DROP TABLE IF EXISTS localisation;  
DROP TABLE IF EXISTS station;  
DROP TABLE IF EXISTS voiture;  
DROP TABLE IF EXISTS entretien;  
DROP TABLE IF EXISTS utilisateur;  
DROP TABLE IF EXISTS location;  
DROP TABLE IF EXISTS typeAbonnement;  
DROP TABLE IF EXISTS abonnement;  
  
CREATE TABLE localisation (  
 idLocalisation INTEGER PRIMARY KEY,  
 latitude FLOAT,  
 longitude FLOAT,  
 pays VARCHAR,  
 ville VARCHAR  
);  
  
CREATE TABLE station (  
 idStation INTEGER PRIMARY KEY,  
 capacite INTEGER,  
 idLocalisation INTEGER REFERENCES localisation(idLocalisation),  
 dateLocalisation DATE  
);  
  
CREATE TABLE voiture (  
 plaque VARCHAR PRIMARY KEY,  
 niveauBatterie INTEGER,  
 idLocalisation INTEGER REFERENCES localisation(idLocalisation),  
 dateLocalisation DATE,  
 idStation INTEGER REFERENCES station(idStation)  
);  
  
CREATE TABLE entretien (  
 idEntretien INTEGER PRIMARY KEY,  
 description VARCHAR,  
 plaque VARCHAR REFERENCES voiture(plaque),  
 dateEntretien DATE  
);  
  
CREATE TABLE utilisateur (  
 idUtilisateur INTEGER PRIMARY KEY,  
 nom VARCHAR,  
 prenom VARCHAR,  
 age INTEGER,  
 adresse VARCHAR,  
 telephone VARCHAR,  
 idLocalisation INTEGER REFERENCES localisation(idLocalisation),  
 dateLocalisation DATE  
);  
  
CREATE TABLE location (  
 idLocation INTEGER PRIMARY KEY,  
 dateLocation DATE,  
 idUtilisateur INTEGER REFERENCES utilisateur(idUtilisateur),  
 plaque VARCHAR REFERENCES voiture(plaque),

dureeUtilisation TIME,

prixLocation FLOAT  
);  
  
CREATE TABLE typeAbonnement (  
 idTypeAbonnement INTEGER PRIMARY KEY,  
 intitulé VARCHAR,  
 prixTypeAbonnement FLOAT,

indiceRemise INTEGER  
);  
  
CREATE TABLE abonnement (  
 idAbonnement INTEGER PRIMARY KEY,  
 dateDebutAbonnement DATE,

dateFinAbonnement DATE,  
 idUtilisateur INTEGER REFERENCES utilisateur(idUtilisateur),  
 idTypeAbonnement INTEGER REFERENCES typeAbonnement(idTypeAbonnement)  
);  
  
  
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
  
  
  
INSERT INTO localisation (idLocalisation,latitude,longitude,pays,ville) VALUES  
;

INSERT INTO station (idStation,capacite,idLocalisation,dateLocalisation DATE) VALUES

(1,10,21,’2019-01-02’),

(2,5,22,’2019-01-02’),

(3,10,23,’2019-01-02’);

INSERT INTO voiture (plaque,niveauBaterie,idLocalisation,dateLocalisation DATE,idStation) VALUES

(‘TL845KD’, 15, 6, ‘2019-01-02’, NULL),

(‘IO855PZ’, 42, 7, ‘2019-01-02’, NULL),

(‘HO154SF’, 59, 8, ‘2019-01-02’, NULL),

(‘PZ532OJ’, 36, 9, ‘2019-01-02’, NULL),

(‘OA734PS’, 84, 10, ‘2019-01-02’,1),

(‘ZL578ZD’, 84,11, ‘2019-01-02’,1),

(‘AP545OA’, 95, 12, ‘2019-01-01’,1),

(‘KD564JE’, 75, 13, ‘2019-01-02’,1),

(‘LA454JD’, 95, 14, ‘2019-01-02’,2),

(‘LK541JZ’, 81, 15, ‘2019-01-01’,2),

(‘KA743OD’, 62, 16, ‘2019-01-02’,3),

(‘AK542OA’, 74, 17, ‘2019-01-02’,3),

(‘AE756DO’, 95, 18, ‘2019-01-02’,3),

(‘AO782OD’, 67, 19, ‘2019-01-02’,3),

(‘GE524ZP’, 76, 20, ‘2019-01-02’,3);

