**ANGUILE-DIOP William**

**BTS SIO**

**PPE 3:Mission 3**

**MCD**



**MLD**



**Traduire le MLD Graphique en un représentation textuelle simplifiée d’une base de Données :**

· Véhicule (**id\_Vehicule**, immatriculation, age, etat, #id\_Agence, #id\_Marque, #id\_Type, #id\_categorie, #id\_modele)

· Marque (**id\_marque**, nom)

· Type (**id\_type**, libelle)

· Catégorie (**id\_categorie**, libelle)

· Modèle (**id\_modele**, denomination, puissance)

· Agence (**id\_Agence**, nom, nb\_employes, #id\_pays)

· Pays (**id\_pays**, nom, nombre\_habitant, superficie)

· a\_loue (**id\_client**, **id\_vehicule**, date\_de\_retrait, date\_de\_retour)

· Client (**id\_client**, nom, adresse, code\_postal, ville)

**Réalisez les requêtes suivantes :**

* Afficher toutes les informations sur les véhicules loués par le Client n°T122

**SELECT** v.\*  
**FROM** `vehicule` v  
**INNER** **JOIN** a\_loue al **ON** v.id\_vehicule = al.id\_vehicule  
**WHERE** al.id\_client = "T122"

* Afficher toutes les locations réalisées par le client n° T122

**SELECT** v.Immatriculation  
**FROM** `vehicule` v  
**INNER** **JOIN** a\_loue al **ON** v.id\_vehicule = al.id\_vehicule  
**WHERE** al.id\_client = "T122"

* Afficher l’immatriculation, l’âge et l’état de tous les véhicules.

**SELECT** Immatriculation, Age, Etat  
**FROM** `vehicule`

* Afficher les noms des clients et les adresses, des clients qui habitent à << Nice >>.

**SELECT** `Nom` , `Adresse`  
**FROM** `client`  
**WHERE** `Ville` = "Nice"

* Affiche la liste des clients par ordre alphabétique croissant des noms

**SELECT** \*  
**FROM** client  
**ORDER** **BY** nom **ASC**

* Ajouter l’attribut kilométrage et Afficher la liste des voitures par ordre décroissant des compteurs (kilométrage)

**SELECT** \*  
**FROM** vehicule  
**ORDER** **BY** `kilometrage` **DESC\***

* Afficher les informations sur les clients qui ont loué la voiture EW 25EW

**SELECT** c . \*  
**FROM** `client` c  
**INNER** **JOIN** a\_loue al **ON** c.id\_Client = al.id\_client  
**INNER** **JOIN** vehicule v **ON** al.id\_vehicule = v.id\_vehicule  
**WHERE** v.immatriculation = "EW25EW"

* Afficher toutes les voitures noires :)

**SELECT** \*  
**FROM** `vehicule`  
**WHERE** `couleur` = 'noir'

* Afficher toutes les voitures ayant un kilométrage <10000 km

**SELECT** \*  
**FROM** `vehicule`  
**WHERE** `kilometrage` < 10000

* Afficher toutes les informations sur les locations réalisées avant 2018

**SELECT** \*  
**FROM** a\_loue  
**WHERE** `Date\_de\_retrait` < '2018-01-01';

* Afficher la moyenne des kilométrages de tous les véhicules du parc.

**SELECT** AVG( `kilometrage` )  
**FROM** vehicule

L. Afficher toutes les locations réalisées en 2018

**SELECT** \*  
**FROM** a\_loue  
**WHERE** `Date\_de\_retrait`  
**BETWEEN** '2018-01-01'  
**AND** '2018-12-31';

M. Afficher le nombre de voitures ayant un kilométrage <10 000 kilomètres

**SELECT** COUNT( \* )  
**FROM** `vehicule`  
**WHERE** `kilometrage` <10000

Partie 2

• Obtenir la liste des véhicules empruntés et rendu le même jour ainsi que l’agence de rattachement

