Rental Car

Company Profile

Abderrafie Elguessab
Achraf elattouaki
Bader eddine Slioui

Introduction

In this project, we focused on developing a database for a rental car company that will assist them in managing their operations efficiently. The database was designed to keep track of the company's clients, cars, rentals, and agencies.

With this database, the rental car company can easily access client information, rental histories, and agency details all in one place. This will not only save time but also reduce the chances of errors that may occur with manual record keeping.

Table of Contents

01 Creating databse

02 Creating Table

03 Inserting Data

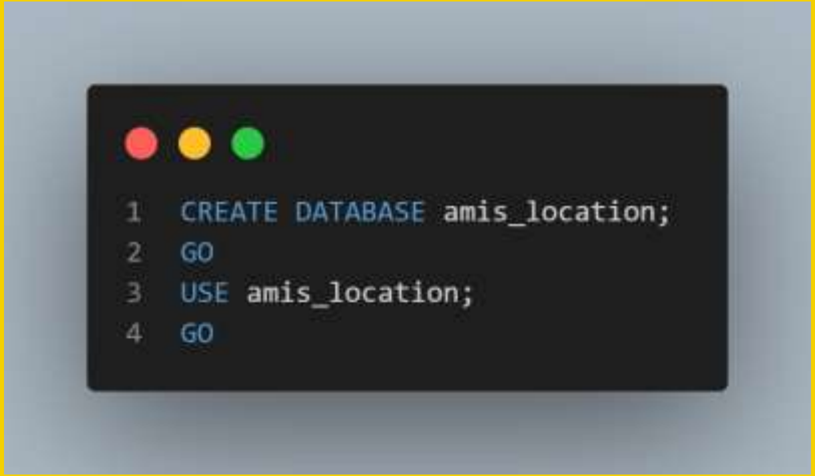
04 Merise

05 SQl & TSQl

01

DataBase Creation

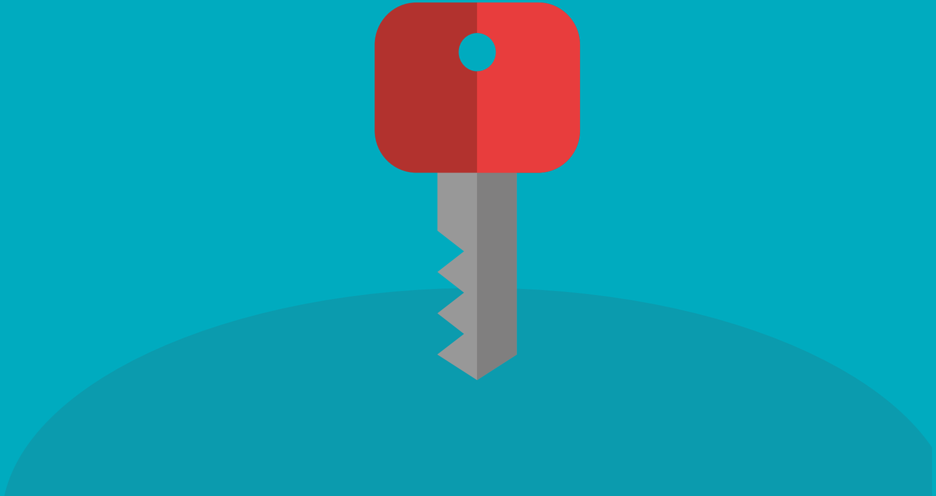
We created a DataBase called “amis location”:



```
CREATE DATABASE amis_location;
GO
USE amis_location;
GO
```

02

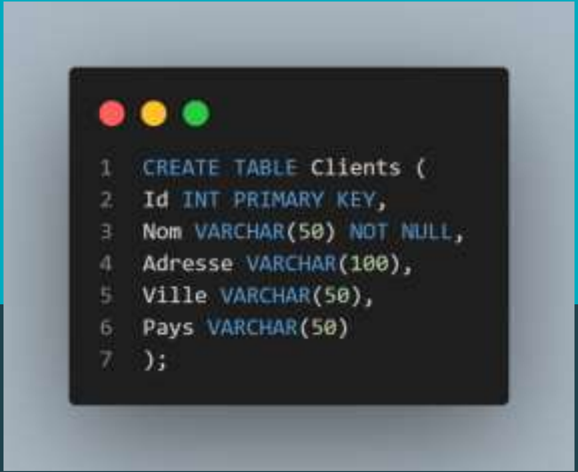
Tables Creation



HOME

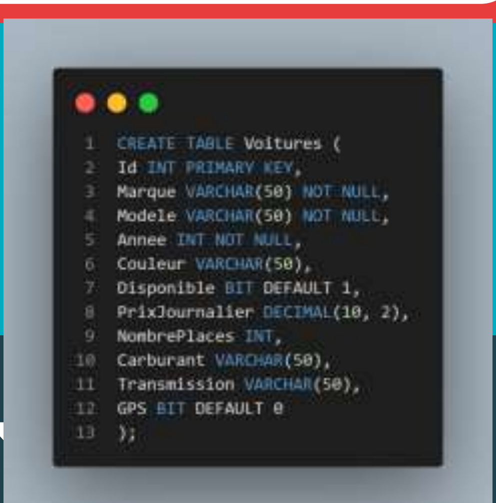
Creation des tables :

Client



```sql
CREATE TABLE Clients (
Id INT PRIMARY KEY,
Nom VARCHAR(50) NOT NULL,
Adresse VARCHAR(100),
Ville VARCHAR(50),
Pays VARCHAR(50)
);
```

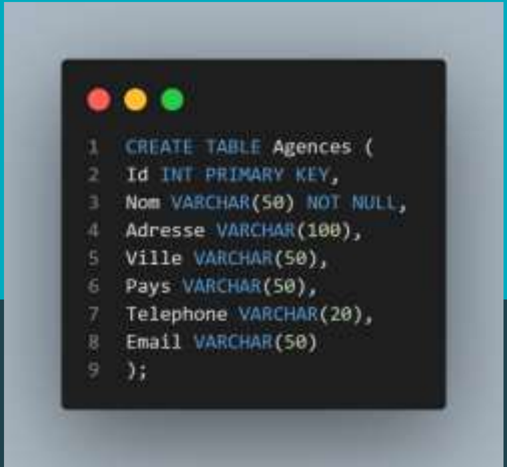
Voiture



```sql
CREATE TABLE Voitures (
Id INT PRIMARY KEY,
Marque VARCHAR(50) NOT NULL,
Modele VARCHAR(50) NOT NULL,
Annee INT NOT NULL,
Couleur VARCHAR(50),
Disponible BIT DEFAULT 1,
PrixJournalier DECIMAL(10, 2),
NombrePlaces INT,
Carburant VARCHAR(50),
Transmission VARCHAR(50),
GPS BIT DEFAULT 0
);
```

