

EXO 1 :

Soit la relation Pays suivante :

Pays (**CodePays**, NomPays, Continent, Population)

Code Pays	Nom Pays	Continent	Population
AF01	Égypte	Afrique	99416400
EU01	Russie	Europe	146267288
EU02	Allemagne	Europe	81174000
AM01	Canada	Amérique	35056064
AF02	Algérie	Afrique	43378027
AM02	USA	Amérique	317627401
AF03	Maroc	Afrique	35689900
EU03	France	Europe	66352469
EU04	Royaume-Uni	Europe	64767115
EU05	Italie	Europe	60795612
EU06	Espagne	Europe	46439864
AM03	Mexique	Amérique	118395054
AF04	Libye	Afrique	6777452
AF05	Mauritanie	Afrique	4077347
AM04	Cuba	Amérique	11184023
AM05	Argentine	Amérique	40677348
AM06	Brésil	Amérique	201032714
AS01	Chine	Asie	1 394960000
AS02	Inde	Asie	1 345060000
AS03	Pakistan	Asie	204094000
AF06	Tunisie	Afrique	11551448
AF07	Soudan	Afrique	41937285
AF08	Mali	Afrique	19973000
AF09	Soudan du Sud	Afrique	12778250
AF10	Somalie	Afrique	15442905
EU07	Belgique	Europe	11258434
EU08	Portugal	Europe	10374822
AS04	Japon	Asie	126220000
AS05	Iran	Asie	82308000
AF11	Tchad	Afrique	15692969
EU09	Suisse	Europe	8236573
EU10	Autriche	Europe	8584926
AU01	Australie	Australie	25200000
AS06	Turquie	Asie	82003882

```
CREATE TABLE Pays (
    CodePays VARCHAR(15) NOT NULL,
    NomPays VARCHAR(30) NOT NULL,
    Continent VARCHAR(30) NOT NULL,
    Population INT NOT NULL,
```

PRIMARY KEY (CodePays)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;

INSERT INTO Pays (CodePays, NomPays, Continent, Population)
VALUES

('AF01', 'Égypte', 'Afrique', 99416400),
('EU01', 'Russie', 'Europe', 146267288),
('EU02', 'Allemagne', 'Europe', 81174000),
('AM01', 'Canada', 'Amérique', 35056064),
('AF02', 'Algérie', 'Afrique', 43378027),
('AM02', 'USA', 'Amérique', 317627401),
('AF03', 'Maroc', 'Afrique', 35689900),
('EU03', 'France', 'Europe', 66352469),
('EU04', 'Royaume-Uni', 'Europe', 64767115),
('EU05', 'Italie', 'Europe', 60795612),
('EU06', 'Espagne', 'Europe', 46439864),
('AM03', 'Mexique', 'Amérique', 118395054),
('AF04', 'Libye', 'Afrique', 6777452),
('AF05', 'Mauritanie', 'Afrique', 4077347),
('AM04', 'Cuba', 'Amérique', 11184023),
('AM05', 'Argentine', 'Amérique', 40677348),
('AM06', 'Brésil', 'Amérique', 201032714),
('AS01', 'Chine', 'Asie', 1394960000),
('AS02', 'Inde', 'Asie', 1345060000),
('AS03', 'Pakistan', 'Asie', 204094000),
('AF06', 'Tunisie', 'Afrique', 11551448),
('AF07', 'Soudan', 'Afrique', 41937285),
('AF08', 'Mali', 'Afrique', 19973000),
('AF09', 'Soudan du Sud', 'Afrique', 12778250),
('AF10', 'Somalie', 'Afrique', 15442905),
('EU07', 'Belgique', 'Europe', 11258434),
('EU08', 'Portugal', 'Europe', 10374822),
('AS04', 'Japon', 'Asie', 126220000),
('AS05', 'Iran', 'Asie', 82308000),
('AF11', 'Tchad', 'Afrique', 15692969),
('EU09', 'Suisse', 'Europe', 8236573),
('EU10', 'Autriche', 'Europe', 8584926),
('AU01', 'Australie', 'Australie', 25200000),
('AS06', 'Turquie', 'Asie', 82003882) ;

1/- Donner le nombre total de population par continent.
On désire connaître le nombre total de population par continent.

La requête :

SELECT SUM(Population)

FROM Pays;

Permet de calculer le nombre de population pour tous les continents.