**SELECT** v.id\_vehicule, a.nom, al.date\_de\_retrait, al.date\_de\_retour  
**FROM** vehicule v  
**INNER** **JOIN** a\_loue al **ON** v.id\_vehicule = al.id\_vehicule  
**INNER** **JOIN** agence a **ON** a.id\_agence = v.id\_agence  
**WHERE** al.date\_de\_retrait = al.date\_de\_retour

• Obtenir le nombre véhicules pour chaque marque

**SELECT** m.nom, COUNT( \* )  
**FROM** vehicule v  
**INNER** **JOIN** marque m **ON** v.id\_marque = m.id\_marque  
**GROUP** **BY** m.nom

• Obtenir les noms des clients qui ont loué plus de 10 véhicules de marque « Renault »

**SELECT** c.nom, COUNT( \* )  
**FROM** vehicule v  
**INNER** **JOIN** marque m **ON** v.id\_marque = m.id\_marque  
**INNER** **JOIN** a\_loue al **ON** v.id\_vehicule = al.id\_vehicule  
**INNER** **JOIN** client c **ON** al.id\_client = c.id\_client  
**WHERE** m.nom = "Renault"  
**GROUP** **BY** m.nom  
**HAVING** COUNT( \* ) >10

• Obtenir le nombre d’agences et d’employés par pays.

**SELECT** p.nom, Nb\_employes, COUNT( a.id\_agence ) **AS** "Nb agence"  
**FROM** agence a  
**INNER** **JOIN** pays p **ON** a.id\_pays = p.id\_pays  
**GROUP** **BY** p.nom

**Exercice 2 :**

Ecrire les requêtes SQL permettant d’afficher :

* **Les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l’ordre alphabétique croisant du nom**

**SELECT** CodeEt, NomEt, DatnEt  
**FROM** ETUDIANT  
**ORDER** **BY** NomEt **ASC** ;

* **Les noms et les grades des enseignants de la matière dont le nom est ‘BD’.**

**SELECT** NomEns, GradeEns  
**FROM** ENSEIGNANT  
**WHERE** CodeMat = (  
**SELECT** CodeMat  
**FROM** MATIERE  
**WHERE** NomMat = 'BD' );

* **La liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade ‘Grd3’.**

**SELECT** **DISTINCT** NomMat, CoefMat  
**FROM** MATIERE  
**WHERE** CodeMat  
**IN** (

**SELECT** CodeMat  
**FROM** ENSEIGNANT  
**WHERE** GradeEns = 'Grd3'

);

* La liste des matières (Nom et Coefficient) qui sont suivies par l’étudiant de code ‘Et321’.

**SELECT** NomMat, CoefMat  
**FROM** MATIERE  
**WHERE** CodeMat  
**IN** (

**SELECT** CodeMat  
**FROM** NOTE  
**WHERE** CodeEt = 'Et321'

);

* Le nombre d’enseignants de la matière dont le nom est ‘Informatique’

**SELECT** COUNT( \* )  
**FROM** ENSEIGNANT  
**WHERE** CodeMat = (  
**SELECT** CodeMat  
**FROM** MATIERE  
**WHERE** NomMat = 'Informatique' );

**Exercice 3 :**

**Exprimez en SQL les requêtes suivantes :**

* **Quelle est la composition de l’équipe Festina (Numéro, nom et pays des coureurs) ?**

**SELECT** C.NumeroCoureur, C.NomCoureur, PAYS.NomPays  
**FROM** COUREUR C  
**INNER** **JOIN** EQUIPE E **ON** C.CodeEquipe = E.CodeEquipe  
**INNER** **JOIN** PAYS **ON** C.CodePays = PAYS.CodePays  
**WHERE** E.NomEquipe = 'Festina'

* le nombre de kilomètres total du Tour de France 97 ?

