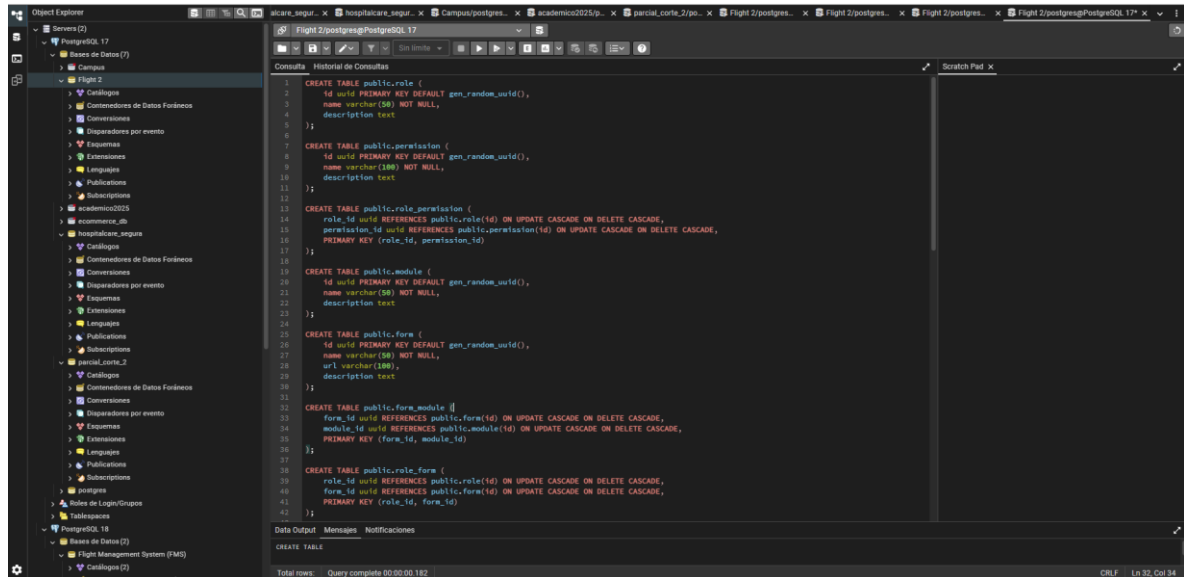
Rta: 1-N (role → user_account) y N-M (role ↔ permission, role ↔ form).

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

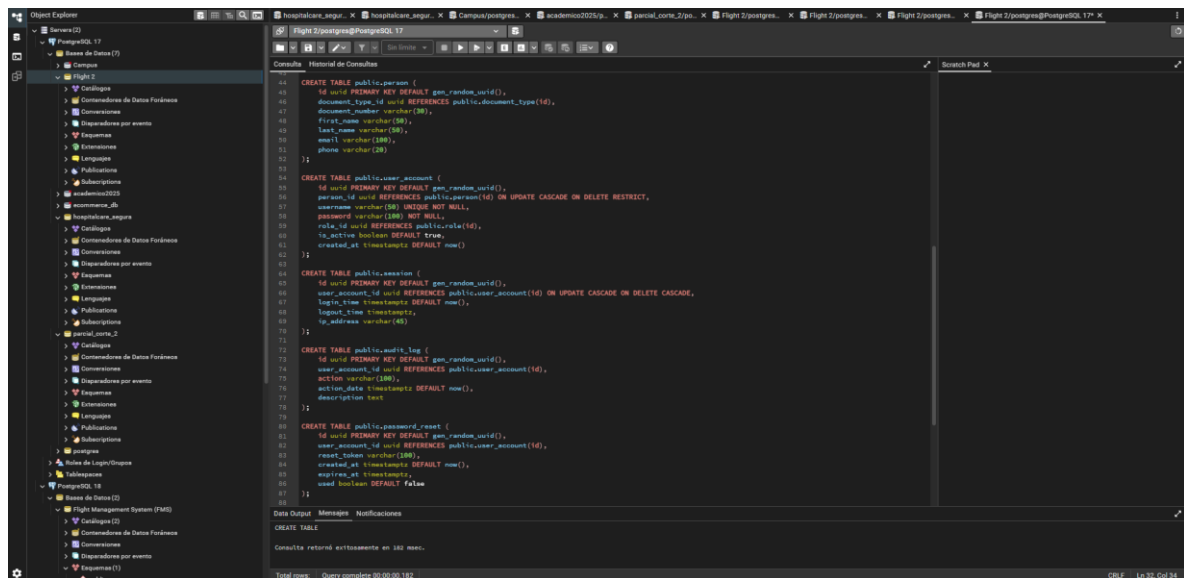
Rta: FOREIGN KEY = fk_useraccount_person, fk_useraccount_role, fk_rolepermission_role, fk_formmodule_module.

