

## 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
<b>continent</b>	id, name	Contiene los continentes donde opera el sistema.
<b>country</b>	id, name, continent_id	Cada país pertenece a un continente.
<b>state</b>	id, name, country_id	Cada estado pertenece a un país.
<b>city</b>	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: FOREING KEY = fk\_country\_continent, fk\_state\_country, fk\_city\_state

### Prácticas DDL

Creación de tablas.

```

CREATE TABLE public.continent (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    status varchar(10) DEFAULT 'ACTIVE',
    created_at timestamp DEFAULT now()
);
COMMENT ON TABLE public.continent IS 'List of continents';
COMMENT ON COLUMN public.continent.name IS 'Continent name';

CREATE TABLE public.country (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    continent_id uuid REFERENCES public.continent(id) ON UPDATE CASCADE ON DELETE RESTRICT,
    status varchar(10) DEFAULT 'ACTIVE',
    created_at timestamp DEFAULT now()
);
COMMENT ON TABLE public.country IS 'Countries grouped by continent';

CREATE TABLE public.state (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    country_id uuid REFERENCES public.country(id) ON UPDATE CASCADE ON DELETE RESTRICT,
    status varchar(10) DEFAULT 'ACTIVE',
    created_at timestamp DEFAULT now()
);
COMMENT ON TABLE public.state IS 'States belonging to countries';

CREATE TABLE public.city (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    state_id uuid REFERENCES public.state(id) ON UPDATE CASCADE ON DELETE RESTRICT,
    status varchar(10) DEFAULT 'ACTIVE',
    created_at timestamp DEFAULT now()
);
COMMENT ON TABLE public.city IS 'Cities registered in each state';

```

## Prácticas DML

Insertar registros en cada Entidad.

```

INSERT INTO public.continent (id, name) VALUES
('gen_random_uuid()', 'America'),
('gen_random_uuid()', 'Europe'),
('gen_random_uuid()', 'Asia');

INSERT INTO public.country (id, name, continent_id)
SELECT gen_random_uuid(), 'Colombia', id FROM public.continent WHERE name='America';
INSERT INTO public.country (id, name, continent_id)
SELECT gen_random_uuid(), 'Spain', id FROM public.continent WHERE name='Europe';
INSERT INTO public.country (id, name, continent_id)
SELECT gen_random_uuid(), 'Japan', id FROM public.continent WHERE name='Asia';

INSERT INTO public.state (id, name, country_id)
SELECT gen_random_uuid(), 'Cundinamarca', id FROM public.country WHERE name='Colombia';
INSERT INTO public.state (id, name, country_id)
SELECT gen_random_uuid(), 'Madrid', id FROM public.country WHERE name='Spain';
INSERT INTO public.state (id, name, country_id)
SELECT gen_random_uuid(), 'Tokyo', id FROM public.country WHERE name='Japan';

INSERT INTO public.city (id, name, state_id)
SELECT gen_random_uuid(), 'Bogotá', id FROM public.state WHERE name='Cundinamarca';
INSERT INTO public.city (id, name, state_id)
SELECT gen_random_uuid(), 'Madrid', id FROM public.state WHERE name='Madrid';
INSERT INTO public.city (id, name, state_id)
SELECT gen_random_uuid(), 'Tokyo', id FROM public.state WHERE name='Tokyo';

```

## Módulo Infrastructure

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

<b>boarding_gate</b>	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)?

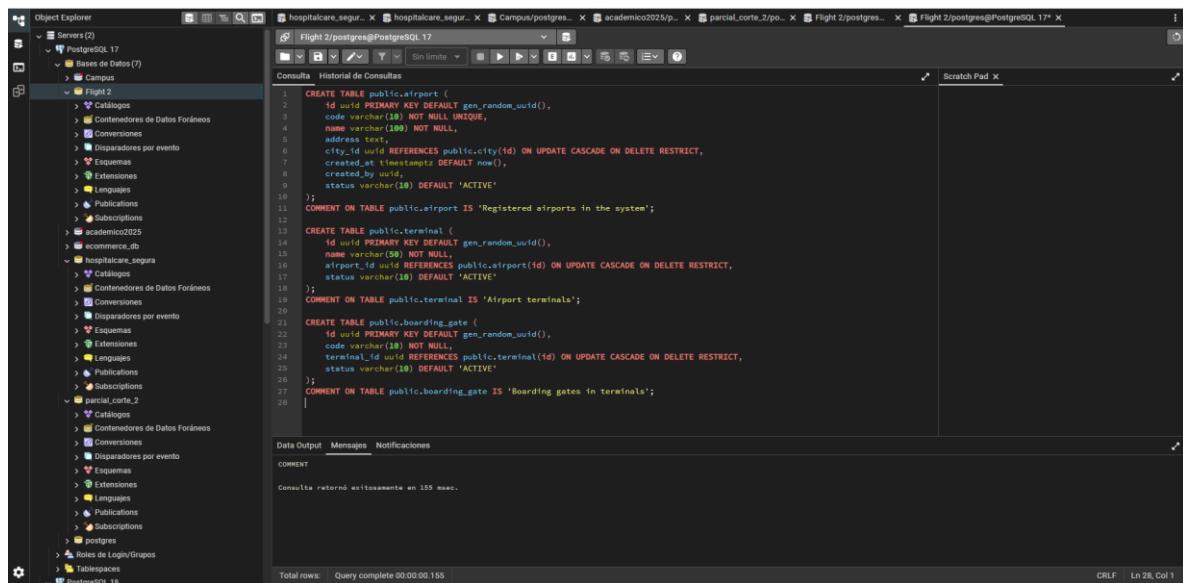
Rta: 1-N (airport → terminal → boarding\_gate).

