

Campus Henry Christophe de l'Université d'État d'Haïti à Limonade



(CHC-UEH-L)



Faculté des Sciences et de la Génies

(FSG)

Devoir du cours : Introduction aux bases de données

Professeur : Andy PIERRE

Préparé par : Blemy JOSEPH

Faculté : Sciences Informatiques

Niveau : Licence 3

Date de remise : 15 / 07 / 2024

PARTIE 4.1 : DEFINITION DE DONNEES (DDL)

1. Donner le script de création de la base de données dans les cas suivants :

1.1. Création de la base de données « TRANSPORT »

```
CREATE DATABASE TRANSPORT ;
```

1.2. Créer les tables sans contraintes de clés primaires et de clés étrangères.

Création de la table Personnel

```
CREATE TABLE Personnel ( codePersonnel int(3),  
  
                           Prenom char(15),  
  
                           Nom char(20),  
  
                           dateNaissance Date,  
  
                           dateembauche Date,  
  
                           Typepersonnel char(10),  
  
                           salaire decimal(10, 2),  
  
                           commission decimal(10, 2)  
  
                           );
```

Création de la table Destination

```
CREATE TABLE Destination ( DestinationId int (3),  
  
                           Ville char(15),  
  
                           departement char(15),  
  
                           Distance decimal(10, 2)  
  
                           );
```

Création de la table Compagnie

```
CREATE TABLE Compagnie ( Code_CompagnieId int (3),  
  
                           Nom char(15),
```

slogan char(50)

);

Création de la table Bus

CREATE TABLE Bus (BusId int(3),

Code_CompagnieId int (3),

NoImma char(6),

Vitesse decimal(10, 2),

capacite int(2)

);

Création de la table Voyage

CREATE TABLE Voyage (VoyageId int(3),

BusId int(3),

DestinationId int (3),

DateDepart Date,

HeureDepart time,

HeureArrive time

);

Création de la table Reservation

CREATE TABLE Reservation (ReservationId int(3),

VoyageId int(3),

Nom char (20),

Prix decimal(10, 2),

Reduction decimal(10, 2),

datereservation date);

Création de la table Equipage

```
CREATE TABLE Equipage ( VoyageId int(3),  
  
                        codePersonnel int(3)  
  
                        );
```

1.3 Implanter les contraintes de clés primaires et de clés étrangères à partir d'ALTER TABLE.

Implantation des contraintes des clés primaires pour les tables

- alter table Personnel add constraint pk_codePersonnel primary
key(codePersonnel) ;
- alter table Destination add constraint pk_DestinationId primary
key(DestinationId);
- alter table Compagnie add constraint pk_Code_CompagnieId primary
key(Code_CompagnieId) ;
- alter table Bus add constraint pk_BusId primary key(BusId);
- alter table Voyage add constraint pk_VoyageId primary key(VoyageId);
- alter table Reservation add constraint pk_ReservationId primary
key(ReservationId);

Implantation des contraintes des clés étrangères pour les tables

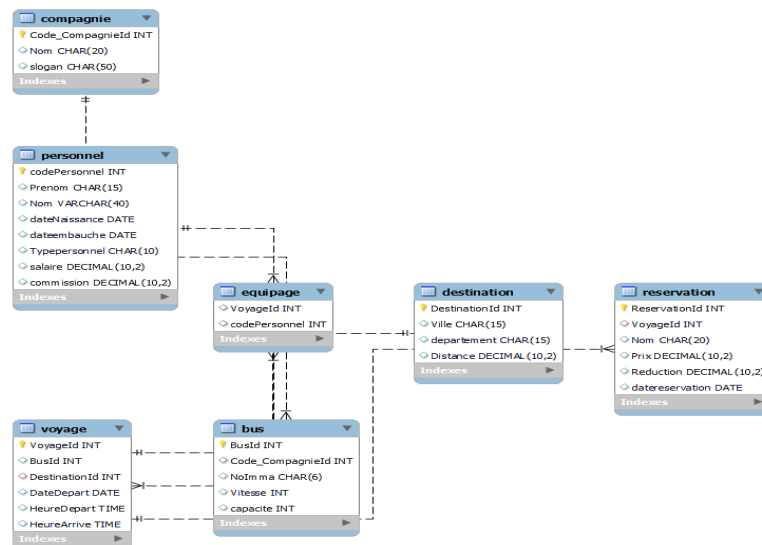
- alter table Bus add constraint fk_Code_CompagnieId foreign
key(Code_CompagnieId) references Compagnie(Code_CompagnieId);
- alter table Voyage add constraint fk_BusId foreign key(BusId) references
Bus(BusId);
- alter table Voyage add constraint fk_DestinationId foreign
key(DestinationId) references Destination(DestinationId);

- alter table Reservation add constraint fk_VoyageId foreign key(VoyageId) references Voyage(VoyageId);
- alter table Equipage add constraint fk_VoyageId2 foreign key(VoyageId) references Voyage(VoyageId);
- alter table Equipage add constraint fk_code_Personnel foreign key(codePersonnel) references Personnel(codePersonnel);

2. Donner le dictionnaire de données de chaque relation du schéma relationnel

| Attribut | Rôle | Domaine |
|---------------------------|--------------------------------|----------------|
| codePersonnel | Identifiant du personnel | Entier (3) |
| Prenom | Prénom du personnel | Chaine (15) |
| Nom | Nom du personnel | Chaine (20) |
| dateNaissance | Date de naissance du personnel | Date |
| dateembauche | Date embauche du personnel | Date |
| Typepersonnel char(10), | Type du personnel | Chaine (10) |
| salaire decimal(10, 2), | Salaire du personnel | Decimal (10,2) |
| commission decimal(10, 2) | Commission du personnel | Decimal (10,2) |