**SELECT** SUM( NbKm ) **AS** "Nombre kilometre total"  
**FROM** etape

* le nombre de kilomètres total des étapes de type "Haute Montagne"

**SELECT** SUM( e.NbKm ) **AS** "Nombre de kilomètres total pour le type Haute Montagne"  
**FROM** etape e  
**INNER** **JOIN** type\_etape t **ON** e.CodeType = t.CodeType  
**WHERE** t.LibelleType = 'Haute Montagne'

* les noms des coureurs qui n’ont pas obtenu de bonifications

**SELECT** c.NomCoureur  
**FROM** coureur c  
**WHERE** c.NumeroCoureur **NOT**  
**IN** (

**SELECT** a.NumeroCoureur  
**FROM** ATTRIBUER\_BONIFICATION a

)

* les noms des coureurs qui ont participé à toutes les étapes ?

**SELECT** c.NomCoureur  
**FROM** coureur c  
**WHERE** **NOT**  
**EXISTS** (

**SELECT** e.NumeroEtap  
**FROM** etape e  
**WHERE** e.NumeroEtap **NOT**  
**IN** (**SELECT** p.NumeroEtap

**FROM** participer p  
 **WHERE** p.NumeroCoureur = c.NumeroCoureur)

)

* le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l’issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?

**SELECT** c.NomCoureur, c.CodeEquipe, c.CodePays, SUM( p.TempsRealise + ab.NbSecondes ) **AS** TotalTemps  
**FROM** coureur c  
**INNER JOIN** participer p **ON** c.NumeroCoureur = p.NumeroCoureur  
**INNER JOIN** ATTRIBUER\_BONIFICATION ab **ON** p.NumeroCoureur = ab.NumeroCoureur  
**AND** p.NumeroEtap = ab.NumeroEtap  
**INNER JOIN** etape e **ON** e.NumeroEtap = p.NumeroEtap  
**WHERE** e.NumeroEtap <=13  
**GROUP** **BY** c.NumeroCoureur  
**ORDER** **BY** TotalTemps

* Quel est le classement par équipe à l’issue des 13 premières étapes (nom et temps des équipes) ?

**SELECT** NomEquipe, SUM( TempsRealise + NbSecondes ) **AS** TempsTotal  
**FROM** equipe e  
**INNER JOIN** coureur c **ON** c.CodeEquipe = e.CodeEquipe  
**INNER JOIN** participer p **ON** p.NumeroCoureur = c.NumeroCoureur  
**LEFT** **JOIN** ATTRIBUER\_BONIFICATION ab **ON** ab.NumeroCoureur = c.NumeroCoureur  
**GROUP** **BY** NomEquipe  
**ORDER** **BY** TempsTotal

**Exercice 4 :**

LES COMMANDES:

CREATE TABLE Client (  
Numcli INT PRIMARY KEY,  
Nomcli VARCHAR(255) NOT NULL,  
Prenomcli VARCHAR(255) NOT NULL,  
adressecli VARCHAR(255) NOT NULL,  
mailcli VARCHAR(255)  
);  
  
  
CREATE TABLE Produit (  
Numprod INT PRIMARY KEY,  
designation VARCHAR(255) NOT NULL,  
prix DECIMAL(10, 2) NOT NULL,  
qte\_stock INT DEFAULT 0  
);

**// Le premier qui m’envoie une tête de pierre en message privée discord, je lui donne ce qu’il veut =) c kdo**

CREATE TABLE Vendeur (  
Idvendeur INT PRIMARY KEY,  
Nomvendeur VARCHAR(255) NOT NULL,  
adresse\_vend VARCHAR(255) NOT NULL  
);  
  
  
CREATE TABLE Commande (  
Numcom INT PRIMARY KEY,  
Numcli INT NOT NULL,  
Idvendeur INT NOT NULL,  
Numprod INT NOT NULL,  
date\_com DATE NOT NULL,  
qte\_com INT NOT NULL,  
FOREIGN KEY (Numcli) REFERENCES Client(Numcli),  
FOREIGN KEY (Idvendeur) REFERENCES Vendeur(Idvendeur),  
FOREIGN KEY (Numprod) REFERENCES Produit(Numprod)  
);