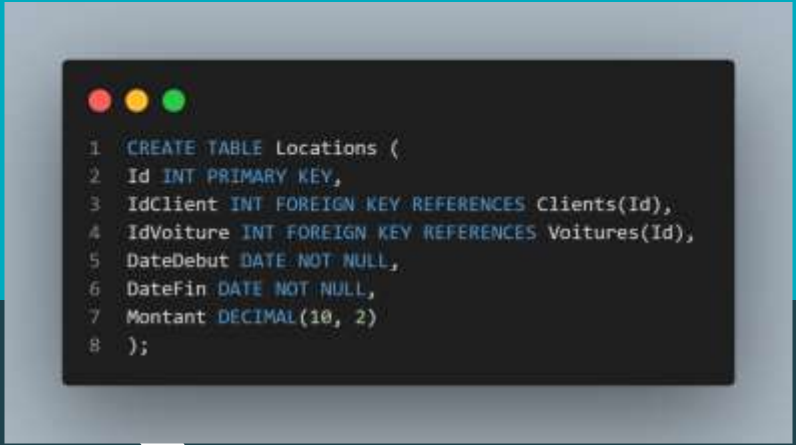
Creation des tables :

Agence



```sql
CREATE TABLE Agences (
Id INT PRIMARY KEY,
Nom VARCHAR(50) NOT NULL,
Adresse VARCHAR(100),
Ville VARCHAR(50),
Pays VARCHAR(50),
Telephone VARCHAR(20),
Email VARCHAR(50)
);
```

Location

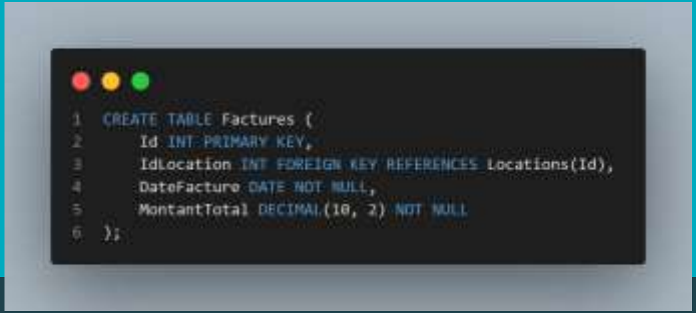


```sql
CREATE TABLE Locations (
Id INT PRIMARY KEY,
IdClient INT FOREIGN KEY REFERENCES Clients(Id),
IdVoiture INT FOREIGN KEY REFERENCES Voitures(Id),
DateDebut DATE NOT NULL,
DateFin DATE NOT NULL,
Montant DECIMAL(10, 2)
);
```

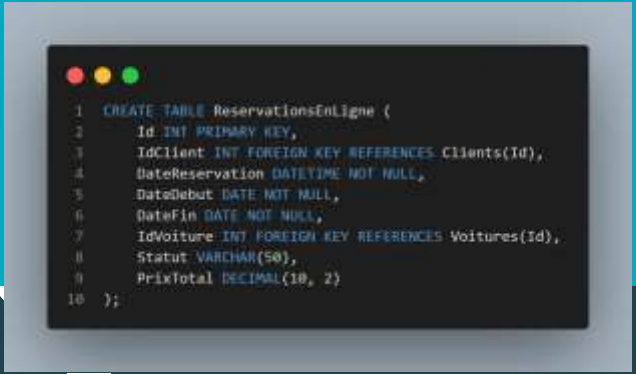
Creation des tables :

facture

Reservation en ligne

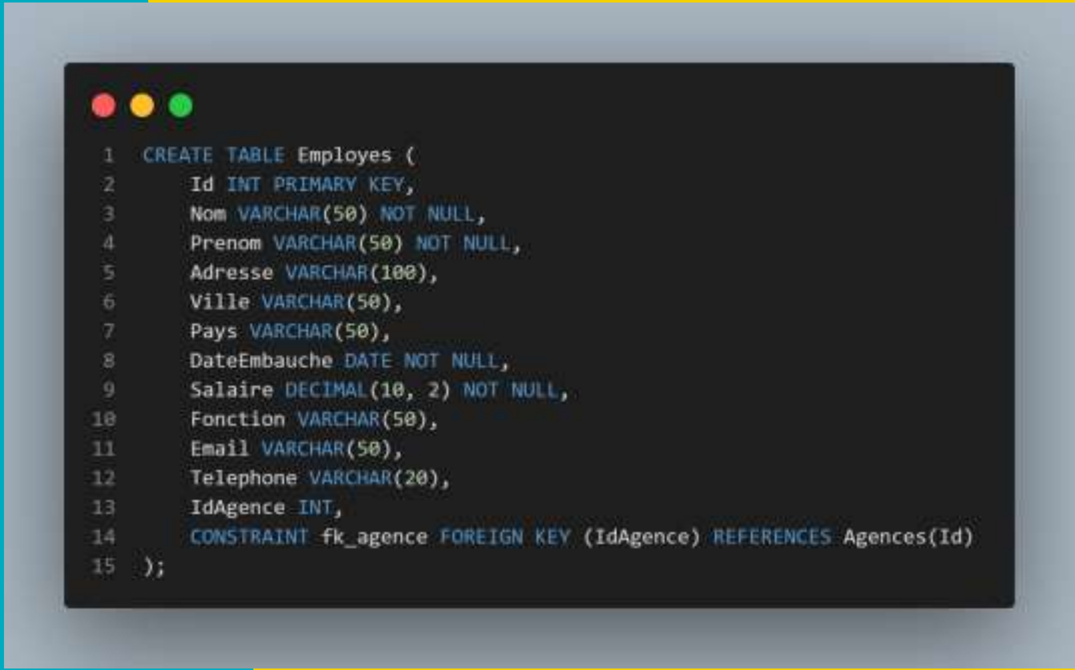


```sql
CREATE TABLE Factures (
    Id INT PRIMARY KEY,
    IdLocation INT FOREIGN KEY REFERENCES Locations(Id),
    DateFacture DATE NOT NULL,
    MontantTotal DECIMAL(10, 2) NOT NULL
);
```



```sql
CREATE TABLE ReservationsEnLigne (
    Id INT PRIMARY KEY,
    IdClient INT FOREIGN KEY REFERENCES Clients(Id),
    DateReservation DATETIME NOT NULL,
    DateDebut DATE NOT NULL,
    DateFin DATE NOT NULL,
    IdVoiture INT FOREIGN KEY REFERENCES Voitures(Id),
    Statut VARCHAR(50),
    PrixTotal DECIMAL(10, 2)
);
```