3. Donner le schéma relationnel de la base donnée



4. Donner la requête SQL permettant de modifier certains types de données précédemment définis.

a. Vitesse devient Int (10)

```
alter table Bus modify Vitesse Int(10);
```

b. Nom devient varchar (40)

```
alter table Personnel modify Nom varchar(40);
```

5. Ecrire les requêtes SQL permettant d'ajouter deux nouvelles colonnes adresse « not null », téléphone « unique » à la table personnel.

```
➤ alter table Personnel add column adresse char(15) not null;
```

```
➤ alter table Personnel add column telephone char(8) unique;
```

6. Ecrivez les commandes permettant la suppression des contraintes clés primaires et étrangères.

La suppression des contraintes clés étrangères

```
➤ ALTER TABLE Bus DROP FOREIGN KEY fk_Code_CompagnieId;
```

```
➤ ALTER TABLE Voyage DROP FOREIGN KEY fk_BusId;
```

```
➤ ALTER TABLE Voyage DROP FOREIGN KEY fk_DestinationId;
```

```
➤ ALTER TABLE Reservation DROP FOREIGN KEY fk_VoyageId;
```

```
➤ ALTER TABLE Equipage DROP FOREIGN KEY fk_VoyageId;
```

```
➤ ALTER TABLE Equipage DROP FOREIGN KEY fk_code_Personnel;
```

La suppression des contraintes clés primaires

```
➤ alter table Personnel DROP primary key;
```

```
➤ alter table Destination DROP primary key;
```

```
➤ alter table Compagnie DROP primary key;
```

```
➤ alter table Bus DROP primary key;
```

- alter table Voyage DROP primary key;
- alter table Reservation DROP primary key;

7. Ecrivez les commandes permettant de remettre en place les contraintes clés primaires et étrangères sans recréer les tables.

Contraintes des clés primaires

- alter table Personnel add constraint pk_codePersonnel primary key(codePersonnel) ;
- alter table Destination add constraint pk_DestinationId primary key(DestinationId);
- alter table Compagnie add constraint pk_Code_CompagnieId primary key(Code_CompagnieId) ;
- alter table Bus add constraint pk_BusId primary key(BusId);
- alter table Voyage add constraint pk_VoyageId primary key(VoyageId);
- alter table Reservation add constraint pk_ReservationId primary key(ReservationId);

Contraintes des clés étrangères

- alter table Bus add constraint fk_Code_CompagnieId foreign key(Code_CompagnieId) references Compagnie(Code_CompagnieId);
- alter table Voyage add constraint fk_BusId foreign key(BusId) references Bus(BusId);
- alter table Voyage add constraint fk_DestinationId foreign key(DestinationId) references Destination(DestinationId);
- alter table Reservation add constraint fk_VoyageId foreign key(VoyageId) references Voyage(VoyageId);

- alter table Equipage add constraint fk_VoyageId foreign key(VoyageId) references Voyage(VoyageId);
- alter table Equipage add constraint fk_code_Personnel foreign key(code_Personnel) references Personnel(code_Personnel);

PARTIE 4.2 EXPRIMEZ LES REQUETES SUIVANTES EN ALGEBRE RELATIONNELLE

1. Donner le numéro, le nom et le salaire de chaque chauffeur ayant participé à plus que 11 voyages.

- $\pi \text{ codePersonnel, Nom, salaire } (\sigma \text{ nombre_voyage} > 11 \wedge \text{Typepersonnel} = \text{'Chauffeur'} (\text{Personnel} \bowtie \text{Equipage}))$

2. Lister les employés qui ont été embauchés entre janvier à décembre 2023

- $\pi \text{ Personnel* } (\sigma \text{ dateembauche} \geq \text{'2023-01-01'} \wedge \text{dateembauche} \leq \text{'2023-12-31'} (\text{Personnel}))$

3. Quels sont les employés habitant Cap-Haitien

- $\pi \text{ Personnel* } (\sigma \text{ adresse} = \text{'Cap-Haitien'} (\text{Personnel}))$

4. Quel est l'employé le plus ancien ?

- $\pi \text{ Personnel* } (\sigma \text{ dateembauche} = \min(\text{dateembauche}) (\text{Personnel}))$

5. Lister les noms et les salaires employés mieux payés que «Jacques Jean Baptiste»

- $\pi \text{ Nom, salaire } (\sigma \text{ salaire} > (\pi \text{ salaire } (\sigma \text{ Prenom} = \text{'Jacques'} \wedge \text{Nom} = \text{'Jean Baptiste'}) (\text{Personnel})) (\text{Personnel}))$

6. Donnez le nom de toutes les compagnies possédant au moins un bus de même type que la compagnie d'identifiant 3.

➤ $\pi \text{ Nom } (\sigma \text{ Code_CompagnieId } \in (\pi \text{ Code_CompagnieId } (\sigma \text{ Code_CompagnieId } = 3 (\text{Bus})))) (\text{Bus} \bowtie \text{Compagnie}))$

7. Donnez pour chaque compagnie la liste des bus qu'elle possède, classée par ordre de capacité croissante.

➤ $\pi \text{ Nom, BusId, Capacité } (\sigma \text{ Capacité_par_ordre_croissante}) (\text{Bus} \bowtie \text{Compagnie})$

8. Informations sur les personnels plus âgés que 40 années

➤ $\pi \text{ Personnel* } (\sigma \text{ dateNaissance } < '1984-01-01' (\text{Personnel}))$

PARTIE 5.3 : MANIPULATION DE DONNEES (DML)

