SOAT FARE

INSERT INTO SOAT_FARE (id, price) VALUES

(100, 118200),
(110, 243700),
(120, 326600),
(130, 758600),
(140, 368100),
(150, 368100),

(211, 789900),
(221, 943100),
(231, 1106200),

(212, 949500),
(222, 1117100),
(232, 1269300),

(310,885000),
(320,1277900),
(330,1615800),

(410,995800),
(420,1255400),
(430,1505000),

(511, 445600),
(521, 542700),
(531, 633800),

(512, 590700),
(522, 675000),
(532, 751600),

(611, 794400),
(621, 1063300),

(612, 1013900),
(622, 1276700),

(711, 268200),
(721, 333000),
(731, 429300),

(712, 334800),
(722, 411200),
(732, 503500),

(810,640300),
(910,633000),
(920,918000)

TECNO FARE

INSERT INTO TECNO_FARE (type,start,end,price) VALUES

(Motos,2023,2025,215541),
(Motos,2018,2022,215841),
(Motos,2009,2017,216141),
(Motos,2008,2008,215841),

(Liviano Particular,2023,2025,330341),
(Liviano Particular,2018,2022,330741),
(Liviano Particular,2009,2017,331041),
(Liviano Particular,2008,2008,330741),

(Liviano Publico,2023,2025,329841),
(Liviano Publico,2018,2022,330141),
(Liviano Publico,2009,2017,330441),
(Liviano Publico,2008,2008,330141),

(Pesado Particular,2023,2025,470741),
(Pesado Particular,2018,2022,471041),
(Pesado Particular,2009,2017,741241),
(Pesado Particular,2008,2008,471041),

(Pesado Público,2023,2025,470341),
(Pesado Público,2018,2022,470541),
(Pesado Público,2009,2017,470741),
(Pesado Público,2008,2008,470541)

LLENAR TABLA DE SERVICIOS

```
WITH usrx AS (
  SELECT id AS usr_id FROM usr WHERE email = 'ana@exa.co'
),
svc_map(amount, days_ago, name, service_type, plate, assurance, duration, state) AS (
  VALUES
   (120, 30, 'SOAT',                'SOAT',    'ABC123', 'SURA',   '1 año',  'APPROVED'),
   (95 , 27, 'Tecnomecánica',       'TECNO',   'ABC123', 'MAPFRE', '1 año',  'APPROVED'),
   (250, 24, 'Curso conducción C2',  'COURSE',  'ABC123', 'N/A',    '3 meses','APPROVED'),
   (180, 21, 'Comparendo SIMIT',     'TICKET',  'ABC123', 'N/A',    'N/A',    'APPROVED'),
   (99 ,  6, 'Curso teórico A2',     'COURSE',  'ABC123', 'N/A',    '2 meses','APPROVED'),
```

```
   (110,  3, 'Revisión comparendos', 'TICKET',  'ABC123', 'N/A',    'N/A',    'APPROVED')
),
targets AS (
  SELECT
    m.*,
    (CURRENT_DATE - (m.days_ago || ' days')::interval)::date AS start_date,
    ((CURRENT_DATE - (m.days_ago || ' days')::interval) + INTERVAL '1 year')::date AS
exp_date
  FROM svc_map m
)
INSERT INTO services (
  payment_id,
  usr_id,
  start_date,
  exp_date,
  name,
  service_type,
  price,
  plate,
  assurance,
  duration
)
SELECT
  NULL::integer,
  u.usr_id,
  t.start_date,
  t.exp_date,
  t.name,
  t.service_type,
  t.amount AS price,
  t.plate,
  t.assurance,
  t.duration
FROM targets t
CROSS JOIN usrx u
WHERE NOT EXISTS (
  SELECT 1
  FROM services
  WHERE services.usr_id     = u.usr_id
    AND services.name       = t.name
    AND services.start_date = t.start_date
);
```

LLENAR TABLA PAYMENTS

```
WITH base AS (
```

```
  SELECT DISTINCT
    services.price      AS amount,
    services.start_date AS release_date,
    'APPROVED'          AS state
  FROM services
  JOIN usr ON usr.id = services.usr_id
  WHERE usr.email = 'ana@exa.co'
),
missing AS (
  SELECT b.*
  FROM base b
  LEFT JOIN payments p
    ON p.amount = b.amount
   AND p.release_date = b.release_date
   AND p.state = b.state
  WHERE p.id IS NULL
)
INSERT INTO payments (amount, release_date, state)
SELECT amount, release_date, state
FROM missing;

WITH pay_map AS (
  SELECT id AS payment_id, amount, release_date
  FROM payments
  WHERE state = 'APPROVED'
)
UPDATE services
SET payment_id = pm.payment_id
FROM pay_map pm, usr
WHERE usr.id = services.usr_id
  AND usr.email = 'ana@exa.co'
  AND services.payment_id IS NULL
  AND services.price = pm.amount
  AND services.start_date = pm.release_date;
```