INSERT INTO entretien (idEntretien,description,plaque,dateEntretien DATE) VALUES  
  
INSERT INTO utilisateur (idUtilisateur,nom,prenom,age,adresse,telephone,idLocalisation,dateLocalisation DATE) VALUES (1,’Jean’, ‘Claude’, 46, ‘5 rue du Boulier, Paris’, ‘0156543546’, 1, ‘2019-01-02’),

(2,’Arnold’,’Schwarzenegger’, 28, ‘45 rue de la Paix, Paris’, ‘0112654656’, 2, ‘2019-01-01’),

(3,’Jacques’, ‘Paul’, 23, ‘205 avenue des Champs-Elysées, Paris’, ‘0265413248’, 3, ‘2018-12-30’),

(4,’John’,’McClane’, 77, ‘404 impasse des Aubergines, Paris’, ‘0154135452’, 4, ‘2019-01-01’),

(5, ‘Boris’,’Viande’, 38, ‘25 rue Albert Einstein, Paris’, ‘0131265255’, 5, ‘2018-12-29’);   
  
INSERT INTO location (idLocation,dateLocation DATE,idUtilisateur,plaque, dureeUtilisation, prixLocation) VALUES

INSERT INTO typeAbonnement (idTypeAbonnement,intitulé,prixTypeAbonnement,indiceRemise) VALUES

(1,’Gratuit’, 0.00, 1),

(2,’Premium’, 9.99, 2),

(3,’Business’, 19.99,3);  
  
INSERT INTO abonnement (idAbonnement,dateDebutAbonnement DATE,dateFinAbonnement DATE,idUtilisateur,idTypeAbonnement) VALUES

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

PREPARE nouvel\_abonne(INTEGER,VARCHAR,VARCHAR,INTEGER,VARCHAR,VARCHAR,INTEGER,DATE) AS

INSERT INTO utilisateur VALUES ($1,$2,$3,$4,$5,$6,$7,$8);

EXECUTE nouvel\_abonne(16, ‘Clooney’, ‘Georges’, 57, ‘Los Angeles’, ‘0135516652’, 5, ‘2016-09-21’);

PREPARE changement\_adresse(INTEGER,VARCHAR) AS

UPDATE utilisateur

SET adresse = $2

WHERE idUtilisateur = $1;

EXECUTE changement\_adresse(16, ‘New York’);

PREPARE changement\_telephone(INTEGER,VARCHAR) AS

UPDATE utilisateur

SET telephone = $2

WHERE idUtilisateur = $1;

EXECUTE changement\_telephone(16, ‘555012834’);

PREPARE changement\_abonnement(INTEGER,INTEGER) AS

UPDATE abonnement

SET idTypeAbonnement = $2

WHERE idUtilisateur = $1;

EXECUTE changement\_abonnement(16,2);

PREPARE nouvelle\_voiture(VARCHAR,INTEGER,INTEGER,DATE,INTEGER) AS

INSERT INTO voiture VALUES ($1,$2,$3,$4,$5);

EXECUTE nouvel\_voiture(‘BR766ET’,100,7, 57,’2019-01-04’,NULL);

PREPARE suppression\_voiture(VARCHAR) AS

DELETE FROM voiture WHERE plaque = $1;

EXECUTE suppression\_voiture(‘BR766ET’);

PREPARE mise\_a\_jour\_batterie(VARCHAR, INTEGER) AS

UPDATE voiture

SET niveauBatterie = $2

WHERE plaque = $1;

EXECUTE suppression\_voiture(‘BR766ET’);

PREPARE nouvel\_entretien\_voiture(INTEGER,VARCHAR,VARCHAR,DATE) AS

INSERT INTO entertien VALUES ($1,$2,$3,$4);

EXECUTE nouvel\_entretien\_voiture(19,’Remplacement plaquettes de frein’,‘BR766ET’,’2019-01-29’);

PREPARE changement\_station(VARCHAR, INTEGER) AS

UPDATE voiture

SET idStation = $2

WHERE plaque = $1;

EXECUTE changement\_station(‘BR766ET’,2);

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// 1) Liste des 10 objets disponibles les plus proches d\_un utilisateur.