- ¿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.



```

CREATE TABLE public.airport (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    code varchar(10) NOT NULL UNIQUE,
    name varchar(100) NOT NULL,
    address text,
    city_id uuid REFERENCES public.city(id) ON UPDATE CASCADE ON DELETE RESTRICT,
    created_at timestamptz DEFAULT now(),
    created_by uuid,
    status varchar(10) DEFAULT 'ACTIVE'
);
COMMENT ON TABLE public.airport IS 'Registered airports in the system';
CREATE TABLE public.terminal (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    airport_id uuid REFERENCES public.airport(id) ON UPDATE CASCADE ON DELETE RESTRICT,
    status varchar(10) DEFAULT 'ACTIVE'
);
COMMENT ON TABLE public.terminal IS 'Airport terminals';
CREATE TABLE public.boarding_gate (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    code varchar(10) NOT NULL,
    terminal_id uuid REFERENCES public.terminal(id) ON UPDATE CASCADE ON DELETE RESTRICT,
    status varchar(10) DEFAULT 'ACTIVE'
);
COMMENT ON TABLE public.boarding_gate IS 'Boarding gates in terminals';

```

# Prácticas DML

Insertar registros en cada Entidad.

The screenshot shows the Visual Studio Code interface with the following details:

- Object Explorer:** Shows the project structure with several databases (Servers 2, Flight 2/postgres\_17, Flight 2/postgres\_18, Flight 2/postgres\_19, Flight 2/postgres\_20, Flight 2/postgres\_21, Flight 2/postgres\_22) and their respective schemas.
- Flight 2/postgres\_17 Database:** The active database tab. It contains a table named "Consulta" with the following data:

id	code	name	address	city_id
1	'BOG'	'EL Dorado International Airport'	'Av. El Dorado, Bogotá'	1
2	'MAD'	'Adolfo Suárez Madrid-Barajas'	'Av. de la Hispanidad, Madrid'	2
3	'HND'	'Tokyo Haneda Airport'	'Ota City, Tokyo'	3
4	'BOS'	'Boston Logan International Airport'	'Pilgrim Street, Boston'	4
5	'CDG'	'Charles de Gaulle Airport'	'Roissy-en-France, Paris'	5
6	'LHR'	'London Heathrow Airport'	'Heathrow, London'	6
7	'SIN'	'Singapore Changi Airport'	'Changi, Singapore'	7
8	'PEK'	'Beijing Capital International Airport'	'Daxing, Beijing'	8
9	'ICN'	'Incheon International Airport'	'Incheon, Seoul'	9
10	'DXB'	'Dubai International Airport'	'Dubai, UAE'	10
11	'JFK'	'John F. Kennedy International Airport'	'Queens, New York'	11
12	'SFO'	'San Francisco International Airport'	'South San Francisco, California'	12
13	'ORD'	'O'Hare International Airport'	'Chicago, Illinois'	13
14	'PHL'	'Philadelphia International Airport'	'Philadelphia, Pennsylvania'	14
15	'LAX'	'Los Angeles International Airport'	'Los Angeles, California'	15
16	'MIA'	'Miami International Airport'	'Miami, Florida'	16
17	'CGK'	'Soekarno-Hatta International Airport'	'Tangerang, Indonesia'	17
18	'PEQ'	'Puerto Vallarta International Airport'	'Puerto Vallarta, Mexico'	18
19	'MEX'	'Mexico City International Airport'	'Mexico City, Mexico'	19
20	'CDMX'	'Mexico City International Airport (Nuevo)'	'Mexico City, Mexico'	20
21	'GRU'	'Guarulhos International Airport'	'São Paulo, Brazil'	21
- Data Output Tab:** Shows the results of the last query: "Consulta retornó exitosamente en 180 msec."
- Status Bar:** Total rows: 21, Query complete 00:00:00.180, CRLF, Ln 21, Col 101.