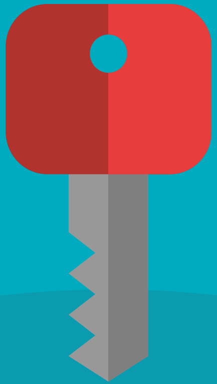
Table Employee



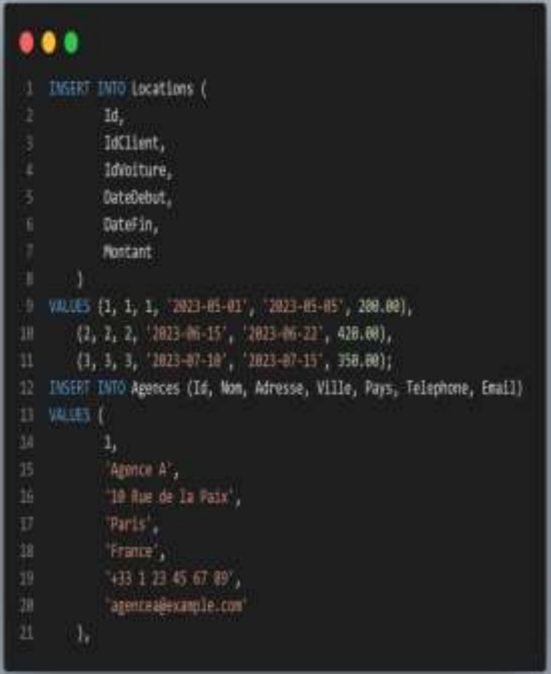
```sql
CREATE TABLE Employes (
    Id INT PRIMARY KEY,
    Nom VARCHAR(50) NOT NULL,
    Prenom VARCHAR(50) NOT NULL,
    Adresse VARCHAR(100),
    Ville VARCHAR(50),
    Pays VARCHAR(50),
    DateEmbauche DATE NOT NULL,
    Salaire DECIMAL(10, 2) NOT NULL,
    Fonction VARCHAR(50),
    Email VARCHAR(50),
    Telephone VARCHAR(20),
    IdAgence INT,
    CONSTRAINT fk_agence FOREIGN KEY (IdAgence) REFERENCES Agences(Id)
);
```

03

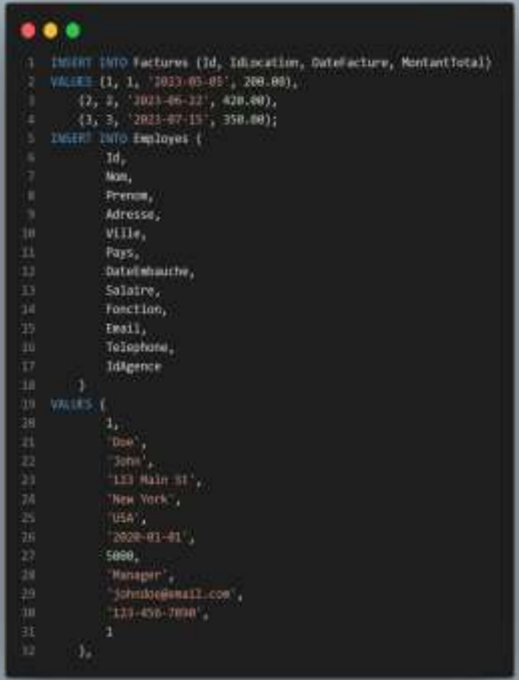
Data Insertion



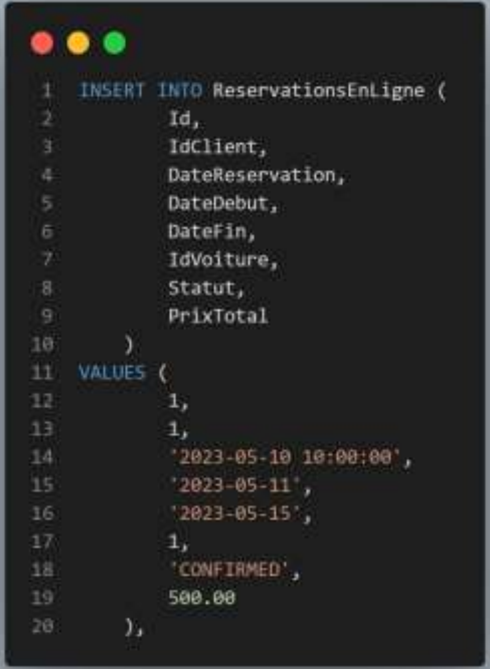
HOME



```sql
INSERT INTO Locations (
        Id,
        IdClient,
        IdVoiture,
        DateDebut,
        DateFin,
        Montant
    )
VALUES (1, 1, 1, '2023-05-01', '2023-05-05', 200.00),
    (2, 2, 2, '2023-06-15', '2023-06-22', 420.00),
    (3, 3, 3, '2023-07-10', '2023-07-15', 350.00);
INSERT INTO Agences (Id, Nom, Adresse, Ville, Pays, Telephone, Email)
VALUES (
        1,
        'Agence A',
        '10 Rue de la Paix',
        'Paris',
        'France',
        '+33 1 23 45 67 89',
        'agencea@example.com'
    ),
```



```sql
INSERT INTO Factures (Id, IdLocation, DateFacture, MontantTotal)
VALUES (1, 1, '2023-05-05', 200.00),
    (2, 2, '2023-06-22', 420.00),
    (3, 3, '2023-07-15', 350.00);
INSERT INTO Employes (
        Id,
        Nom,
        Prenom,
        Adresse,
        Ville,
        Pays,
        DateEmbauche,
        Salaire,
        Fonction,
        Email,
        Telephone,
        IdAgence
    )
VALUES (
        1,
        'Doe',
        'John',
        '123 Main St',
        'New York',
        'USA',
        '2020-01-01',
        5000,
        'Manager',
        'johndoe@email.com',
        '123-456-7890',
        1
    ),
```



```sql
INSERT INTO ReservationsEnLigne (
        Id,
        IdClient,
        DateReservation,
        DateDebut,
        DateFin,
        IdVoiture,
        Statut,
        PrixTotal
    )
VALUES (
        1,
        1,
        '2023-05-10 10:00:00',
        '2023-05-11',
        '2023-05-15',
        1,
        'CONFIRMED',
        500.00
    ),
```

04

Merise

HOME

MCD goes here

MLD

MLD

Clients (ID_Client,

Nom,

Adresse,

Ville,

Pays)