PREPARE voitures\_proches(INTEGER) AS

SELECT v.plaque, v.niveauBatterie, SQRT(SQUARE(lu.latitude - lv.latitude) + SQUARE(lu.longitude - lv.longitude) AS distance

FROM voiture AS v NATURAL JOIN localisation AS lv, utilisateur AS u NATURAL JOIN localisation AS luaa

WHERE v.idStation != NULL

AND SELECT SQRT(SQUARE(lu.latitude - lv.latitude) + SQUARE(lu.longitude - lv.longitude))

ORDER BY distance

LIMIT 10;

// 2) Objets nécessitant une intervention.

// Véhicules n’ayant pas eu d\_entretien depuis un an.

SELECT v1.plaque

FROM voiture AS v

WHERE DATEDIFF(CAST(NOW() AS DATE),

(SELECT MIN(DATEDIFF(CAST(NOW() AS DATE), e.dateEntretien))

FROM entretien AS e

WHERE e.plaque = v.plaque)

) > 365;

// 3) Calcul du prix d’une location selon l’abonnement.

PREPARE calcul\_prix\_location(INTEGER) AS

SELECT (l.dureeUtilisation \* 0.01) / t.indiceRemise

FROM location AS l NATURAL JOIN utilisateur AS u NATURAL JOIN abonnement AS a NATURAL JOIN typeAbonnement AS t

WHERE l.idLocation = $1;

EXECUTE calcul\_prix\_location(2);

// 4) Calcul du prix total des locations d’un client au cours du mois dernier.

PREPARE calcul\_prix\_total(INTEGER)

SELECT u.idUtilisateur, SUM(prixLocation)

FROM location AS l NATURAL JOIN utilisateur AS u NATURAL JOIN abonnement AS a NATURAL JOIN typeAbonnement AS t

WHERE u.idUtilisateur = $1 AND MONTH(l.dateLocation) = MONTH(NOW());

EXECUTE calcul\_prix\_total(3);

// 5) Durée moyenne d’une location par utilisateur, puis au total.

SELECT u.idUtilisateur, u.nom, AVG(l.dureeUtilisation),SUM(l.dureeUtilisation)

FROM location AS l NATURAL JOIN utilisateur AS u

GROUP BY idUtilisateur;

// 6) Chiffre d’affaire généré au cours du mois dernier.

(SELECT SUM(l.prixLocation)

FROM location AS l

WHERE MONTH(l.dateLocation) = MONTH(NOW()))

+

(SELECT SUM(t.prixTypeAbonnement)

FROM typeAbonnement AS t NATURAL JOIN abonnement AS a

WHERE MONTH(dateDebutAbonnement) = MONTH(NOW()))

// 7) Calcul des places disponibles pour une station

PREPARE nombre\_places(INTEGER) AS

SELECT s.capacite - COUNT(v.plaque)

FROM voiture AS v NATURAL JOIN station AS s

WHERE s.idStation = $1 AND v.idStation = $1;

EXECUTE nombre\_places(1);

// 8) Tous les utilisateurs dont l’abonnement se termine au maximum dans $1 jours

// Nous partons du principe que tous nos abonnements sont mensuel ou annuel.

PREPARE alerte\_abonnement(INTEGER) AS

SELECT id.Utilisateur, nom

FROM utilisateur NATURAL JOIN abonnement

WHERE DATEDIFF(dateFinAbonnement,CAST(NOW() AS DATE)) <= $1;

EXECUTE alerte\_abonnement(10);

// 9) Voiture étant apparié à une station mais n’étant pas garé sur cette station

SELECT v.plaque, s.idStation

FROM voiture AS v NATURAL JOIN station AS s

WHERE v.idStation = NOT NULL

AND v.idStation = s.idStation

AND ((SELECT l1.longitude FROM localisation AS l1 WHERE s.idLocalisation = l1.idLocalisation) != (SELECT l2.longitude FROM localisation AS l2 WHERE v.idLocalisation = l2.idLocalisation)

OR (SELECT l3.latitude FROM localisation AS l3 WHERE s.idLocalisation = l3.idLocalisation) != (SELECT l4.latitude FROM localisation AS l4 WHERE v.idLocalisation = l4.idLocalisation));

// 10) Renvoie les utilisateurs ayant dépensés plus de 150 € d’abonnement

SELECT u.idUtilisateur, u.nom, SUM(t.prixTypeAbonnement)

FROM utilisateur AS u NATURAL JOIN abonnement AS a NATURAL JOIN typeAbonnement AS t

GROUP BY u.idUtilisateur

HAVING SUM(t.prixTypeAbonnement) >= 150;