* **la liste des clients de Marrakech.**

**SELECT** Nomcli, Prenomcli, adressecli  
**FROM** Client  
**WHERE** adressecli = 'Marrakech'

* **la liste des produits (Numprod, désignation, prix) classés de plus cher au moins cher.**

**SELECT** Numprod, designation, prix  
**FROM** Produit  
**ORDER** **BY** prix **DESC**

* **noms et adresses des vendeurs dont le nom commence par la lettre ‘M’.**

**SELECT** Nomvendeur, adresse\_vend  
**FROM** Vendeur  
**WHERE** Nomvendeur **LIKE** 'M%'

* **la liste des commandes effectuées par le vendeur "Mohammed" entre le 1er et 30 janvier 2020.**

**SELECT** c.\*  
**FROM** Commande c  
**INNER** **JOIN** Vendeur v **ON** c.Idvendeur = v.Idvendeur  
**WHERE** Nomvendeur = 'Mohammed'  
**AND** date\_com  
**BETWEEN** '2020-01-01'  
**AND** '2020-01-30'

**5. le nombre des commandes contenant le produit n° 365.**

**SELECT** COUNT( \* ) **AS** "Nombre de commande pour le produit n°365"  
**FROM** Commande  
**WHERE** Numprod =365

**Exercice 5 :**

**// Le premier qui m’envoie un drapeau albanais en message privé discord, je lui donne ce qu’il veut =) c kdo (non-compatible si vous avez déjà envoyé une tête de pierre)**

les commandes SQL permettant de rechercher :

* La liste de tous les étudiants.

**SELECT** nom, prenom  
**FROM** etudiant

* Nom et coefficient des matières.

**SELECT** nom\_matiere, coefficient  
**FROM** matiere

* Les numéros des cartes d’identité des étudiants dont la moyenne entre 7 et 12.

**SELECT** numero\_carte\_etudiant  
**FROM** ETUDIANT  
**WHERE** numero\_carte\_etudiant  
**IN** (

**SELECT** numero\_carte\_etudiant  
**FROM** NOTE  
**GROUP** **BY** numero\_carte\_etudiant  
**HAVING** AVG( note\_examen )  
**BETWEEN** 7  
**AND** 12

)

* La liste des étudiants dont le nom commence par ‘ben’.

**SELECT** nom, prenom, numero\_carte\_etudiant  
**FROM** ETUDIANT  
**WHERE** nom **LIKE** 'ben%'

* Le nombre des étudiants qui ont comme matière ‘12518’.

**SELECT** COUNT( \* ) **AS** "nombre des étudiants qui ont comme matière 12518"  
**FROM** (

**SELECT** **DISTINCT** numero\_carte\_etudiant  
**FROM** NOTE  
**WHERE** code\_matiere = '12518'

) **AS** students

* La somme des coefficients des matières.

**SELECT** SUM( coefficient ) **AS** "total coefficient"  
**FROM** MATIERE

* Les noms des étudiants qui une note\_examen >10.

**SELECT** nom, prenom  
**FROM** etudiant e  
**INNER JOIN** note n **ON** e.numero\_carte\_etudiant = n.numero\_carte\_etudiant  
**WHERE** note\_examen >10

8- Afficher les noms et les coefficients des matières étudier par l’étudiant "01234568".

**SELECT** nom\_matiere, coefficient  
**FROM** matiere m  
**INNER** **JOIN** note n **ON** n.code\_matiere = n.code\_matiere  
**INNER** **JOIN** etudiant e **ON** n.numero\_carte\_etudiant = e.numero\_carte\_etudiant  
**WHERE** e.numero\_carte\_etudiant = '01234568'