## Módulo Parameterization

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

```

CREATE TABLE public.document_type (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL
);
COMMENT ON TABLE public.document_type IS 'Types of identity documents';

CREATE TABLE public.flight_type (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL
);
COMMENT ON TABLE public.flight_type IS 'Types of flights (Domestic, International)';

CREATE TABLE public.aircraft_type (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    capacity int,
    status varchar(10) DEFAULT 'ACTIVE'
);
COMMENT ON TABLE public.aircraft_type IS 'Aircraft models and capacity';

CREATE TABLE public.crew_role (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL
);
COMMENT ON TABLE public.crew_role IS 'Crew roles (Pilot, Attendant)';

CREATE TABLE public.ticket_class (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL
);
COMMENT ON TABLE public.ticket_class IS 'Ticket class (Economy, Business)';

```

## Prácticas DML

Insertar registros en cada Entidad.

```

Object Explorer
  ▾ Servers(2)
    ▾ PostgreSQL 17
      ▾ Bases de Datos (?)
        Flight 2
          ▾ Catalogos
          ▾ Contenedores de Datos Foraneos
          ▾ Conversiones
          ▾ Disparadores por evento
          ▾ Esquemas
          ▾ Extensiones
          ▾ Lenguajes
          ▾ Publicaciones
          ▾ Subscripciones
          ▾ Academicos2023
          ▾ ecommerce_db
          ▾ Hospitalice_segura
            ▾ Catalogos
            ▾ Contenedores de Datos Foraneos
            ▾ Conversiones
            ▾ Disparadores por evento
            ▾ Esquemas
            ▾ Extensiones
            ▾ Lenguajes
            ▾ Publicaciones
            ▾ Subscripciones
            ▾ parcial_corte_2
              ▾ Catalogos
              ▾ Contenedores de Datos Foraneos
              ▾ Conversiones
              ▾ Disparadores por evento
              ▾ Esquemas
              ▾ Extensiones
              ▾ Lenguajes
              ▾ Publicaciones
              ▾ Subscripciones
              ▾ postgres
                ▾ Roles de Login/Grupos
                ▾ Tableros
            ▾ Databases(16)

Flight 2/postgres@PostgreSQL 17* x Flight 2/postgres... x Flight 2/postgres@PostgreSQL 17* x

Consulta Historial de Consultas
Scratch Pad X

Consulta: Consulta retomó exitosamente en 99 msec. ✓ CRLF | Ln 25, Col 1

1 INSERT INTO public.document_type (id, name) VALUES
2 (gen_random_uuid(), 'Passport');
3 (gen_random_uuid(), 'National ID');
4 (gen_random_uuid(), 'Driver License');
5
6 INSERT INTO public.flight_type (id, name) VALUES
7 (gen_random_uuid(), 'Domestic'),
8 (gen_random_uuid(), 'International'),
9 (gen_random_uuid(), 'Charter');
10
11 INSERT INTO public.aircraft_type (id, name, capacity) VALUES
12 (gen_random_uuid(), 'Airbus A320', 180);
13 (gen_random_uuid(), 'Boeing 737', 200);
14 (gen_random_uuid(), 'Embraer E190', 180);
15
16 INSERT INTO public.create_role (id, name) VALUES
17 (gen_random_uuid(), 'Pilot');
18 (gen_random_uuid(), 'Flight Attendant'),
19 (gen_random_uuid(), 'Copilot');
20
21 INSERT INTO public.ticket_class (id, name) VALUES
22 (gen_random_uuid(), 'Economy'),
23 (gen_random_uuid(), 'Business'),
24 (gen_random_uuid(), 'First');
25

Data Output Mensajes Notificaciones
INSERT 0 0

Consultas retornó exitosamente en 99 msec.

Total rows: Query complete 00:00:00.099