MLD

Voitures (

ID_Voiture,

Marque,

Modèle,

Année,

Couleur,

Disponible,

Prix_journalier,

Nombre_places,

Carburant,

Transmission,

GPS

)

MLD

Locations (

ID_Location,

Date_début,

Date_fin,

Montant,

ID_Client,

ID_Voiture

)

MLD

Factures (

ID_Facture,

Date_facture,

Montant_total,

ID_Location

)

MLD

Réservations (

ID_Réservation,

ID_Client,

Date_réservation,

Date_début,

Date_fin,

ID_Voiture,

Statut,

Prix_total

)

MLD

Agences (

ID_Agence,

Nom,

Adresse,

Ville,

Pays,

Téléphone,

Email

)

MLD

Employés (

ID_Employé,

Nom,

Prénom,

Adresse,

Ville,

Pays,

Date_embauche,

Salaire,

Fonction,

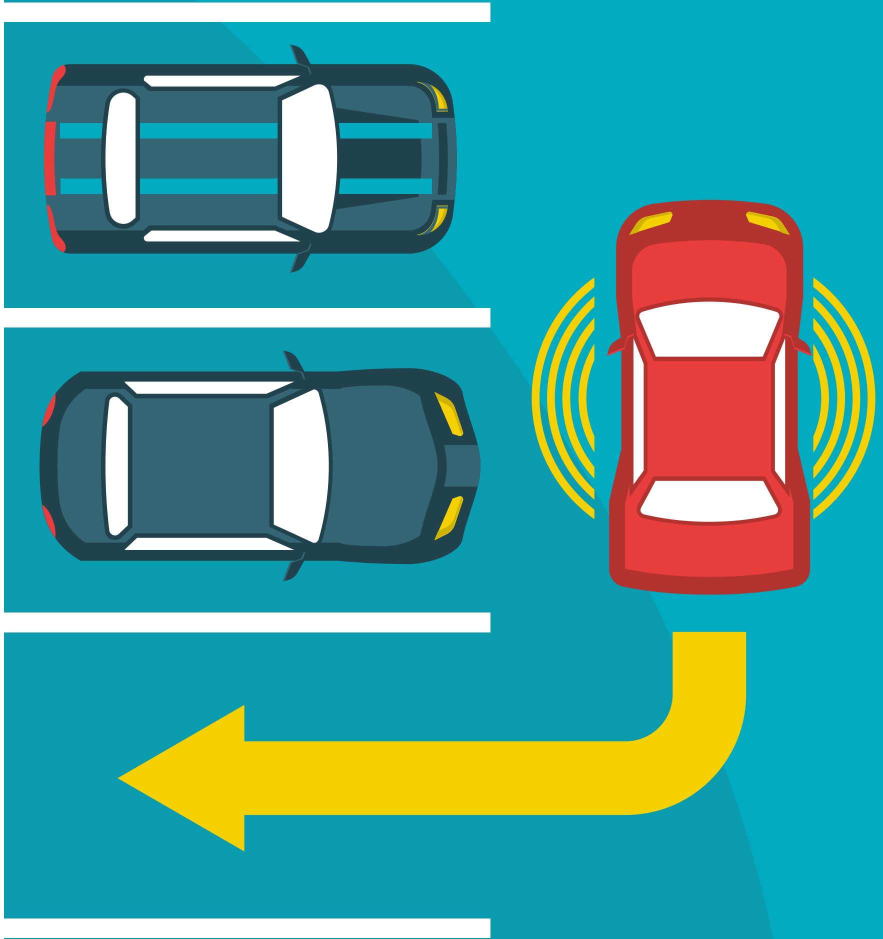
Email,

Téléphone,

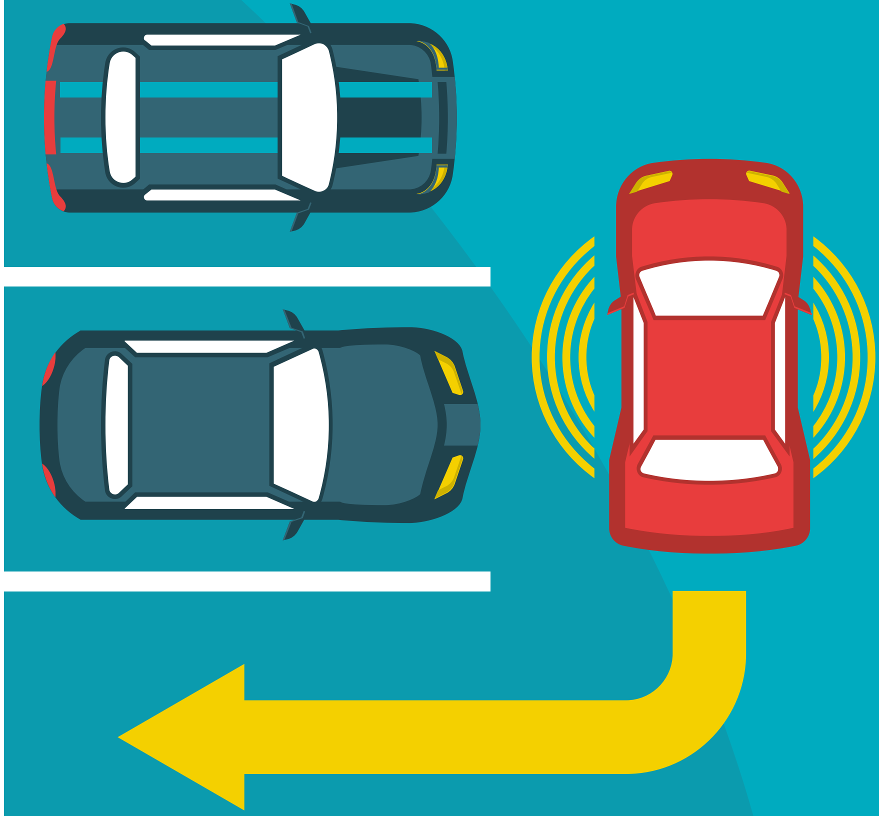
ID_Agence

)