1. Donner les couples chauffeurs qui gagnent le même salaire.

➤ `SELECT p1.Prenom, p1.Nom, p2.Prenom, p2.Nom, p1.salaire FROM Personnel p1, Personnel p2 WHERE p1.salaire = p2.salaire AND p1.codePersonnel < p2.codePersonnel AND p1.Typepersonnel = 'Chauffeur' AND p2.Typepersonnel = 'Chauffeur';`

```
mysql> SELECT p1.Prenom, p1.Nom, p2.Prenom, p2.Nom, p1.salaire FROM Personnel p1, Personnel p2 WHERE p1.salaire = p2.salaire AND p1.codePersonnel < p2.codePersonnel;
+-----+-----+-----+-----+-----+
| Prenom | Nom   | Prenom | Nom   | salaire |
+-----+-----+-----+-----+-----+
| Emmanuel | Jedy | Louvensky | Jean | 4030.00 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

2. Donner le numéro, le nom et le salaire de chaque chauffeur ayant participé à plus que 11 voyages.

➤ `SELECT Personnel.codePersonnel, Personnel.Prenom, Personnel.Nom, Personnel.salaire FROM Personnel, Equipage WHERE Personnel.codePersonnel = Equipage.codePersonnel AND Personnel.Typepersonnel = 'Chauffeur' GROUP BY`

Personnel.codePersonnel, Personnel.Prenom, Personnel.Nom,
Personnel.salaire HAVING COUNT(Equipage.VoyageId) > 11;

```
mysql> SELECT Personnel.codePersonnel, Personnel.Prenom, Personnel.Nom, Personnel.salaire FROM Personnel, Equipage WHERE
Personnel.codePersonnel = Equipage.codePersonnel AND Personnel.Typepersonnel = 'Chauffeur' GROUP BY Personnel.codePerso
nnel, Personnel.Prenom, Personnel.Nom, Personnel.salaire HAVING COUNT(Equipage.VoyageId) > 11;
Empty set (0.00 sec)

mysql>
```

3. Lister les employés qui ont été embauchés entre Mars et septembre 2023

- Select * from Personnel where dateembauche between '2023-03-01' and '2023-09-30';

```
mysql> Select * from Personnel where dateembauche between '2023-03-01' and '2023-09-30';
```

| | codePersonnel | Prenom | Nom | dateNaissance | dateembauche | Typepersonnel | salaire | commission | adresse | telephone |
|---|---------------|---------------|------------|---------------|--------------|---------------|---------|-------------|----------|-----------|
| 1 | Jacques | Jean Baptiste | 1980-01-15 | 2023-05-20 | Chauffeur | 3250.00 | 200.00 | Cap-Haitien | 35359939 | |
| 2 | Lesly | JEAN PIERRE | 1975-02-20 | 2023-06-15 | Chauffeur | 3510.00 | 150.00 | Cap-Haitien | 33571738 | |

```
2 rows in set (0.02 sec)

mysql>
```

4. Quels sont les employés habitant Cap-Haitien.

- Select * from Personnel where adresse = 'Cap-Haitien';

```
mysql> Select * from Personnel where adresse = 'Cap-Haitien';
```

| | codePersonnel | Prenom | Nom | dateNaissance | dateembauche | Typepersonnel | salaire | commission | adresse | telephone |
|----|---------------|---------------|------------|---------------|--------------|---------------|---------|-------------|----------|-----------|
| 1 | Jacques | Jean Baptiste | 1980-01-15 | 2023-05-20 | Chauffeur | 3250.00 | 200.00 | Cap-Haitien | 35359939 | |
| 2 | Lesly | JEAN PIERRE | 1975-02-20 | 2023-06-15 | Chauffeur | 3510.00 | 150.00 | Cap-Haitien | 33571738 | |
| 7 | Louvinsky | Jean | 1995-07-15 | 2018-11-20 | Chauffeur | 4030.00 | 270.00 | Cap-Haitien | 35515001 | |
| 8 | Arisno | Norwil | 1983-08-10 | 2009-12-30 | Chauffeur | 2665.00 | 220.00 | Cap-Haitien | 35519991 | |
| 10 | Blemy | Joseph | 1985-10-10 | 2024-02-20 | Directeur | 3315.00 | 200.00 | Cap-Haitien | 30229991 | |

```
5 rows in set (0.00 sec)

mysql>
```

5. Quel est l'employé le plus ancien ?

- SELECT Prenom, Nom, dateembauche FROM Personnel ORDER BY dateembauche ASC LIMIT 1;

```
mysql> SELECT Prenom, Nom, dateembauche FROM Personnel ORDER BY dateembauche ASC LIMIT 1;
+-----+-----+-----+
| Prenom | Nom   | dateembauche |
+-----+-----+-----+
| Dieuleste | Ledix | 2008-09-10    |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

6. Lister les noms et les salaires employés mieux payés que « Jacques Jean Baptiste ».

- SELECT Prenom, Nom, salaire FROM Personnel WHERE salaire > (SELECT salaire FROM Personnel WHERE Prenom = 'Jacques' AND Nom = 'Jean Baptiste');

```
mysql> SELECT Prenom, Nom, salaire FROM Personnel WHERE salaire > (SELECT salaire FROM Personnel WHERE Prenom = 'Jacques' AND Nom = 'Jean Baptiste');
+-----+-----+-----+
| Prenom | Nom       | salaire |
+-----+-----+-----+
| Lesly  | JEAN PIERRE | 3510.00 |
| Emmanuel | Judy      | 4030.00 |
| Walnes | Colas      | 4160.00 |
| Debora | Larose     | 3380.00 |
| Louvensky | Jean     | 4030.00 |
| Richkard | Gue       | 3315.00 |
| Blémy  | Joseph     | 8315.00 |
+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> |
```