```

## Módulo Security

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

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

```
CREATE TABLE public.permission (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(100) NOT NULL,
    description text
);

CREATE TABLE public.role_permission (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    role_id uuid REFERENCES public.role(id) ON UPDATE CASCADE ON DELETE CASCADE,
    permission_id uuid REFERENCES public.permission(id) ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (role_id, permission_id)
);

CREATE TABLE public.module (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    description text
);

CREATE TABLE public.form (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50) NOT NULL,
    url varchar(100),
    description text
);

CREATE TABLE public.form_module (
    form_id uuid REFERENCES public.form(id) ON UPDATE CASCADE ON DELETE CASCADE,
    module_id uuid REFERENCES public.module(id) ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (form_id, module_id)
);

CREATE TABLE public.role_form (
    role_id uuid REFERENCES public.role(id) ON UPDATE CASCADE ON DELETE CASCADE,
    form_id uuid REFERENCES public.form(id) ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (role_id, form_id)
);
```

```
CREATE TABLE public.person (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    document_type_id uuid REFERENCES public.document_type(id),
    document_number varchar(20),
    first_name varchar(50),
    last_name varchar(50),
    email varchar(100),
    phone varchar(20)
);

CREATE TABLE public.user_account (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    person_id uuid REFERENCES public.person(id) ON UPDATE CASCADE ON DELETE RESTRICT,
    username varchar(50) UNIQUE NOT NULL,
    password varchar(100) NOT NULL,
    role_id uuid REFERENCES public.role(id),
    is_active boolean DEFAULT true,
    created_at timestampz DEFAULT now()
);

CREATE TABLE public.user_session (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id) ON UPDATE CASCADE ON DELETE CASCADE,
    login_time timestampz DEFAULT now(),
    logout_time timestampz,
    ip_address varchar(40)
);

CREATE TABLE public.user_login_log (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id),
    action varchar(100),
    action_date timestampz DEFAULT now(),
    description text
);

CREATE TABLE public.user_expression_log (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id),
    reset_token varchar(200),
    created_at timestampz DEFAULT now(),
    expired_at timestampz,
    used boolean DEFAULT false
);
```

```

CREATE TABLE public.mailsession (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id) ON UPDATE CASCADE ON DELETE CASCADE,
    login_time timestampz DEFAULT now(),
    logout_time timestampz,
    ip_address varchar(45)
);

CREATE TABLE public.audit_log (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id),
    action_type varchar(100),
    action_date timestampz DEFAULT now(),
    description text
);

CREATE TABLE public.password_reset (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id),
    reset_token varchar(100),
    expiration_at timestampz,
    used boolean DEFAULT false
);

CREATE TABLE public.preference (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id),
    preference_key varchar(100),
    preference_value varchar(100)
);

CREATE TABLE public.security_question (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    question_text text NOT NULL
);

CREATE TABLE public.user_security_answer (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    user_account_id uuid REFERENCES public.user_account(id),
    security_question_id uuid REFERENCES public.security_question(id),
    answer_text
);

```

## Prácticas DML

Insertar registros en cada Entidad.

```

1. INSERT INTO public.role (id, name) VALUES
2. (gen_random_uuid(), 'Admin'),
3. (gen_random_uuid(), 'Operator'),
4. (gen_random_uuid(), 'User');

5. INSERT INTO public.permission (id, name) VALUES
6. (gen_random_uuid(), 'Create Flight'),
7. (gen_random_uuid(), 'View Reports'),
8. (gen_random_uuid(), 'Manage Users');

9. INSERT INTO public.role_permission (role_id, permission_id)
10. SELECT r.id, p.id FROM public.role r, public.permission p WHERE r.name='Admin';

11. INSERT INTO public.module (id, name) VALUES
12. (gen_random_uuid(), 'Geolocation'),
13. (gen_random_uuid(), 'Flight Operations'),
14. (gen_random_uuid(), 'Security');

15. INSERT INTO public.form (id, name, url) VALUES
16. (gen_random_uuid(), 'City Form', '/city'),
17. (gen_random_uuid(), 'Flight Form', '/flight'),
18. (gen_random_uuid(), 'User Form', '/user');

19. INSERT INTO public.form_module (form_id, module_id)
20. SELECT f.id, m.id FROM public.form f, public.module m WHERE (f.name='City Form' AND m.name='Geolocation');
21. INSERT INTO public.form_module (form_id, module_id)
22. SELECT f.id, m.id FROM public.form f, public.module m WHERE (f.name='Flight Form' AND m.name='Flight Operations');
23. INSERT INTO public.form_module (form_id, module_id)
24. SELECT f.id, m.id FROM public.form f, public.module m WHERE (f.name='User Form' AND m.name='Security');

25. INSERT INTO public.person (id, document_type_id, document_number, first_name, last_name, email, phone)
26. SELECT gen_random_uuid(), (SELECT id FROM public.document_type WHERE name='Passport'), 'P1234', 'Carlos', 'Perez', 'carlos@example.com', '3180000000';
27. INSERT INTO public.person (id, document_type_id, document_number, first_name, last_name, email, phone)
28. SELECT gen_random_uuid(), (SELECT id FROM public.document_type WHERE name='National ID'), 'C456', 'Ana', 'Gomez', 'ana@example.com', '3110000000';
29. INSERT INTO public.person (id, document_type_id, document_number, first_name, last_name, email, phone)
30. SELECT gen_random_uuid(), (SELECT id FROM public.document_type WHERE name='Driver License'), 'D789', 'Luis', 'Martinez', 'luis@example.com', '3120000000';

31. INSERT INTO public.user_account (id, person_id, username, password, role_id)
32. SELECT gen_random_uuid(), (SELECT id FROM public.person WHERE first_name='Carlos' AND r.name='Admin');
33. INSERT INTO public.user_account (id, person_id, username, password, role_id)
34. SELECT gen_random_uuid(), (SELECT id FROM public.person WHERE first_name='Ana' AND r.name='User');
35. INSERT INTO public.user_account (id, person_id, username, password, role_id)
36. SELECT gen_random_uuid(), (SELECT id FROM public.person WHERE first_name='Luis' AND r.name='Operator');

37. INSERT INTO public.audit_log (id, user_account_id, action_type, action_date)
38. SELECT gen_random_uuid(), (SELECT id FROM public.user_account WHERE first_name='Carlos' AND r.name='Admin');
39. INSERT INTO public.audit_log (id, user_account_id, action_type, action_date)
40. SELECT gen_random_uuid(), (SELECT id FROM public.user_account WHERE first_name='Ana' AND r.name='User');
41. INSERT INTO public.audit_log (id, user_account_id, action_type, action_date)
42. SELECT gen_random_uuid(), (SELECT id FROM public.user_account WHERE first_name='Luis' AND r.name='Operator');

43. SELECT gen_random_uuid(), p.id, 'luis', '12345', r.id FROM public.person p, public.role r WHERE p.first_name='Luis' AND r.name='Operator';
44. SELECT gen_random_uuid(), p.id, 'luis', '12345', r.id FROM public.person p, public.role r WHERE p.first_name='Luis' AND r.name='Operator';