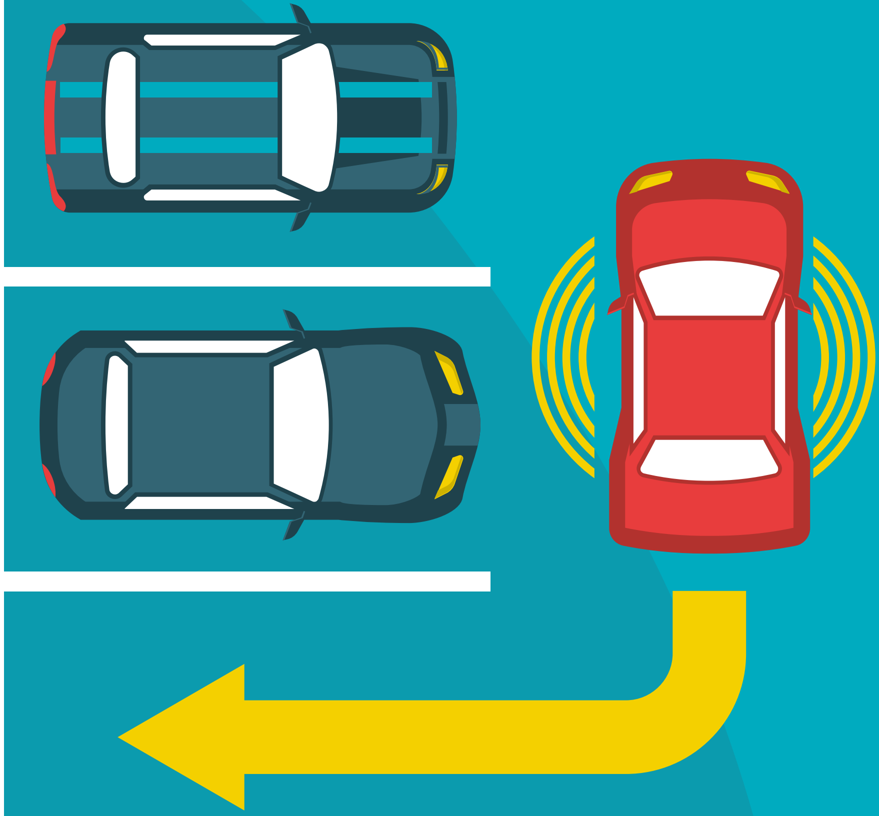
MPD



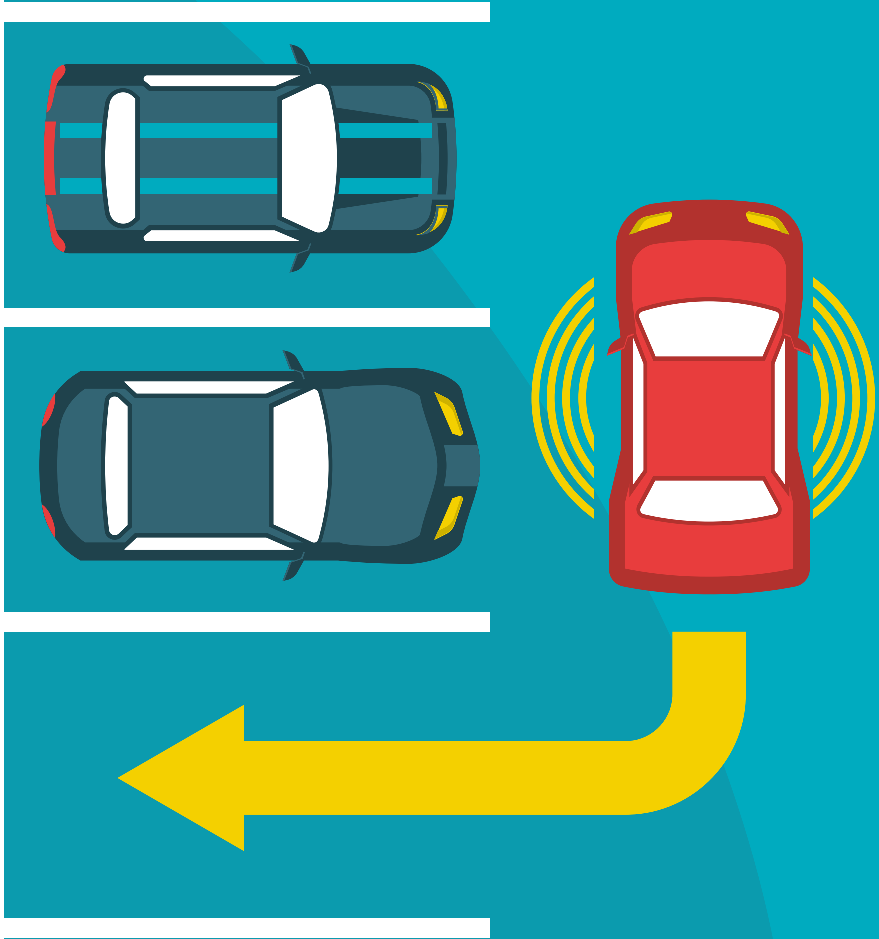
```
CLIENTS (id, nom, adresse, ville, pays)
PK: id
VOITURES (
   id,
   marque,
   modele,
   annee,
   couleur,
   disponible,
   prix_journalier,
   nombre_places,
   carburant,
   transmission,
   gps)
PK: id
```



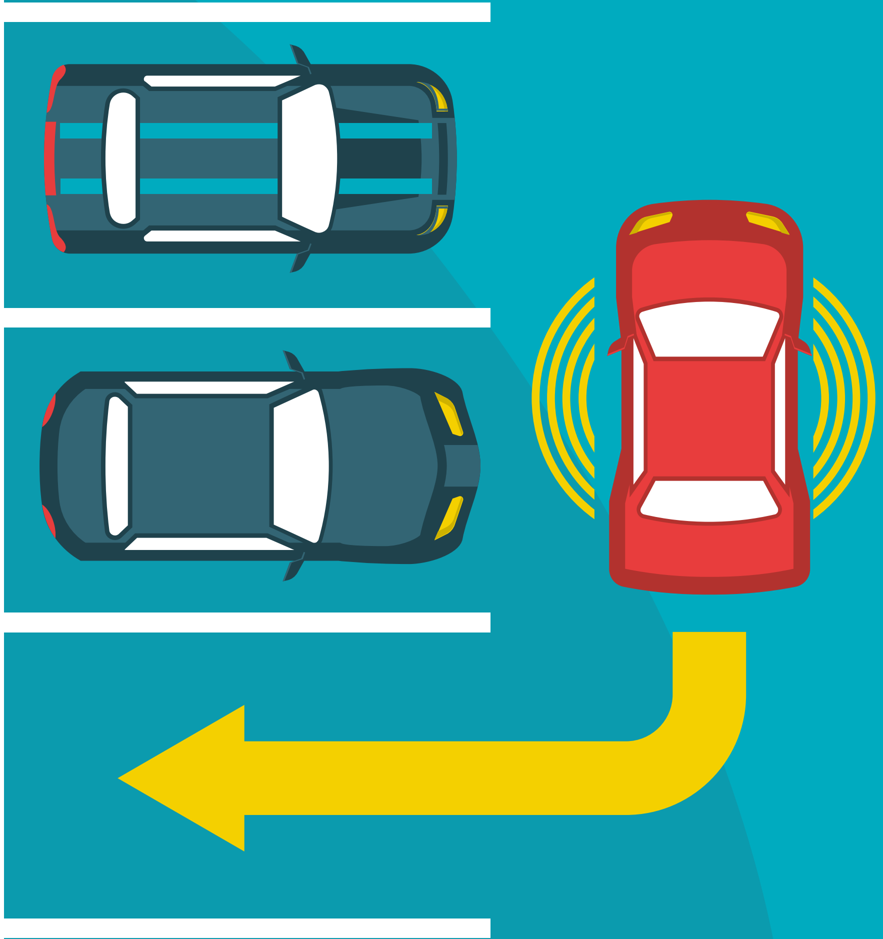
```
AGENCES (id, nom, adresse, ville, pays,
      telephone, email) PK: id
EMPLOYES (
   id,
   nom,
   prenom,
   adresse,
   ville,
   pays,
   date_embauche,
   salaire,
   fonction,
   email,
   telephone,
   id_agence
) PK: id FK: id_agence REFERENCES
      AGENCES(id)
```