7. Donnez le nom de toutes les compagnies possédant au moins un bus de même type que la compagnie d'identifiant 3.

- SELECT DISTINCT Compagnie.Nom FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId AND Bus.capacite IN (SELECT capacite FROM Bus WHERE Code_CompagnieId = 3);

```
mysql> SELECT DISTINCT Compagnie.Nom FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId AND Bus.capacite IN (SELECT capacite FROM Bus WHERE Code_CompagnieId = 3);
+-----+
| Nom       |
+-----+
| Flamboyant |
| TravelFast |
+-----+
2 rows in set (0.04 sec)

mysql> |
```

8. *Donnez pour chaque compagnie la liste des bus qu'elle possède, classée par ordre de capacité croissante.*

- `SELECT Compagnie.Nom, Bus.NoImma, Bus.capacite FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId ORDER BY Bus.capacite ASC;`

```
mysql> SELECT Compagnie.Nom, Bus.NoImma, Bus.capacite FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId ORDER BY Bus.capacite ASC;
```

| Nom | NoImma | capacite |
|--------------|--------|----------|
| ColombeRide | FG6789 | 40 |
| LeopardBus | BC2345 | 45 |
| Sunshine | HI8901 | 45 |
| Transporteur | AB1234 | 50 |
| VIPBus | EF5678 | 50 |
| JoyTravel | DE4567 | 55 |
| ComfortBus | IJ9012 | 55 |
| Flamboyant | CD3456 | 60 |
| TravelFast | JK0123 | 60 |
| ExpressLine | GH7890 | 65 |

10 rows in set (0.00 sec)

```
mysql> |
```

9. *Informations sur les personnels plus âgés que 40 années.*

- `Select * from Personnel where dateNaissance < '1984-01-01';`

```
mysql> Select * from Personnel where dateNaissance < '1984-01-01';
```

| codePersonnel | Prenom | Nom | dateNaissance | dateembauche | Typepersonnel | salaire | commission | adresse | telephone |
|---------------|-----------|---------------|---------------|--------------|---------------|---------|------------|-------------|-----------|
| 1 | Jacques | Jean Baptiste | 1980-01-15 | 2023-05-20 | Chauffeur | 2500.00 | 200.00 | Cap-Haitien | 35359939 |
| 2 | Lesly | JEAN PIERRE | 1975-02-20 | 2023-06-15 | Chauffeur | 2700.00 | 150.00 | Cap-Haitien | 33571738 |
| 5 | Dieuleste | Ledix | 1982-05-30 | 2008-09-10 | Chauffeur | 2800.00 | 100.00 | Vaudreuil | 42315901 |
| 6 | Debora | Larose | 1978-06-05 | 2011-10-12 | Chauffeur | 3000.00 | 180.00 | Limbe | 40515901 |
| 8 | Arisno | Normil | 1983-08-10 | 2009-12-30 | Chauffeur | 2050.00 | 220.00 | Cap-Haitien | 35519991 |

5 rows in set (0.01 sec)

```
mysql> _
```

10. *Ecrire la requête qui affiche pour chaque personnel le nom, le salaire, la commission ainsi que le gain annuel.*

- `SELECT Nom, salaire, commission, (salaire * 12 + commission) AS gain_annuel FROM Personnel;`

```
mysql> SELECT Nom, salaire, commission, (salaire * 12 + commission) AS gain_annuel FROM Personnel;
```

| Nom | salaire | commission | gain_annuel |
|---------------|---------|------------|-------------|
| Jean Baptiste | 3250.00 | 200.00 | 39200.00 |
| JEAN PIERRE | 3510.00 | 150.00 | 42270.00 |
| Jeudy | 4030.00 | 210.00 | 48570.00 |
| Colas | 4160.00 | 300.00 | 50220.00 |
| Ledix | 1690.00 | 100.00 | 20380.00 |
| Larose | 3300.00 | 180.00 | 40740.00 |
| Jean | 4030.00 | 270.00 | 48630.00 |
| Normil | 2665.00 | 220.00 | 32200.00 |
| Gue | 3315.00 | 230.00 | 40010.00 |
| Joseph | 8315.00 | 200.00 | 99980.00 |

10 rows in set (0.00 sec)

```
mysql> |
```

11. Insérer un jeu de données cohérent dans vos relations (10 enregistrements par relation).

➤ **Insertion dans la table Personnel**

```
INSERT INTO Personnel (codePersonnel, Prenom, Nom, dateNaissance,  
dateembauche, Typepersonnel, salaire, commission, adresse, telephone)  
VALUES
```

```
(1, 'Jacques', 'Jean Baptiste', '1980-01-15', '2023-05-20', 'Chauffeur',  
3250.00, 200.00, 'Cap-Haitien', '35359939'),
```

```
(2, 'Lesly', 'JEAN PIERRE', '1975-02-20', '2023-06-15', 'Chauffeur',  
3510.00, 150.00, 'Cap-Haitien', '33571738'),
```

```
(3, 'Emmanuel', 'Jeudy', '1988-03-10', '2015-07-18', 'Chauffeur', 4030.00,  
210.00, 'Limonade', '41415909'),
```

```
(4, 'Walnes', 'Colas', '1990-04-25', '2017-08-22', 'Chauffeur', 4160.00,  
300.00, 'Caracol', '41415901'),
```

```
(5, 'Dieuleste', 'Ledix', '1982-05-30', '2008-09-10', 'Chauffeur', 1690.00,  
100.00, 'Vaudreuil', '42315901'),
```

```
(6, 'Debora', 'Larose', '1978-06-05', '2011-10-12', 'Secetaire', 3380.00,  
180.00, 'Limbe', '40515901'),
```

```
(7, 'Louvensky', 'Jean', '1995-07-15', '2018-11-20', 'Chauffeur', 4030.00,  
270.00, 'Cap-Haitien', '35515001'),
```

```
(8, 'Arisno', 'Normil', '1983-08-10', '2009-12-30', 'Chauffeur', 2665.00,  
220.00, 'Cap-Haitien', '35519991'),
```

```
(9, 'Richkard', 'Gue', '1992-09-25', '2013-01-15', 'Chauffeur', 3315.00,  
230.00, 'Verette', '30009991'),
```