```

## Módulo Human Resources

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

<b>training_program</b>	id, name	Capacitaciones disponibles.
<b>employee_training</b>	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\_employeetraining\_trainingprogram.

## Prácticas DDL

Creación de tablas.

```

CREATE TABLE public.employee (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    person_id uuid REFERENCES public.person(id),
    position varchar(50),
    hire_date date
);

CREATE TABLE public.contract (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    employee_id uuid REFERENCES public.employee(id),
    start_date date,
    end_date date,
    salary numeric(10,2)
);

CREATE TABLE public.training_program (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(100)
);

CREATE TABLE public.employee_training (
    employee_id uuid REFERENCES public.employee(id),
    training_id uuid REFERENCES public.training_program(id),
    PRIMARY KEY (employee_id, training_id)
);

```

Data Output Mensajes Notificaciones

CREATE TABLE

Consultas retornó exitosamente en 106 msec.

## Prácticas DML

Insertar registros en cada Entidad.

```

INSERT INTO public.employee (id, person_id, position, hire_date)
SELECT gen_random_uuid(), p.id, 'Pilot', '1990-01-01' FROM public.person p WHERE p.first_name='Carlos';
INSERT INTO public.employee (id, person_id, position, hire_date)
SELECT gen_random_uuid(), p.id, 'Flight Attendant', '2021-02-01' FROM public.person p WHERE p.first_name='Ana';
INSERT INTO public.employee (id, person_id, position, hire_date)
SELECT gen_random_uuid(), p.id, 'Copilot', '2022-03-01' FROM public.person p WHERE p.first_name='Luis';

INSERT INTO public.contract (id, employee_id, start_date, salary)
SELECT gen_random_uuid(), e.id, '2020-01-01', 5000 FROM public.employee e WHERE e.position='Pilot';
INSERT INTO public.contract (id, employee_id, start_date, salary)
SELECT gen_random_uuid(), e.id, '2021-02-01', 3000 FROM public.employee e WHERE e.position='Flight Attendant';
INSERT INTO public.contract (id, employee_id, start_date, salary)
SELECT gen_random_uuid(), e.id, '2022-03-01', 4000 FROM public.employee e WHERE e.position='Co-pilot';

INSERT INTO public.training_program (id, name)
VALUES
    (gen_random_uuid(), 'Safety Training'),
    (gen_random_uuid(), 'Customer Service'),
    (gen_random_uuid(), 'Emergency Protocols');

```

Data Output Mensajes Notificaciones

INSERT 3

Consultas retornó exitosamente en 104 msec.

## Módulo Aircraft Management

Entidad	Atributos	Notas del estudiante
<b>aircraft</b>	id, registration, aircraft_type_id	Aeronaves registradas.
<b>maintenance_type</b>	id, name	Tipo de mantenimiento (A Check, B Check, etc.).
<b>maintenance_record</b>	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)?