Prácticas DDL

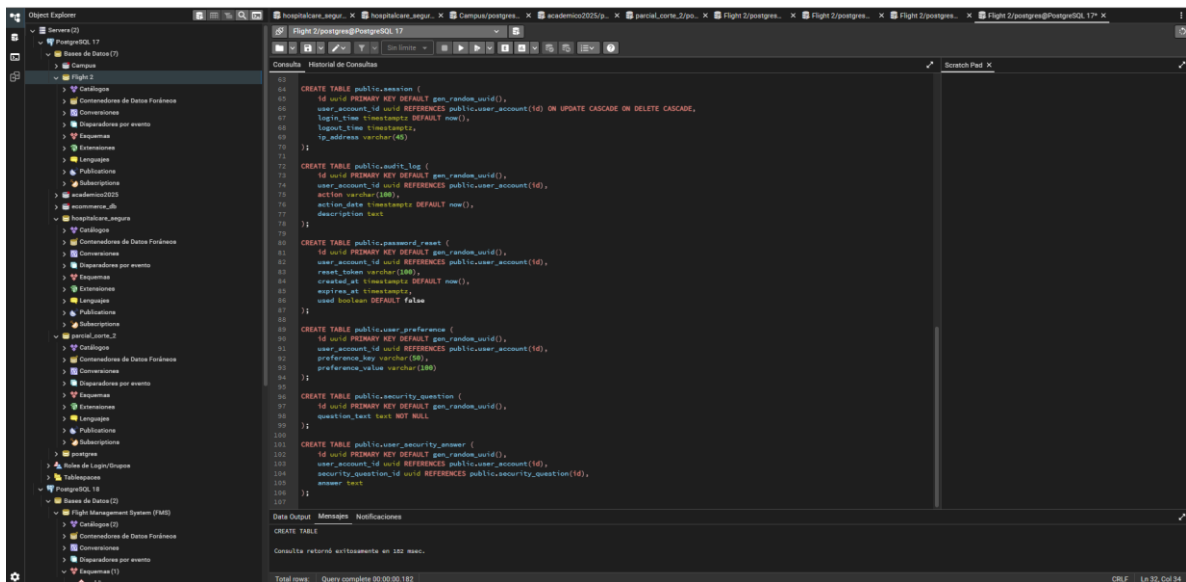
Creación de tablas.



```
1 CREATE TABLE public.role (
2     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
3     name varchar(50) NOT NULL,
4     description text
5 );
6
7 CREATE TABLE public.permission (
8     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
9     name varchar(100) NOT NULL,
10    description text
11 );
12
13 CREATE TABLE public.role_permission (
14     role_id uuid REFERENCES public.role(id) ON UPDATE CASCADE ON DELETE CASCADE,
15     permission_id uuid REFERENCES public.permission(id) ON UPDATE CASCADE ON DELETE CASCADE,
16     PRIMARY KEY (role_id, permission_id)
17 );
18
19 CREATE TABLE public.module (
20     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
21     name varchar(50) NOT NULL,
22     description text
23 );
24
25 CREATE TABLE public.form (
26     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
27     name varchar(50) NOT NULL,
28     url varchar(100),
29     description text
30 );
31
32 CREATE TABLE public.form_module (
33     form_id uuid REFERENCES public.form(id) ON UPDATE CASCADE ON DELETE CASCADE,
34     module_id uuid REFERENCES public.module(id) ON UPDATE CASCADE ON DELETE CASCADE,
35     PRIMARY KEY (form_id, module_id)
36 );
37
38 CREATE TABLE public.role_form (
39     role_id uuid REFERENCES public.role(id) ON UPDATE CASCADE ON DELETE CASCADE,
40     form_id uuid REFERENCES public.form(id) ON UPDATE CASCADE ON DELETE CASCADE,
41     PRIMARY KEY (role_id, form_id)
42 );
```