(10, 'Blemy', 'Joseph', '1985-10-10', '2024-02-20', 'Directeur', 8315.00, 200.00, 'Cap-Haitien', '30229991');

➤ **Insertion dans la table Destination**

INSERT INTO Destination (DestinationId, Ville, departement, Distance)
VALUES

(1, 'Port-au-Prince', 'Ouest', 150.00),

(2, 'Cap-Haïtien', 'Nord', 210.00),

(3, 'Les Cayes', 'Sud', 400.00),

(4, 'Gonaïves', 'Artibonite', 300.00),

(5, 'Jacmel', 'Sud-Est', 180.00),

(6, 'Hinche', 'Centre', 220.00),

(7, 'Port-de-Paix', 'Nord-Ouest', 270.00),

(8, 'Jeremie', 'Nord-Est', 330.00),

(9, 'Mirebalais', 'Centre', 190.00),

(10, 'Saint-Marc', 'Artibonite', 210.00);

➤ **Insertion dans la table Compagnie**

INSERT INTO Compagnie (Code_CompagnieId, Nom, slogan) VALUES

(1, 'Transporteur', 'Voyagez avec Confort'),

(2, 'LeopardBus', 'Rapide et Sûr'),

(3, 'Flamboyant', 'Votre Voyage, Notre Passion'),

(4, 'JoyTravel', 'Voyage en folie'),

(5, 'VIPBus', 'Voyage de Luxe'),

(6, 'ColombeRide', 'Sécurité et Confort'),

(7, 'ExpressLine', 'Voyage Express'),

(8, 'Sunshine', 'Voyagez avec le Soleil'),
(9, 'ComfortBus', 'Confort avant tout'),
(10, 'TravelFast', 'Voyage Rapide et Confortable');

➤ **Insertion dans la table Bus**

```
INSERT INTO Bus (BusId, Code_CompagnieId, NoImma, Vitesse,  
capacite) VALUES  
(100, 1, 'AB1234', 105.00, 50),  
(102, 2, 'BC2345', 110.00, 45),  
(103, 3, 'CD3456', 130.00, 60),  
(104, 4, 'DE4567', 125.00, 55),  
(105, 5, 'EF5678', 115.00, 50),  
(106, 6, 'FG6789', 120.00, 40),  
(107, 7, 'GH7890', 135.00, 65),  
(108, 8, 'HI8901', 110.00, 45),  
(109, 9, 'IJ9012', 125.00, 55),  
(110, 10, 'JK0123', 130.00, 60);
```

➤ **Insertion dans la table Voyage**

```
INSERT INTO Voyage (VoyageId, BusId, DestinationId, DateDepart,  
HeureDepart, HeureArrive) VALUES  
(1, 101, 1, '2024-07-01', '08:00:00', '12:00:00'),  
(2, 102, 2, '2024-07-02', '09:00:00', '13:00:00'),  
(3, 103, 3, '2024-07-03', '10:00:00', '14:00:00'),  
(4, 104, 4, '2024-07-04', '11:00:00', '15:00:00'),  
(5, 105, 5, '2024-07-05', '12:00:00', '16:00:00'),
```

(6, 106, 6, '2024-07-06', '13:00:00', '17:00:00'),
(7, 107, 7, '2024-07-07', '14:00:00', '18:00:00'),
(8, 108, 8, '2024-07-08', '15:00:00', '19:00:00'),
(9, 109, 9, '2024-07-09', '16:00:00', '20:00:00'),
(10, 110, 10, '2024-07-10', '17:00:00', '21:00:00');

➤ **Insertion dans la table Reservation**

INSERT INTO Reservation (ReservationId, VoyageId, Nom, Prix,
Reduction, datereservation) VALUES

(1, 1, 'Illa Ledix', 100.00, 10.00, '2024-06-01'),
(2, 2, 'Kesline Lebrun', 120.00, 15.00, '2024-06-02'),
(3, 3, 'Marvensia Charles', 150.00, 20.00, '2024-06-03'),
(4, 4, 'Widline Phanord', 130.00, 12.00, '2024-06-04'),
(5, 5, 'Hancy Joseph', 110.00, 5.00, '2024-06-05'),
(6, 6, 'Rodrigue Fatal', 140.00, 18.00, '2024-06-06'),
(7, 7, 'Destine Fritz', 160.00, 22.00, '2024-06-07'),
(8, 8, 'Dachena Ledix', 115.00, 8.00, '2024-06-08'),
(9, 9, 'Anne-ruth Larose', 135.00, 10.00, '2024-06-09'),
(10, 10, 'Banesta Andre', 125.00, 12.00, '2024-06-10');

➤ **Insertion dans la table Equipage**

INSERT INTO Equipage (VoyageId, codePersonnel) VALUES

(1, 1),
(2, 2),
(3, 3),
(4, 4),

(5, 5),
(6, 6),
(7, 7),
(8, 8),
(9, 9),
(10, 10);