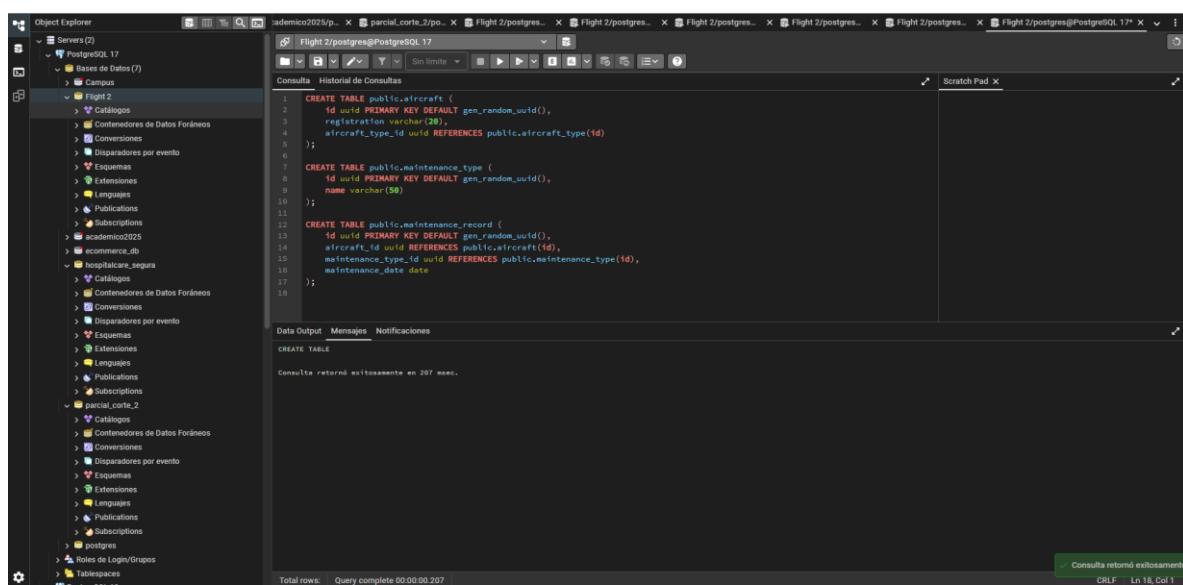
Rta:1-N (aircraft → maintenance\_record).

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

Rta: FOREING KEY = fk\_aircraft\_aircrafttype, fk\_maintenancerecord\_aircraft.

## Prácticas DDL

Creación de tablas.



```
CREATE TABLE public.aircraft (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    registration varchar(20),
    aircraft_type_id uuid REFERENCES public.aircraft_type(id)
);

CREATE TABLE public.maintenance_type (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(50)
);

CREATE TABLE public.maintenance_record (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    aircraft_id uuid REFERENCES public.aircraft(id),
    maintenance_type_id uuid REFERENCES public.maintenance_type(id),
    maintenance_date date
);
```

## Prácticas DML

Insertar registros en cada Entidad.

The screenshot shows the pgAdmin 4 interface with multiple tabs open. The left pane displays the Object Explorer with a tree view of databases, schemas, and tables. The right pane shows the Query Editor with the following SQL code:

```

INSERT INTO public.maintenance_type (id, name) VALUES
(gen_random_uuid(), 'A Check'),
(gen_random_uuid(), 'C Check'),
(gen_random_uuid(), 'D Check');

INSERT INTO public.aircraft (id, registration, aircraft_type_id)
SELECT gen_random_uuid(), 'HK-320' , id FROM public.aircraft_type WHERE name='Airbus A320';
INSERT INTO public.aircraft (id, registration, aircraft_type_id)
SELECT gen_random_uuid(), 'EC-737' , id FROM public.aircraft_type WHERE name='Boeing 737';
INSERT INTO public.aircraft (id, registration, aircraft_type_id)
SELECT gen_random_uuid(), '3A-198' , id FROM public.aircraft_type WHERE name='Embraer E190';

```

The Data Output tab shows the results of the query, indicating 0 rows affected. The status bar at the bottom right says "Consulta retornó exitosamente en 109 msec." (Query completed successfully in 109 msec).

## Módulo Flight Operations

Entidad	Atributos (en inglés)	Notas del estudiante
<b>flight</b>	id, flight_number, flight_type_id, aircraft_id, origin_airport_id, destination_airport_id, departure_time, arrival_time	Vuelos programados.
<b>crew_assignment</b>	id, flight_id, employee_id, crew_role_id	Asignación de tripulación.
<b>ticket</b>	id, flight_id, passenger_id, ticket_class_id, seat, price	Boletos vendidos.
<b>baggage</b>	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: FOREING KEY = fk\_flight\_aircraft, fk\_ticket\_flight, fk\_baggage\_ticket, fk\_crewassignment\_flight.

## Prácticas DDL

## Creación de tablas.

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** On the left, under "Servers (2)", there are two servers: "Flight 2" and "Flight 2/postgres". The "Flight 2" server is expanded, showing databases like "Flight 2", "Flight 2/catalogs", "Flight 2/constraints", "Flight 2/sequences", "Flight 2/types", and "Flight 2/roles".
- Flight 2 PostgreSQL Database:** The central window displays the "Flight 2/postgres@PostgreSQL 17" connection. It shows a script editor with the following SQL code:

```
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     start_airport_id uuid REFERENCES public.airport(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 character(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
```