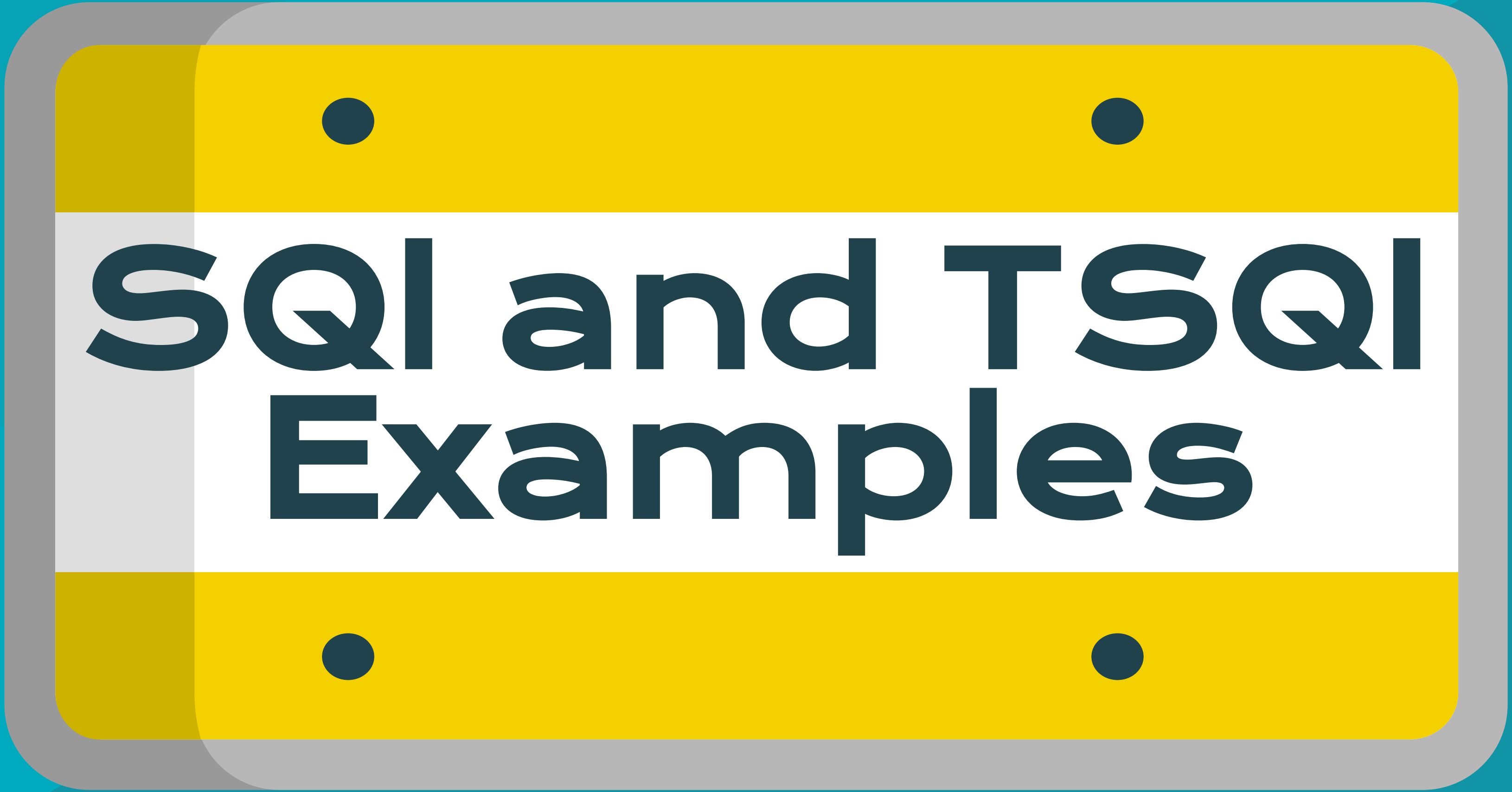
```
LOCATIONS (
   id,
   id_client,
   id_voiture,
   date_debut,
   date_fin,
   montant
)
PK: id
FK: id_client REFERENCES
      CLIENTS(id)
FK: id_voiture
      REFERENCES
      VOITURES(id)
```



```
FACTURES (
id,
id_location,
date_facture,
montant_total)

 PK: id
FK: id_location
      REFERENCES
      LOCATIONS(id)
```



```
RESERVATIONS_EN_LIGN(
   id,
   id_client,
   date_reservation,
   date_debut,
   date_fin,
   id_voiture,
   statut,
   prix_total
)
PK: id
FK: id_client REFERENCES


 CLIENTS(id) FK: id_voiture
      REFERENCES
      VOITURES(id)
```



SQl and TSQl Examples

Question: Afficher tous les clients dans la table Clients:

```
SELECT *
FROM Clients;
```

Afficher les voitures disponibles dans la table Voitures:

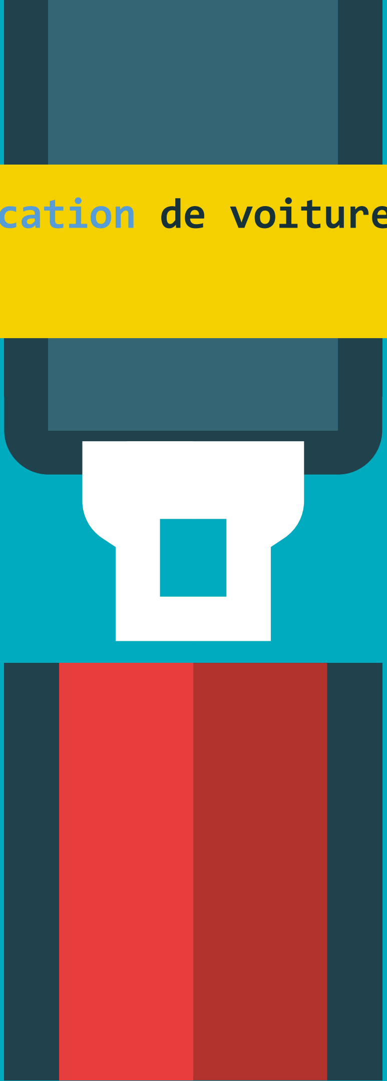
```
SELECT *
FROM Voitures
WHERE Disponible = 1;
```

Afficher les locations de voitures qui ont eu lieu entre deux dates spécifiques dans la table Locations:

```
SELECT *
FROM Locations
WHERE DateDebut BETWEEN '2023-05-
    01' AND '2023-05-05';
```

Afficher le montant total facturé pour chaque location de voiture dans la table Factures et la table Locations:

```
SELECT Locations.Id,
    SUM(Factures.MontantTot
   al) AS
   MontantTotalFacture
FROM Loc
```



Sélectionner tous les clients qui habitent dans la ville de Paris:

```
SELECT *
FROM Clients
WHERE Ville =
'Paris';
```

Afficher le nombre de voitures disponibles dans chaque agence dans la table Agences et la table Voitures:

```
SELECT Agences.Nom,
    COUNT(Voitures.Id)
AS
NombreVoituresDisponib
les
FROM Agences
    JOIN Voitures ON
Agences.Id =
Voitures.IdAgence
WHERE
Voitures.Disponible =
1
GROUP BY Agences.Nom;
```

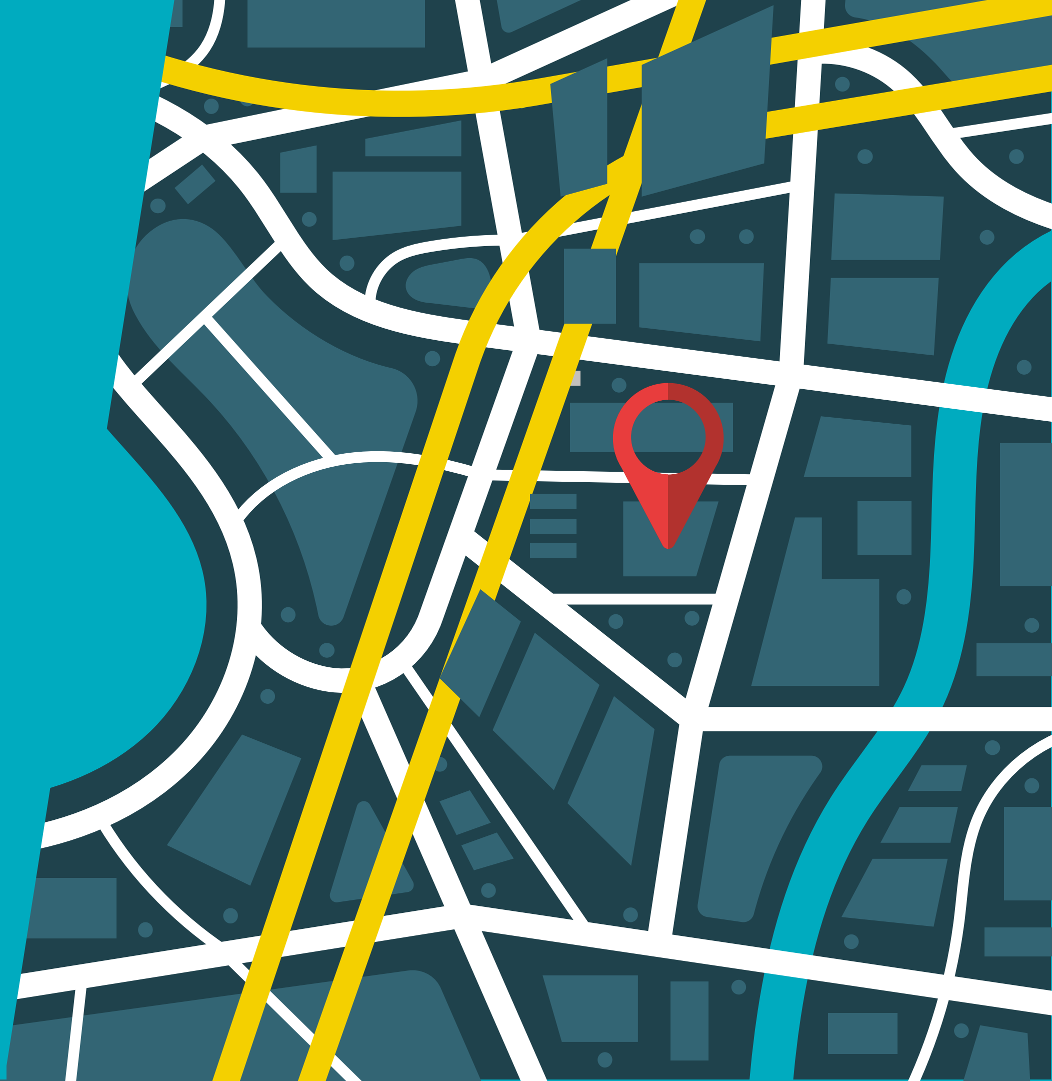
Sélectionner tous les employés qui ont été embauchés après le 1er janvier 2020:

```
SELECT *
FROM Employes
WHERE DateEmbauche > '2020-
   01-01';
```

Q: Calculate the average salary of all employees in the database.

Answer

```
DECLARE @totalSalary DECIMAL(10, 2) = 0;

DECLARE @count INT = 0;

FOR employee IN (SELECT * FROM
Employes)BEGINSET @totalSalary +=
employee.Salaire;

SET @count += 1;END;SELECT
@totalSalary/@count;
```

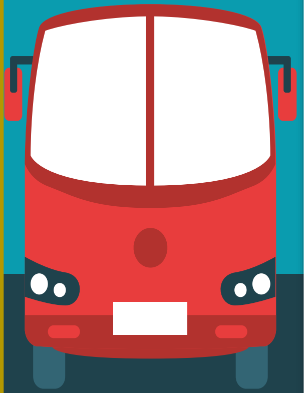


Best Display the total revenue generated by all car rentals in the month of May 2023.

Calculate the average rental duration of all car rentals in the database.