12. Donner les couples chauffeurs qui gagnent le même salaire.

- SELECT p1.Prenom, p1.Nom, p2.Prenom, p2.Nom, p1.salaire FROM
Personnel p1, Personnel p2 WHERE p1.salaire = p2.salaire AND
p1.codePersonnel < p2.codePersonnel AND p1.Typepersonnel = 'Chauffeur'
AND p2.Typepersonnel = 'Chauffeur';

```
mysql> SELECT p1.Prenom, p1.Nom, p2.Prenom, p2.Nom, p1.salaire FROM Personnel p1, Personnel p2 WHERE p1.salaire = p2.salaire AND p1.codePersonnel < p2.codePersonnel;  
+-----+-----+-----+-----+-----+  
| Prenom | Nom | Prenom | Nom | salaire |  
+-----+-----+-----+-----+-----+  
| Emmanuel | Jedy | Louvensky | Jean | 4030.00 |  
+-----+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

13. Donner le numéro, le nom et le salaire de chaque chauffeur ayant participé à plus que 10 voyages.

- SELECT Personnel.codePersonnel, Personnel.Prenom, Personnel.Nom,
Personnel.salaire FROM Personnel, Equipage WHERE
Personnel.codePersonnel = Equipage.codePersonnel AND
Personnel.Typepersonnel = 'Chauffeur' GROUP BY
Personnel.codePersonnel, Personnel.Prenom, Personnel.Nom,
Personnel.salaire HAVING COUNT(Equipage.VoyageId) > 10;

```
mysql> SELECT Personnel.codePersonnel, Personnel.Prenom, Personnel.Nom, Personnel.salaire FROM Personnel, Equipage WHERE
Personnel.codePersonnel = Equipage.codePersonnel AND Personnel.Typepersonnel = 'Chauffeur' GROUP BY Personnel.codePerso
nnel, Personnel.Prenom, Personnel.Nom, Personnel.salaire HAVING COUNT(Equipage.VoyageId) > 10;
Empty set (0.00 sec)

mysql> |
```

14. Lister les employés qui ont été embauchés entre janvier à mars 2024.

- Select * from Personnel where dateembauche between '2024-01-01' and '2024-03-31';

```
mysql> Select * from Personnel where dateembauche between '2024-01-01' and '2024-03-31';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| codePersonnel | Prenom | Nom | dateNaissance | dateembauche | Typepersonnel | salaire | commission | adresse | telephone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 10 | Blemy | Joseph | 1985-10-10 | 2024-02-20 | Directeur | 8315.00 | 200.00 | Cap-Haitien | 30229991 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

15. Quels sont les employés habitant Cap-Haitien

- Select * from Personnel where adresse = 'Cap-Haitien';

```
mysql> Select * from Personnel where adresse = 'Cap-Haitien';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| codePersonnel | Prenom | Nom | dateNaissance | dateembauche | Typepersonnel | salaire | commission | adresse | telephone |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Jacques | Jean Baptiste | 1980-01-15 | 2023-05-20 | Chauffeur | 3250.00 | 200.00 | Cap-Haitien | 35359939 |
| 2 | Lesly | JEAN PIERRE | 1975-02-20 | 2023-06-15 | Chauffeur | 3510.00 | 150.00 | Cap-Haitien | 33571738 |
| 7 | Louvensky | Jean | 1995-07-15 | 2018-11-20 | Chauffeur | 4030.00 | 270.00 | Cap-Haitien | 35515001 |
| 8 | Arisno | Normil | 1983-08-10 | 2009-12-30 | Chauffeur | 2665.00 | 220.00 | Cap-Haitien | 35519991 |
| 10 | Blemy | Joseph | 1985-10-10 | 2024-02-20 | Directeur | 8315.00 | 200.00 | Cap-Haitien | 30229991 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> |
```

16. Quel est l'employé le plus ancien ?

- SELECT Prenom, Nom, dateembauche FROM Personnel ORDER BY dateembauche ASC LIMIT 1;

```
mysql> SELECT Prenom, Nom, dateembauche FROM Personnel ORDER BY dateembauche ASC LIMIT 1;
+-----+-----+-----+
| Prenom | Nom | dateembauche |
+-----+-----+-----+
| Dieuleste | Ledix | 2008-09-10 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

17. Lister les noms et les salaires employés mieux payés que « Lesly JEAN PIERRE ».

- `SELECT Prenom, Nom, salaire FROM Personnel WHERE salaire > (SELECT salaire FROM Personnel WHERE Prenom = 'Lesly' AND Nom = 'Jean Pierre');`

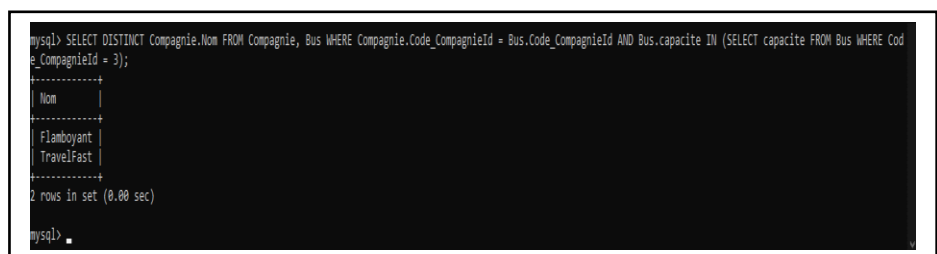


```
mysql> SELECT Prenom, Nom, salaire FROM Personnel WHERE salaire > (SELECT salaire FROM Personnel WHERE Prenom = 'Lesly' AND Nom = 'Jean Pierre');
+-----+-----+-----+
| Prenom | Nom   | salaire |
+-----+-----+-----+
| Emmanuel | Jedy | 4838.00 |
| Malnes   | CoLas | 4160.00 |
| Louvensky | Jean | 4838.00 |
| Bleny    | Joseph | 8315.00 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