**Status Bar:** At the bottom right, it says "Consulta retornó exitosamente en 99 ms".

Prácticas DML

Insertar registros en cada Entidad.

```

INSERT INTO public.flight (id, flight_number, flight_type_id, aircraft_id, origin_airport_id, destination_airport_id, departure_time, arrival_time)
SELECT gen_random_uuid(), 'AV100', ft.id, a.id, so.id, ad.id, '2025-11-10 08:00', '2025-11-10 13:00'
FROM public.flight_type ft, public.aircraft a, public.airport so, public.airport ad
WHERE ft.name='International' AND a.registration='HK-320' AND so.code='BOG' AND ad.code='MAD';

INSERT INTO public.flight (id, flight_number, flight_type_id, aircraft_id, origin_airport_id, destination_airport_id, departure_time, arrival_time)
SELECT gen_random_uuid(), 'IB200', ft.id, a.id, so.id, ad.id, '2025-11-11 09:00', '2025-11-11 15:00'
FROM public.flight_type ft, public.aircraft a, public.airport so, public.airport ad
WHERE ft.name='International' AND a.registration='EC-737' AND so.code='MAD' AND ad.code='BOG';

INSERT INTO public.ticket (id, flight_id, passenger_id, ticket_class_id, seat, price)
SELECT gen_random_uuid(), 'IB200', f.id, p.id, tc.id, '12A', 500.00
FROM public.flight f, public.person p, public.ticket_class tc
WHERE f.flight_number='IB200' AND p.first_name='Luis' AND tc.name='Economy';

INSERT INTO public.ticket (id, flight_id, passenger_id, ticket_class_id, seat, price)
SELECT gen_random_uuid(), f.id, p.id, tc.id, '14B', 700.00
FROM public.flight f, public.person p, public.ticket_class tc
WHERE f.flight_number='IB200' AND p.first_name='Ana' AND tc.name='Business';

INSERT INTO public.baggage (id, ticket_id, weight, tag_number)
SELECT gen_random_uuid(), t.id, 22.5, 'BG1001' FROM public.ticket t WHERE t.seat='12A';
INSERT INTO public.baggage (id, ticket_id, weight, tag_number)
SELECT gen_random_uuid(), t.id, 18.0, 'BG2001' FROM public.ticket t WHERE t.seat='14B';

```

## Módulo Passengers & services

Entidad	Atributos	Notas del estudiante
<b>passenger_service</b>	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)?

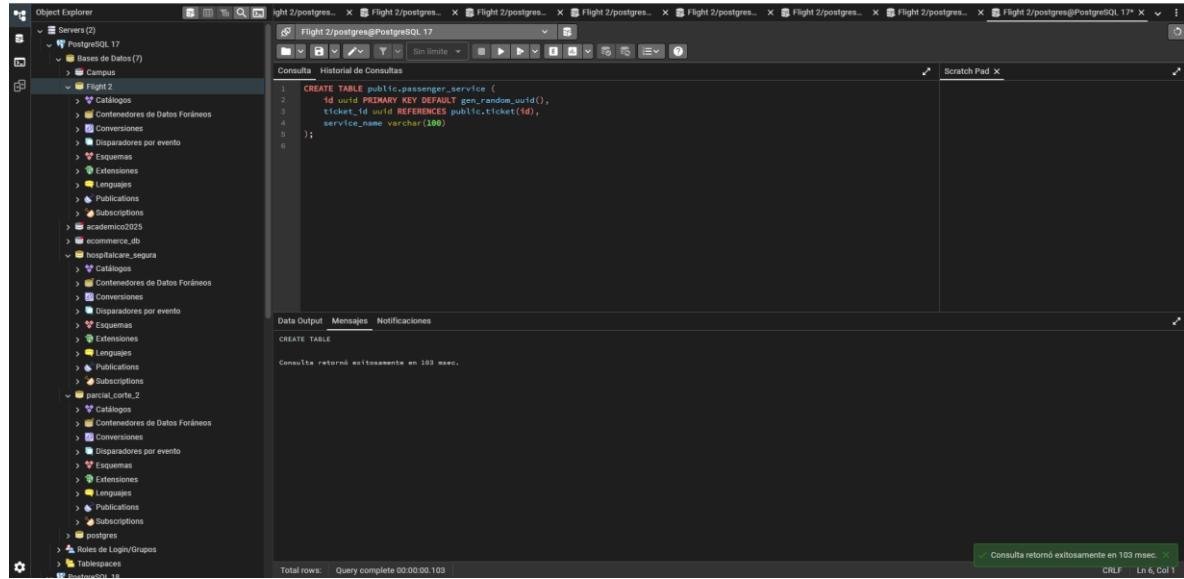
Rta:1-N (ticket → passenger\_service).

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

Rta: FOREIGN KEY = fk\_passengerservice\_ticket.

## Prácticas DDL

Creación de tablas.