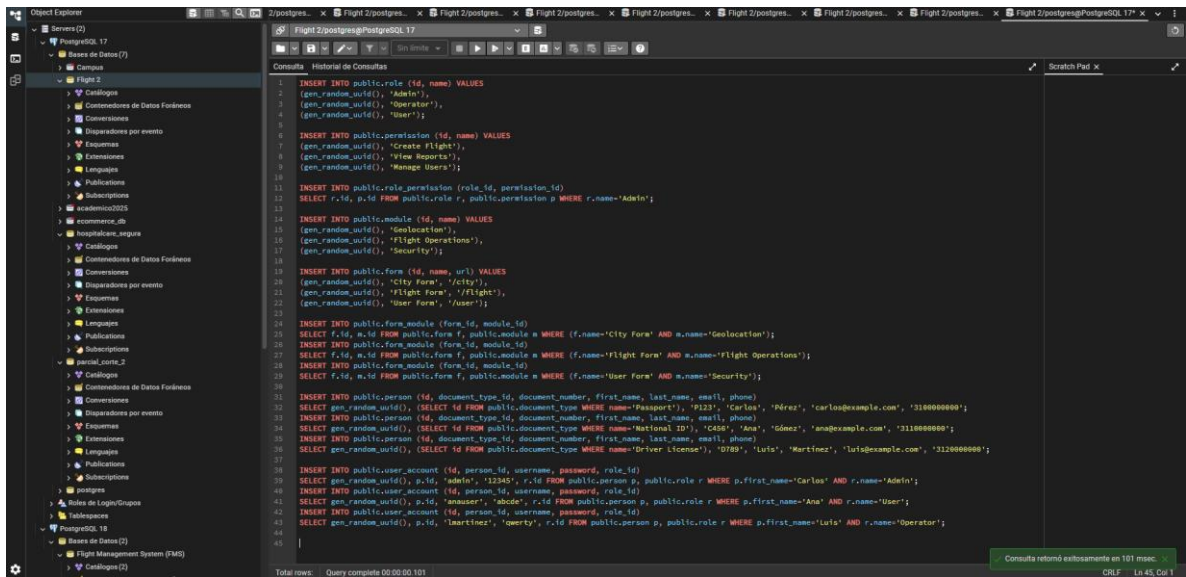


```
43 CREATE TABLE public.person (
44     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
45     document_type_id uuid REFERENCES public.document_type(id),
46     document_number varchar(80),
47     first_name varchar(50),
48     last_name varchar(50),
49     email varchar(100),
50     phone varchar(20)
51 );
52
53 CREATE TABLE public.user_account (
54     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
55     person_id uuid REFERENCES public.person(id) ON UPDATE CASCADE ON DELETE RESTRICT,
56     username varchar(50) UNIQUE NOT NULL,
57     password varchar(100) NOT NULL,
58     role_id uuid REFERENCES public.role(id),
59     is_active boolean DEFAULT true,
60     created_at timestamp DEFAULT now()
61 );
62
63 CREATE TABLE public.session (
64     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
65     user_account_id uuid REFERENCES public.user_account(id) ON UPDATE CASCADE ON DELETE CASCADE,
66     login_time timestamp DEFAULT now(),
67     logout_time timestamp,
68     ip_address varchar(40)
69 );
70
71 CREATE TABLE public.notification (
72     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
73     user_account_id uuid REFERENCES public.user_account(id),
74     action varchar(100),
75     action_date timestamp DEFAULT now(),
76     description text
77 );
78
79 CREATE TABLE public.password_reset (
80     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
81     user_account_id uuid REFERENCES public.user_account(id),
82     reset_token varchar(100),
83     created_at timestamp DEFAULT now(),
84     expires_at timestamp,
85     used boolean DEFAULT false
86 );
```



Prácticas DML

Insertar registros en cada Entidad.



Módulo Human Resources

Entidad	Atributos	Notas del estudiante
employee	id, person_id, position, hire_date	Representa un empleado.
contract	id, employee_id, start_date, end_date, salary	Contrato laboral del empleado.

training_program	id, name	Capacitaciones disponibles.
employee_training	employee_id, training_id	Relación muchos a muchos entre empleados y capacitaciones.

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **Id**, excepto employee_training ya q es compuesta.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q debe definirse con NOT NULL son person_id, position, start_date, name.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

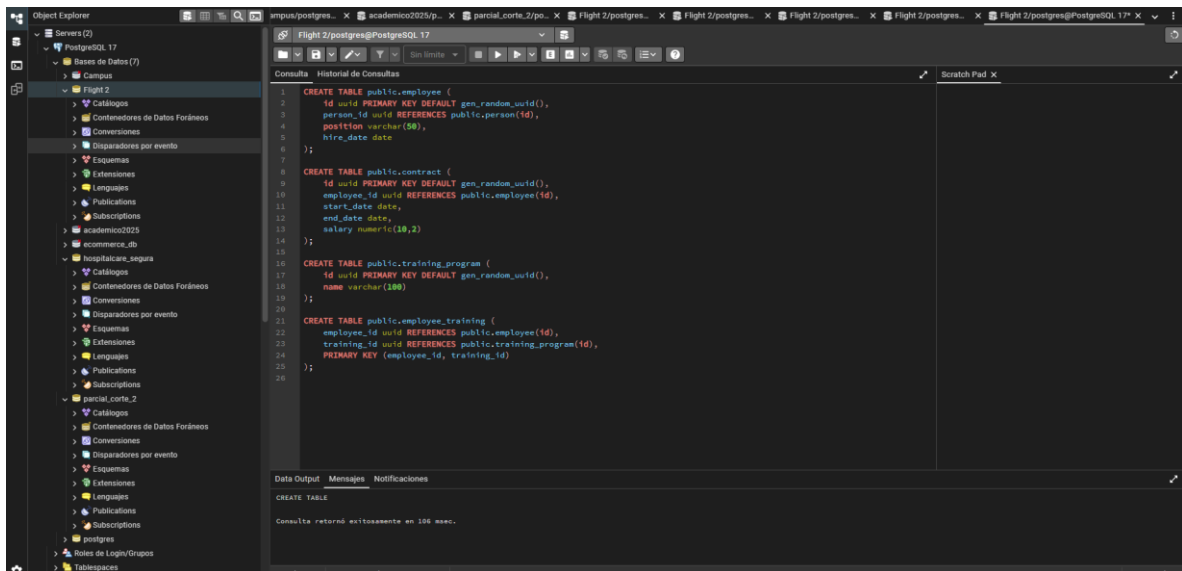
Rta: 1-N (employee → contract) y N-M (employee ↔ training_program).