COMPROBAR SI FUNCIONA

```
SELECT services.id, services.usr_id, services.name, services.service_type,
       services.price, services.start_date, services.exp_date, services.payment_id
FROM services
JOIN usr ON usr.id = services.usr_id
WHERE usr.email = 'ana@exa.co'
ORDER BY services.start_date;
```

```
-- ========================
-- DATASET PARA INTEGRATION TESTS (PostgreSQL)
-- Tablas: usr, vehicle
-- Variante A: nombres de columna en snake_case
-- ========================

BEGIN;

-- (Opcional) Crear tablas si no existen: ajusta tipos y longitudes según tu DDL real.
-- Si Hibernate ya crea el esquema, puedes omitir esta sección CREATE TABLE.
CREATE TABLE IF NOT EXISTS usr (
  id              INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  name            VARCHAR(120),
  identification  VARCHAR(60) UNIQUE NOT NULL,
  email           VARCHAR(180) UNIQUE NOT NULL,
  password        VARCHAR(120) NOT NULL,
  role            VARCHAR(60),
  lon             BIGINT,
  lat             BIGINT
);

CREATE TABLE IF NOT EXISTS vehicle (
  id                  INTEGER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
  type                VARCHAR(40),
  plate               VARCHAR(15) UNIQUE NOT NULL,
  soat_rate_type      VARCHAR(40),
  techno_classification  VARCHAR(60),
  soat_expiration     DATE,
  techno_expiration   DATE,
  usr_id              INTEGER REFERENCES usr(id) ON DELETE CASCADE
);
```

```
-- Limpia e inicializa (para que el script sea idempotente al probar varias veces)
TRUNCATE TABLE vehicle RESTART IDENTITY CASCADE;
TRUNCATE TABLE usr RESTART IDENTITY CASCADE;

-- Password fijo (bcrypt) para todos, como indicaste:
-- $2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq
-- Coordenadas (Bogotá) en microgrados como enteros: lat ~ 4.6°, lon ~ -74.1° => 45940736
/ -74108108

INSERT INTO usr (name, identification, email, password, role, lon, lat) VALUES
  ('Ana Romero',      'CC1001', 'ana.romero@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq', 'USER',
-74108108, 45940736),
  ('Bruno Pérez',     'CC1002', 'bruno.perez@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq', 'USER',
-74120000, 45950000),
  ('Carla Díaz',      'CC1003', 'carla.diaz@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq', 'USER',
-74090000, 45930000),
  ('Diego Fernández', 'CC1004', 'diego.fernandez@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq', 'ADMIN',
-74105000, 45935000),
  ('Elena García',    'CC1005', 'elena.garcia@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq', 'USER',
-74099000, 45942000),
  ('Fabio Núñez',     'CC1006', 'fabio.nunez@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq',
'PARTNER',-74115000, 45928000),
  ('Gina López',      'CC1007', 'gina.lopez@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq', 'USER',
-74101000, 45941000),
  ('Hugo Salazar',    'CC1008', 'hugo.salazar@example.com',
'$2a$10$8TcBzrwjMGYiNAMipQ4YO.rfc1AMhpOBhd7lVJRYnKLlLvFb/.5Eq', 'USER',
-74095000, 45946000);

-- Asignación de vehículos (algunos con fechas vencidas antes de 2025-10-01 y otros
vigentes)
-- Nota: usa ids 1..8 según los inserts anteriores
INSERT INTO vehicle (type, plate, soat_rate_type, techno_classification, soat_expiration,
techno_expiration, usr_id) VALUES
  -- Ana (id 1) -> uno vencido, otro vigente
  ('Car',    'ABC123', 'Particular', 'M1', '2025-06-15', '2025-07-10', 1), -- vencidos
  ('Motorcycle','XYZ09A', 'Moto',    'L3e','2026-03-20', '2026-04-05', 1), -- vigentes

  -- Bruno (id 2) -> ambos vencidos
  ('Car',    'DEF456', 'Publico',    'M1', '2024-12-20', '2025-01-15', 2),
  ('Truck',  'TRK789', 'Carga',      'N2', '2025-02-28', '2025-03-10', 2),
```

```
  -- Carla (id 3) -> uno vigente, uno vencido
  ('Car',    'GHI321', 'Particular', 'M1', '2026-01-01', '2025-11-01', 3), -- SOAT vigente, Tecno casi vigente
  ('Motorcycle','MOT777','Moto',     'L3e','2025-05-05', '2025-06-01', 3), -- vencidos

  -- Diego (id 4)
  ('SUV',    'JKL654', 'Particular', 'M1', '2026-08-30', '2026-09-15', 4),

  -- Elena (id 5)
  ('Car',    'MNO987', 'Particular', 'M1', '2025-08-01', '2025-08-20', 5), -- vencidos recientes

  -- Fabio (id 6) -> varios para probar búsquedas por placa y tipos
  ('Truck',  'CGR111', 'Carga',      'N3', '2026-02-14', '2026-02-28', 6),
  ('Van',    'VAN222', 'Publico',    'N1', '2025-09-15', '2025-09-20', 6), -- vencidos

  -- Gina (id 7)
  ('Car',    'PRV333', 'Particular', 'M1', '2026-05-10', '2026-05-25', 7),

  -- Hugo (id 8)
  ('Motorcycle','BIK444','Moto',     'L3e','2025-04-14', '2025-04-30', 8); -- vencidos

COMMIT;

-- Comprobaciones rápidas sugeridas:
-- SELECT u.id, u.name, v.plate, v.soat_expiration, v.techno_expiration
-- FROM usr u JOIN vehicle v ON v.usr_id = u.id
-- ORDER BY u.id, v.id;
```