18. Donnez le nom de toutes les compagnies possédant au moins un bus de même type que la compagnie d'identifiant 3.

- `SELECT DISTINCT Compagnie.Nom FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId AND Bus.capacite IN (SELECT capacite FROM Bus WHERE Code_CompagnieId = 3);`



```
mysql> SELECT DISTINCT Compagnie.Nom FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId AND Bus.capacite IN (SELECT capacite FROM Bus WHERE Code_CompagnieId = 3);
+-----+
| Nom       |
+-----+
| Flamboyant |
| TravelFast |
+-----+
2 rows in set (0.00 sec)

mysql>
```

19. Donnez pour chaque compagnie la liste des bus qu'elle possède, classée par ordre de capacité croissante.

- `SELECT Compagnie.Nom, Bus.NoImma, Bus.capacite FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId ORDER BY Bus.capacite ASC;`

```
mysql> SELECT Compagnie.Nom, Bus.NoImma, Bus.capacite FROM Compagnie, Bus WHERE Compagnie.Code_CompagnieId = Bus.Code_CompagnieId ORDER BY Bus.capacite ASC;
```

| Nom | NoImma | capacite |
|--------------|--------|----------|
| ColombeRide | FG6789 | 40 |
| LeopardBus | BC2345 | 45 |
| Sunshine | HI8901 | 45 |
| Transporteur | AB1234 | 50 |
| VIPBus | EF5678 | 50 |
| JoyTravel | DE4567 | 55 |
| ComfortBus | IJ9012 | 55 |
| Flamboyant | CD3456 | 60 |
| TravelFast | JK0123 | 60 |
| ExpressLine | GH7890 | 65 |

```
10 rows in set (0.00 sec)

mysql>
```

20. Informations sur les personnels plus âgés que 45 années.

- Select * from Personnel where dateNaissance < '1979-01-01';

```
mysql> Select * from Personnel where dateNaissance < '1979-01-01';
```

| codePersonnel | Prenom | Nom | dateNaissance | dateembauche | Typepersonnel | salaire | commission | adresse | telephone |
|---------------|--------|-------------|---------------|--------------|---------------|---------|------------|-------------|-----------|
| 2 | Lesly | JEAN PIERRE | 1975-02-20 | 2023-06-15 | Chauffeur | 2700.00 | 150.00 | Cap-Haitien | 33571738 |
| 6 | Debora | Larose | 1978-06-05 | 2011-10-12 | Chauffeur | 3000.00 | 180.00 | Limbe | 40515901 |

```
2 rows in set (0.00 sec)

mysql>
```

21. Ecrire la requête qui affiche pour chaque personnel le nom, le salaire, la commission ainsi que le gain annuel.

- SELECT Nom, salaire, commission, (salaire * 12 + commission) AS gain_annuel FROM Personnel;

```
mysql> SELECT Nom, salaire, commission, (salaire * 12 + commission) AS gain_annuel FROM Personnel;
```

| Nom | salaire | commission | gain_annuel |
|---------------|---------|------------|-------------|
| Jean Baptiste | 3250.00 | 200.00 | 39200.00 |
| JEAN PIERRE | 3510.00 | 150.00 | 42270.00 |
| Jeady | 4030.00 | 210.00 | 48570.00 |
| Colas | 4160.00 | 300.00 | 50220.00 |
| Ledix | 1690.00 | 100.00 | 20380.00 |
| Larose | 3380.00 | 180.00 | 40740.00 |
| Jean | 4030.00 | 270.00 | 48630.00 |
| Normil | 2665.00 | 220.00 | 32200.00 |
| Gue | 3315.00 | 230.00 | 40010.00 |
| Joseph | 8315.00 | 200.00 | 99980.00 |

```
10 rows in set (0.00 sec)

mysql>
```

22. Donner la masse salariale du personnel par type personnel.

- SELECT Typepersonnel, SUM(salaire) AS masse_salariale FROM Personnel GROUP BY Typepersonnel;

```
mysql> SELECT Typepersonnel, SUM(salaire) AS masse_salariale FROM Personnel GROUP BY Typepersonnel;
```

| Typepersonnel | masse_salariale |
|---------------|-----------------|
| Chauffeur | 26650.00 |
| Secrétaire | 3380.00 |
| Directeur | 8315.00 |

```
3 rows in set (0.00 sec)

mysql> |
```

23. Ecrire la requête qui affiche le nom, le salaire mensuel des personnels ainsi que le salaire augmenté de : 15% si le salaire est > 20000 25% si le salaire est compris entre 15000 et 20000 45% si le salaire est inférieur à 15000.

➤ SELECT nom, salaire,

CASE

WHEN salaire > 20000 THEN salaire * 1.15

WHEN salaire >= 15000 AND salaire <= 20000 THEN salaire * 1.25

WHEN salaire < 15000 THEN salaire * 1.45

END as salaire_augmente FROM Personnel;

```
mysql> select nom, salaire,
-> CASE
-> WHEN salaire > 20000 THEN salaire * 1.15
-> WHEN salaire >= 15000 AND salaire <= 20000 THEN salaire * 1.25
-> WHEN salaire < 15000 THEN salaire * 1.45
-> END as salaire_augmente FROM Personnel;
```

| nom | salaire | salaire_augmente |
|---------------|---------|------------------|
| Jean Baptiste | 3250.00 | 4712.5000 |
| JEAN PIERRE | 3510.00 | 5089.5000 |
| Jeudy | 4030.00 | 5843.5000 |
| Colas | 4160.00 | 6022.0000 |
| Ledix | 1690.00 | 2450.5000 |
| Larose | 3380.00 | 4901.0000 |
| Jean | 4030.00 | 5843.5000 |
| Normil | 2665.00 | 3864.2500 |
| Gue | 3315.00 | 4806.7500 |
| Joseph | 8315.00 | 12056.7500 |

```
10 rows in set (0.00 sec)
```

24. Donner le nom, âge, salaire mensuel du chauffeur et un champ commentaire qui aura pour alias CMT TRES ELEVE si le salaire est supérieur 20000, ELEVE si le salaire est entre 15000 et 20000, MOYEN si le salaire est inférieur à 15000.