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: FOREIGN KEY = fk_employee_person, fk_contract_employee, fk_employee_training_trainingprogram.

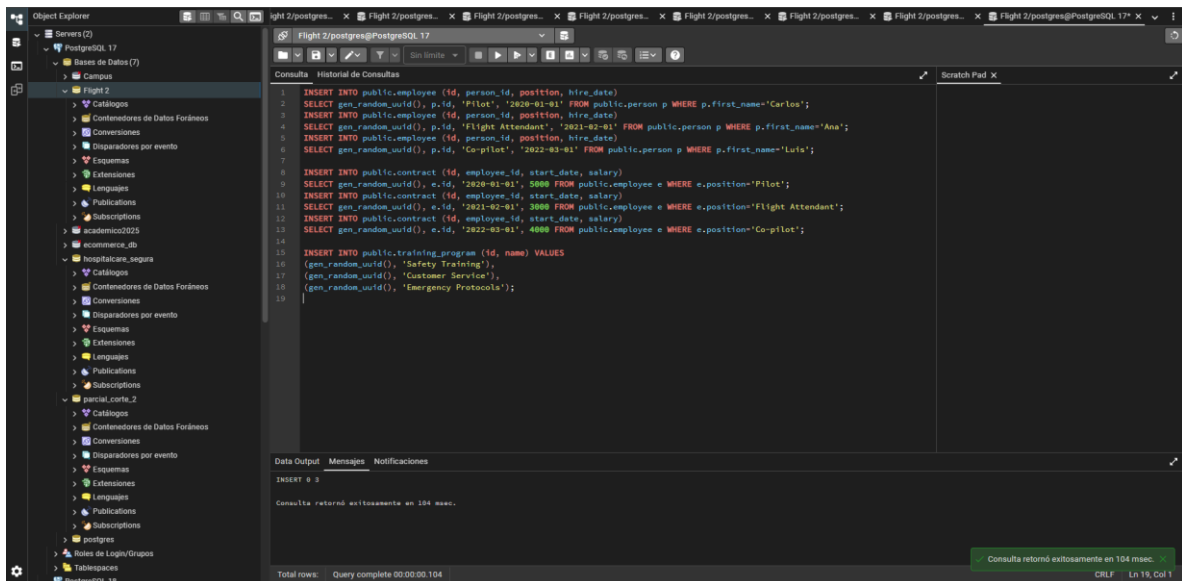
Prácticas DDL

Creación de tablas.



Prácticas DML

Insertar registros en cada Entidad.



Módulo Aircraft Management

Entidad	Atributos	Notas del estudiante
aircraft	id, registration, aircraft_type_id	Aeronaves registradas.
maintenance_type	id, name	Tipo de mantenimiento (A Check, B Check, etc.).
maintenance_record	id, aircraft_id, maintenance_type_id, maintenance_date	Registro de mantenimientos.

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **Id**.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q debe definirse con NOT NULL son registration, aircraft_type_id, maintenance_date.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