Answer

```
DECLARE @totalRevenue
DECIMAL(10, 2) = 0;

FOR location IN (SELECT *
FROM Locations ;

WHERE MONTH(DateDebut) = 5
AND YEAR(DateDebut) = 2023)

BEGINSET @totalRevenue +=
location.Montant;

END;

SELECT @totalRevenue;
```



Answer

```
DECLARE @totalDuration INT = 0;

DECLARE @count INT = 0;

FOR location IN (SELECT * FROM
Locations)

BEGINSET @totalDuration +=
DATEDIFF(DAY,
location.DateDebut,
location.DateFin);

SET @count += 1;END;SELECT
@totalDuration/@count;
```

Print Client Information

Exemple explained in video!

1

```
DECLARE @ClientId INT,
        @ClientNom VARCHAR(50),
        @ClientAdresse VARCHAR(100),
        @ClientVille VARCHAR(50),
        @ClientPays VARCHAR(50);

-- Declare the cursor
DECLARE client_cursor CURSOR FOR
SELECT Id, Nom, Adresse, Ville, Pays
FROM Clients;

-- Open the cursor
OPEN client_cursor;

-- Fetch the first row
FETCH NEXT FROM client_cursor INTO @ClientId,
@ClientNom, @ClientAdresse, @ClientVille,
@ClientPays;
-- Start the loop
WHILE @@FETCH_STATUS = 0
BEGIN
```

MON	9AM TO 3PM
TUE	8AM TO 4PM
WED	8AM TO 4PM
THU	8AM TO 4PM
FRI	10AM TO 1PM

Print Client Information

Exemple explained in video!

```
-- Print the client information
   PRINT 'Client ID: ' + CONVERT(VARCHAR(10),
@ClientId);
   PRINT 'Client Name: ' + @ClientNom;
   PRINT 'Client Address: ' + ISNULL(@ClientAdresse, '');
   PRINT 'Client City: ' + ISNULL(@ClientVille, '');
   PRINT 'Client Country: ' + ISNULL(@ClientPays, '');
   PRINT '';

   -- Fetch the next row
   FETCH NEXT FROM client_cursor INTO @ClientId,
@ClientNom, @ClientAdresse, @ClientVille,
@ClientPays;
END

-- Close the cursor
CLOSE client_cursor;

-- Deallocate the cursor
DEALLOCATE client_cursor;
```

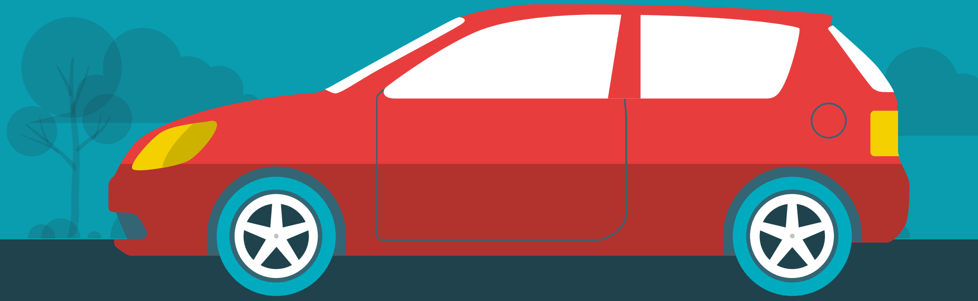
MON	9AM TO 3PM
TUE	8AM TO 4PM
WED	8AM TO 4PM
THU	8AM TO 4PM
FRI	10AM TO 1PM

Triggers example :

Triggers that update avg salary :

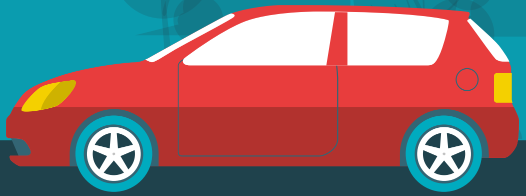
Exemple explained in video!

```
CREATE TRIGGER
UpdateReservationCount ON
LocationsAFTER INSERT, UPDATE,
DELETEASBEGIN
-- Mettre à jour le nombre de réservations
pour chaque voiture
UPDATE Voitures    SET
NombreReservations = (        SELECT
COUNT(*) FROM Locations WHERE
Locations.IdVoiture = Voitures.Id     )
FROM Voitures    INNER JOIN (
SELECT IdVoiture FROM inserted
UNION        SELECT IdVoiture FROM
deleted    ) AS Changes ON Voitures.Id =
Changes.IdVoiture;END;GO
```



Triggers that update nombre reservation :

```
CREATE TRIGGER
UpdateNombreReservations AFTER
INSERT, DELETE ON
ReservationsEnLigne FOR EACH
ROW BEGIN UPDATE Clients SET
NombreReservations = ( SELECT
COUNT(*) FROM
ReservationsEnLigne WHERE
IdClient = NEW.IdClient ) WHERE Id
= NEW.IdClient; END;
```



Procedure example :

Procedure that insert values into the database client

```
-- Create the stored procedure
CREATE PROCEDURE ProcessClients
AS
BEGIN
    -- Declare variables
    DECLARE @ClientId INT,
            @ClientNom VARCHAR(50),
            @ClientAdresse VARCHAR(100),
            @ClientVille VARCHAR(50),
            @ClientPays VARCHAR(50);

    -- Declare the cursor
    DECLARE client_cursor CURSOR FOR
    SELECT Id, Nom, Adresse, Ville, Pays
    FROM Clients;
    -- Open the cursor
    OPEN client_cursor;
```

Procedure that insert values into the database client

```
 -- Fetch the first row
     FETCH NEXT FROM client_cursor INTO
       @ClientId, @ClientNom, @ClientAdresse,
       @ClientVille, @ClientPays;
  -- Start the loop
     WHILE @@FETCH_STATUS = 0
     BEGIN
         -- Perform operations using the fetched data

         -- Print the client information
         PRINT 'Client ID: ' +
       CONVERT(VARCHAR(10), @ClientId);
         PRINT 'Client Name: ' + @ClientNom;
         PRINT 'Client Address: ' +
       ISNULL(@ClientAdresse, '');
         PRINT 'Client City: ' + ISNULL(@ClientVille,
       '');
         PRINT 'Client Country: ' +
       ISNULL(@ClientPays, '');
```

Procedure that insert values into the database client

```
        -- Fetch the next row
        FETCH NEXT FROM client_cursor
      INTO @ClientId, @ClientNom,
      @ClientAdresse, @ClientVille,
      @ClientPays;
    END

    -- Close the cursor
    CLOSE client_cursor;

    -- Deallocate the cursor
    DEALLOCATE client_cursor;
END
```

Merci