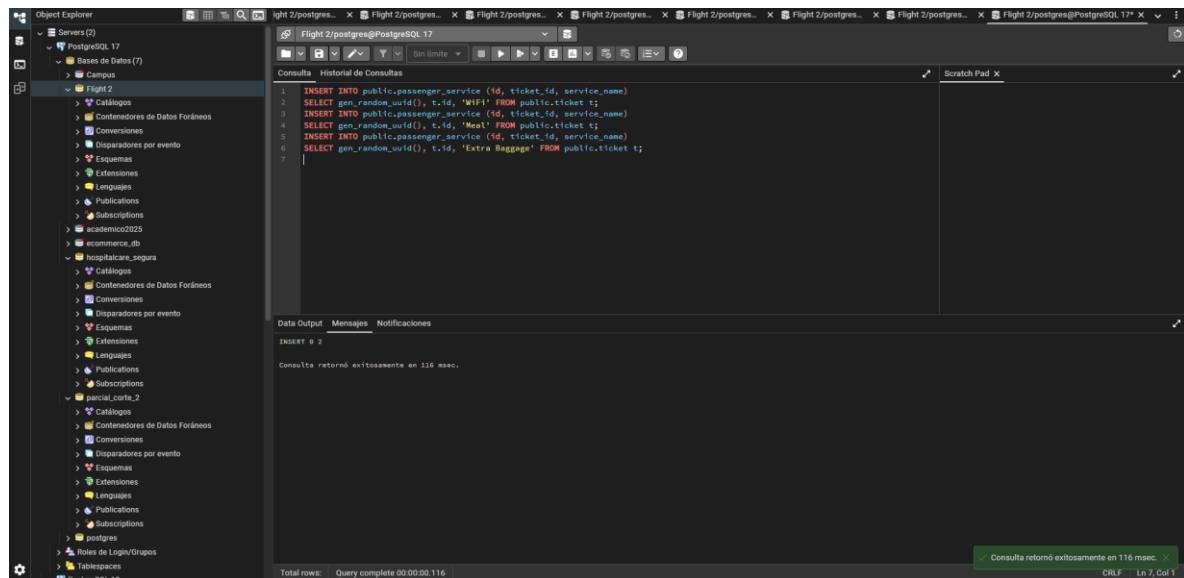
The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer tree displays a database structure under 'Flight 2'. In the center, a query editor window titled 'Flight 2/postgres@PostgreSQL 17' contains the following SQL code:

```
1 CREATE TABLE public.passenger_service (
2     id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
3     ticket_id uuid REFERENCES public.ticket(id),
4     service_name varchar(100)
5 );
```

Below the code, the 'Data Output' tab shows the result of the query: 'CREATE TABLE'. A status message at the bottom right says 'Consulta retornó exitosamente en 103 msec.'

## Prácticas DML

Insertar registros en cada Entidad.



The screenshot shows the pgAdmin 4 interface. The Object Explorer tree on the left shows the same database structure as the previous screenshot. The central query editor window has the following SQL code:

```
1 INSERT INTO public.passenger_service (id, ticket_id, service_name)
2     SELECT gen_random_uuid(), t.id, 'WiFi' FROM public.ticket t;
3 INSERT INTO public.passenger_service (id, ticket_id, service_name)
4     SELECT gen_random_uuid(), t.id, 'Meal' FROM public.ticket t;
5 INSERT INTO public.passenger_service (id, ticket_id, service_name)
6     SELECT gen_random_uuid(), t.id, 'Extra Baggage' FROM public.ticket t;
```

The 'Data Output' tab shows the result: 'INSERT 0 2'. A status message at the bottom right says 'Consulta retornó exitosamente en 116 msec.'

## Módulo Notifications

Entidad	Atributos (en inglés)	Notas del estudiante
<b>message_template</b>	id, name, content	Plantillas de mensajes automáticos (correo, SMS, etc.).
<b>notification</b>	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: FOREING KEY = k\_notification\_template.

## Prácticas DDL

Creación de tablas.

```

CREATE TABLE public.message_template (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    name varchar(100),
    content text
);

CREATE TABLE public.notification (
    id uuid PRIMARY KEY DEFAULT gen_random_uuid(),
    template_id uuid REFERENCES public.message_template(id),
    recipient varchar(100),
    sent_at timestampz DEFAULT now()
);

```

Total rows: 0 Query complete 00:00:00.089

Consulta retornó exitosamente en 89 msec. CRLF | Ln 13, Col 1

## Prácticas DML

Insertar registros en cada Entidad.

```

INSERT INTO public.message_template (id, name, content) VALUES
('gen_random_uuid()', 'Flight Delay', 'Your flight is delayed'),
('gen_random_uuid()', 'Check-in Open', 'Check-in is now available'),
('gen_random_uuid()', 'Boarding Now', 'Boarding has started');

INSERT INTO public.notification (id, template_id, recipient)
SELECT gen_random_uuid(), mt.id, 'carlos@example.com' FROM public.message_template mt WHERE mt.name='Flight Delay';
INSERT INTO public.notification (id, template_id, recipient)
SELECT gen_random_uuid(), mt.id, 'luis@example.com' FROM public.message_template mt WHERE mt.name='Check-in Open';
INSERT INTO public.notification (id, template_id, recipient)
SELECT gen_random_uuid(), mt.id, 'lula@example.com' FROM public.message_template mt WHERE mt.name='Boarding Now';

```

Consulta retornó exitosamente en 260 msec.

Consulta retornó exitosamente en 260 msec.