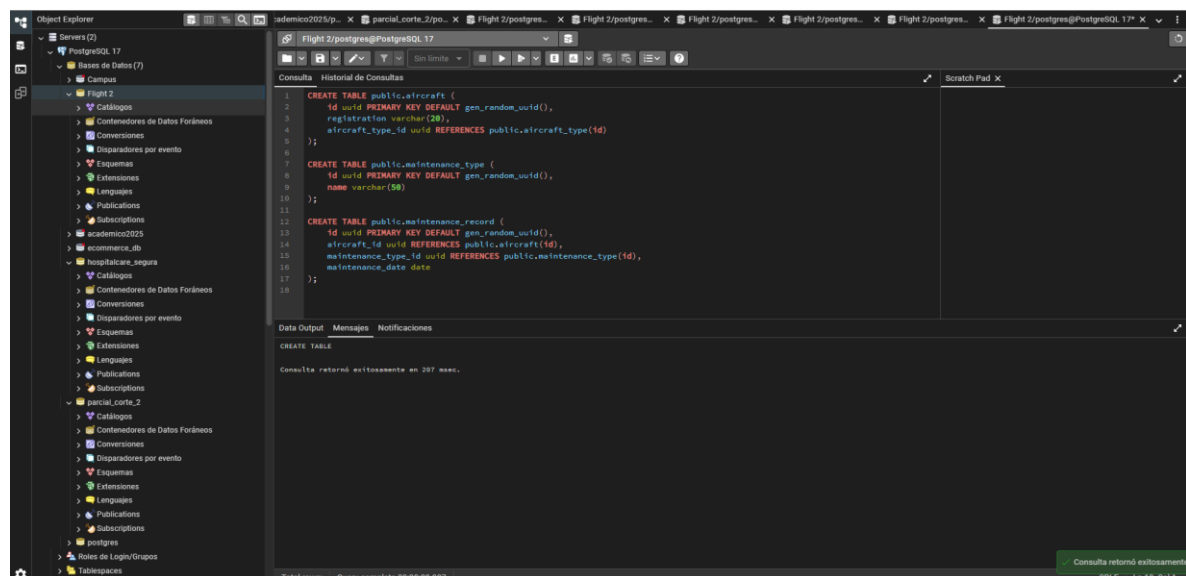
Rta: 1-N (aircraft → maintenance_record).

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: FOREIGN KEY = fk_aircraft_aircrafttype, fk_maintenancerecord_aircraft.

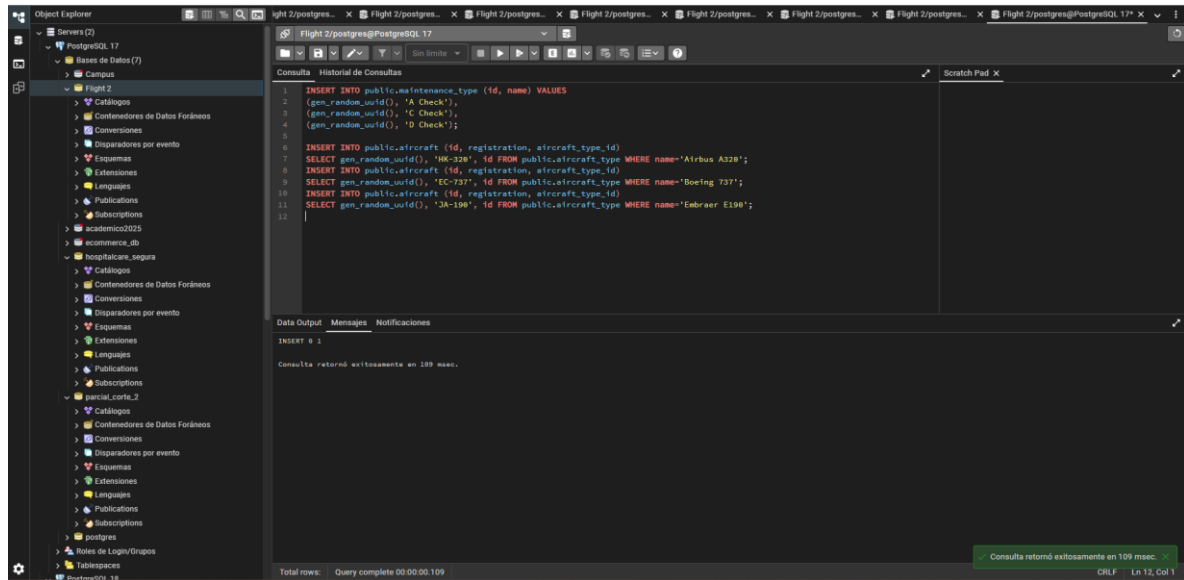
Prácticas DDL

Creación de tablas.



Prácticas DML

Insertar registros en cada Entidad.



Módulo Flight Operations

Entidad	Atributos (en inglés)	Notas del estudiante
flight	id, flight_number, flight_type_id, aircraft_id, origin_airport_id, destination_airport_id, departure_time, arrival_time	Vuelos programados.
crew_assignment	id, flight_id, employee_id, crew_role_id	Asignación de tripulación.
ticket	id, flight_id, passenger_id, ticket_class_id, seat, price	Boletos vendidos.
baggage	id, ticket_id, weight, tag_number	Equipaje registrado.