➤ SELECT Nom, 2024 - YEAR(dateNaissance) AS age, salaire,

CASE

WHEN salaire > 20000 THEN 'TRES ELEVE'

WHEN salaire BETWEEN 15000 AND 20000 THEN 'ELEVE'

WHEN salaire < 15000 THEN 'MOYEN'

END AS CMT

FROM Personnel

WHERE Typepersonnel = 'Chauffeur';

```
mysql> SELECT Nom, 2024 - YEAR(dateNaissance) AS age, salaire,
-> CASE
-> WHEN salaire > 20000 THEN 'TRES ELEVE'
-> WHEN salaire BETWEEN 15000 AND 20000 THEN 'ELEVE'
-> WHEN salaire < 15000 THEN 'MOYEN'
-> END AS CMT
-> FROM Personnel
-> WHERE Typepersonnel = 'Chauffeur';
+-----+-----+-----+-----+
| Nom      | age | salaire | CMT  |
+-----+-----+-----+-----+
| Jean Baptiste | 44 | 3250.00 | MOYEN |
| JEAN PIERRE   | 49 | 3510.00 | MOYEN |
| Jeudy        | 36 | 4030.00 | MOYEN |
| Colas       | 34 | 4160.00 | MOYEN |
| Ledix       | 42 | 1690.00 | MOYEN |
| Jean        | 29 | 4030.00 | MOYEN |
| Normil      | 41 | 2665.00 | MOYEN |
| Gue        | 32 | 3315.00 | MOYEN |
+-----+-----+-----+-----+
8 rows in set (0.02 sec)

mysql> |
```

25. Modification et Suppression de données Ecrire les requêtes SQL permettant

- D'augmenter de 30% le salaire des chauffeurs.

- UPDATE Personnel SET salaire = salaire * 1.30 WHERE Typepersonnel = 'Chauffeur';

```
mysql> UPDATE Personnel SET salaire = salaire * 1.30 WHERE Typepersonnel = 'Chauffeur';
Query OK, 8 rows affected (0.03 sec)
Rows matched: 8 Changed: 8 Warnings: 0

mysql> |
```

- D'augmenter de 1500 gdes le salaire du chauffeur le moins payé.

- UPDATE Personnel SET salaire = salaire + 1500 WHERE Typepersonnel = 'Chauffeur' ORDER BY salaire ASC LIMIT 1 ;

```
mysql> UPDATE Personnel SET salaire = salaire + 1500 WHERE Typepersonnel = 'Chauffeur' ORDER BY salaire ASC LIMIT 1 ;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> |
```

- De diminuer de 750 gdes le salaire du chauffeur le plus payé.

- UPDATE Personnel SET salaire = salaire - 750 WHERE Typepersonnel = 'Chauffeur' ORDER BY salaire DESC LIMIT 1 ;

```
mysql> UPDATE Personnel SET salaire = salaire - 750 WHERE Typepersonnel = 'Chauffeur' ORDER BY salaire DESC LIMIT 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql>
```

• **De supprimer les chauffeurs plus âgés que 65 ans.**

- DELETE FROM Personnel WHERE Typepersonnel = 'Chauffeur' AND 2024 - YEAR(dateNaissance) > 65;

26. Respectant le format suivant, donner la requête vous permettant d'afficher :

• **Le nom du personnel et son revenue annuel (qu'il reçoit de commission ou pas)**

| <i>Personnel</i> | <i>a un gain annuel</i> | <i>sur 12 mois</i> |
|------------------|-------------------------|--------------------|
| ----- | ----- | ----- |
| <Nom et prénom> | gagne <Salaire> | par an. |

- select CONCAT(nom, ' ', prenom) as Personnel, CONCAT(' gagne ', salaire * 12) as 'a un gain annuel ', CONCAT(' par an ') as 'sur 12 mois ' from personnel;

```
mysql> select CONCAT(nom, ' ', prenom) as Personnel, CONCAT(' gagne ', salaire * 12) as 'a un gain annuel ', CONCAT(' par an ') as 'sur 12 mois ' from personnel;
+-----+-----+-----+
| Personnel | a un gain annuel | sur 12 mois |
+-----+-----+-----+
| Jean Baptiste Jacques | gagne 56700.00 | par an |
| JEAN PIERRE Lesly | gagne 54756.00 | par an |
| Jedy Emmanuel | gagne 62868.00 | par an |
| Colas Walnes | gagne 55896.00 | par an |
| Lédix Dieuleste | gagne 44364.00 | par an |
| Larose Debora | gagne 48560.00 | par an |
| Jean Louvensky | gagne 62868.00 | par an |
| Normil Arisno | gagne 41574.00 | par an |
| Gue Rickard | gagne 51714.00 | par an |
| Joseph Bleny | gagne 99788.00 | par an |
+-----+-----+-----+
10 rows in set, 2 warnings (0.01 sec)

mysql>
```