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **id**.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q debe definirse con NOT NULL son flight_number, aircraft_id, origin_airport_id, destination_airport_id.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

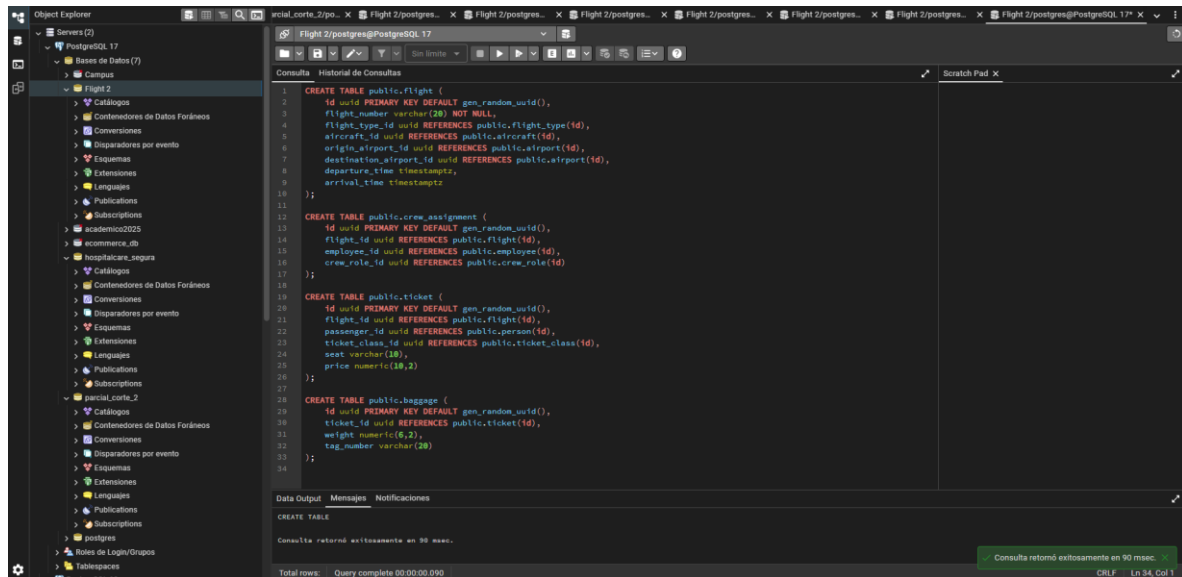
Rta: 1-N (flight → ticket, ticket → baggage) y N-M (flight ↔ employee).

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: FOREIGN KEY = fk_flight_aircraft, fk_ticket_flight, fk_baggage_ticket, fk_crewassignment_flight.

Prácticas DDL

Creación de tablas.



```

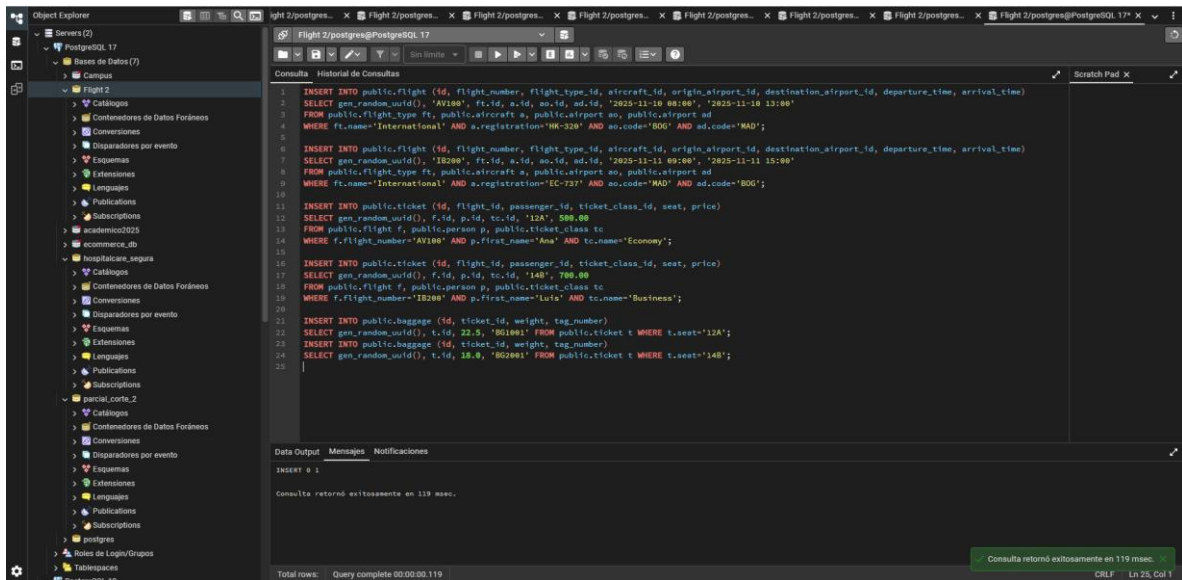
1 CREATE TABLE public.flight (
2     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
3     flight_number varchar(20) NOT NULL,
4     flight_type_id uuid REFERENCES public.flight_type(id),
5     aircraft_id uuid REFERENCES public.aircraft(id),
6     origin_airport_id uuid REFERENCES public.airport(id),
7     destination_airport_id uuid REFERENCES public.airport(id),
8     departure_time timestamp,
9     arrival_time timestamp
10 );
11
12 CREATE TABLE public.crew_assignment (
13     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
14     flight_id uuid REFERENCES public.flight(id),
15     employee_id uuid REFERENCES public.employee(id),
16     crew_role_id uuid REFERENCES public.crew_role(id)
17 );
18
19 CREATE TABLE public.ticket (
20     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
21     flight_id uuid REFERENCES public.flight(id),
22     passenger_id uuid REFERENCES public.person(id),
23     ticket_class_id uuid REFERENCES public.ticket_class(id),
24     seat varchar(10),
25     price numeric(10,2)
26 );
27
28 CREATE TABLE public.baggage (
29     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
30     ticket_id uuid REFERENCES public.ticket(id),
31     weight numeric(6,2),
32     tag_number varchar(20)
33 );
34

```

Data Output: CREATE TABLE
Consulta retornó exitosamente en 90 msec.

Prácticas DML

Insertar registros en cada Entidad.



Módulo Passengers & services

Entidad	Atributos	Notas del estudiante
passenger_service	id, ticket_id, service_name	Servicios adicionales solicitados por los pasajeros.

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **Id**.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q debe definirse con NOT NULL son ticket_id, service_name.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

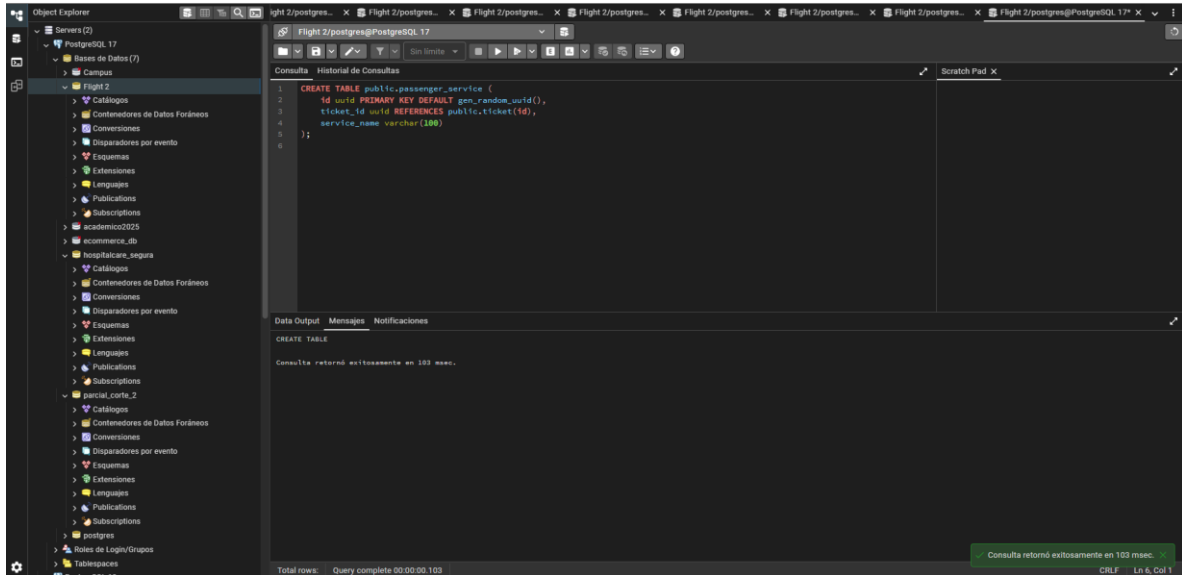
Rta: 1-N (ticket → passenger_service).

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: FOREIGN KEY = fk_passengerservice_ticket.

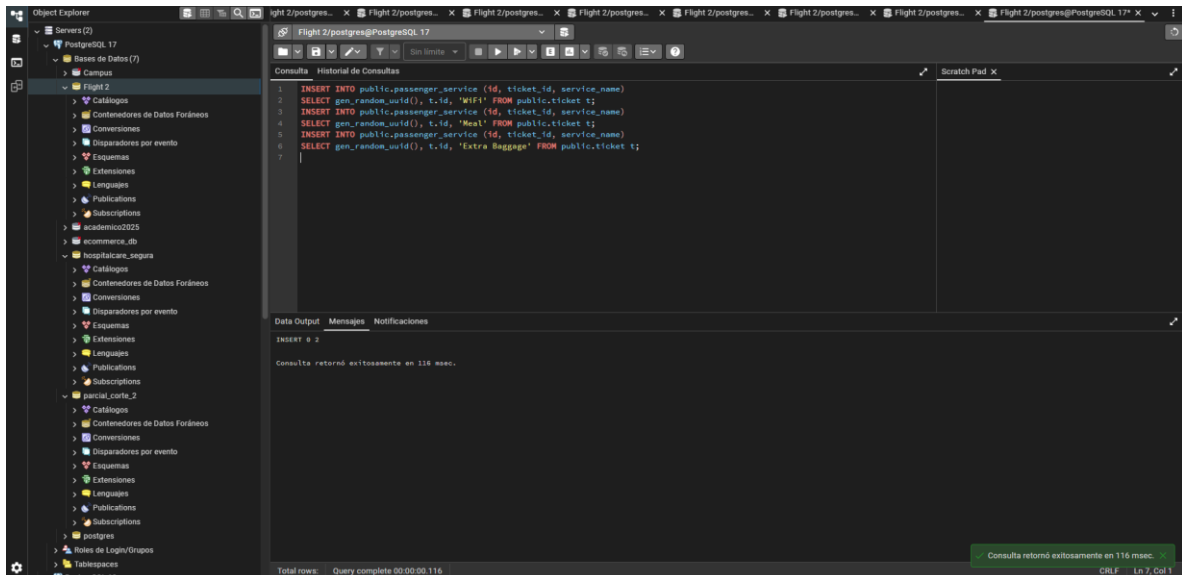
Prácticas DDL

Creación de tablas.



Prácticas DML

Insertar registros en cada Entidad.



Módulo Notifications

Entidad	Atributos (en inglés)	Notas del estudiante
message_template	id, name, content	Plantillas de mensajes automáticos (correo, SMS, etc.).
notification	id, template_id, recipient, sent_at	Notificaciones enviadas a los usuarios.

Preguntas para reflexionar:

- ¿Cuál sería la clave primaria de cada entidad?

Rta: La PRIMARY KEY para cada entidad es **Id**.

- ¿Qué atributos deberían definirse como NOT NULL?

Rta: Los atributos q debe definirse con NOT NULL son template_id, recipient, sent_at.

- ¿Qué tipo de relaciones se identifican (uno a muchos, muchos a muchos)?

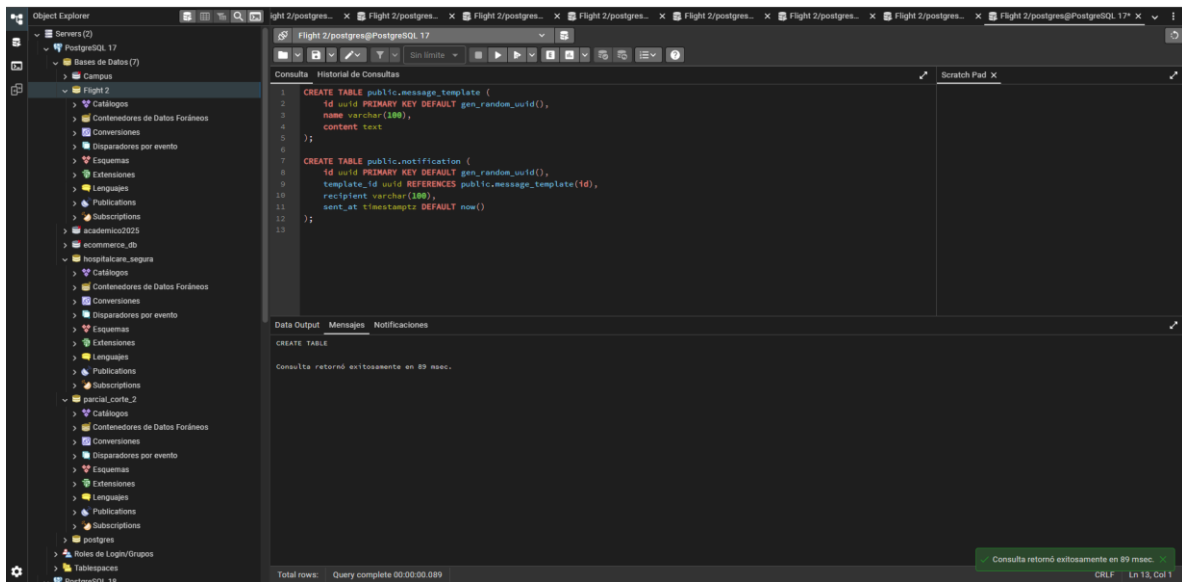
Rta: 1-N (message_template → notification).

- ¿Qué nombres asignarías a las claves foráneas según las convenciones del modelo?

Rta: FOREIGN KEY = k_notification_template.

Prácticas DDL

Creación de tablas.



Prácticas DML

Insertar registros en cada Entidad.