Pour calculer le nombre de population par continent, il faut regrouper les enregistrements par continent (faire des groupes de pays appartenant au même continent) et calculer la somme de la population pour chaque groupe (sommer la population des lignes appartenant au même groupe donc appartenant au même continent). Dans cette requête, la fonction à appliquer sur chaque groupe est la fonction **SUM** (d'autres fonctions peuvent être appliquées aux groupes de données pour d'autres usages, tel que la fonction **COUNT** qui permet de compter les lignes de chaque groupe, la fonction **AVG** qui permet de calculer la moyenne de chaque groupe pour la colonne donnée en paramètre à la fonction, les deux fonctions **MIN** et **MAX** qui permettent respectivement d'obtenir le **MIN** et le **MAX** de chaque groupe de données pour la colonne spécifiée pour ces deux fonctions...etc.).





La requête devient ainsi :

```
SELECT Continent, SUM(Population)
FROM Pays
GROUP BY Continent;
```

+ Options				Continent	SUM(Population)
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Afrique	306714983
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Amérique	723972604
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Asie	3234645882
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Australie	25200000
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Europe	504251103

Si on veut attribuer une un nom à la colonne de la somme de la population :

```
SELECT Continent, SUM(Population) AS TotalPopulationContinent
FROM Pays
GROUP BY Continent;
```

				Continent	TotalPopulationContinent
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Afrique	306714983
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Amérique	723972604
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Asie	3234645882
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Australie	25200000
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Europe	504251103

Si on veut ordonner les continents par ordre alphabétique décroissant (ordre alphabétique inversé ; Sachant qu'il existe deux ordres de tri : **ASC** qui est l'ordre croissant par défaut et **DESC** qui est l'ordre décroissant ou l'ordre inversé) :

```
SELECT Continent, SUM(Population) AS TotalPopulationContinent
FROM Pays
GROUP BY Continent
ORDER BY Continent DESC;
```

	Continent	TotalPopulationContinent
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Europe	504251103
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Australie	25200000
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Asie	3234645882
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Amérique	723972604
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Afrique	306714983

2/- Donner pour chaque continent, le pays qui a la plus grande population.

Dans cette requête nous allons regrouper les lignes par continent, pour chaque groupe de lignes appartenant au même continent nous allons afficher le pays qui comporte la plus grande population (**MAX(Population)**).

```
SELECT NomPays, Continent, MAX(Population)
FROM Pays
GROUP BY Continent;
```

	NomPays	Continent	Population
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Égypte	Afrique	99416400
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Canada	Amérique	35056064
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Chine	Asie	1394960000
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Australie	Australie	25200000
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Russie	Europe	146267288






Si nous voulons afficher le résultat du pays le plus peuplé au pays le moins peuplé :

```
SELECT NomPays, Continent, MAX(Population)
FROM Pays
GROUP BY Continent
ORDER BY MAX(Population) ASC;
```

	NomPays	Continent	MAX(Population)
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Australie	Australie	25200000
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Égypte	Afrique	99416400
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Russie	Europe	146267288
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Canada	Amérique	317627401
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Chine	Asie	1394960000

Si nous voulons afficher le résultat du pays le plus peuplé au pays le moins peuplé :

```
SELECT NomPays, Continent, MAX(Population) AS PopulationMaximale
FROM Pays
GROUP BY Continent
ORDER BY MAX(Population) DESC;
```

			NomPays	Continent	MAX(Population) ▾ 1	
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Chine	Asie	1394960000
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Canada	Amérique	317627401
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Russie	Europe	146267288
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Égypte	Afrique	99416400
<input type="checkbox"/>	 Éditer	 Copier	 Supprimer	Australie	Australie	25200000

3/- Donner la moyenne de la population pour chaque continent ?

```
SELECT Continent, AVG(Population) AS MoyenPopulationContinent
FROM Pays
GROUP BY Continent;
```

				Continent	MoyenPopulationContinent
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Afrique	27883180.2727
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Amérique	120662100.6667
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Asie	539107647.0000
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Australie	25200000.0000
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Europe	50425110.3000

Si on veut afficher les continents par ordre croissant de la population moyenne:

```
SELECT Continent, AVG(Population) AS MoyenPopulationContinent
FROM Pays
GROUP BY Continent
ORDER BY AVG(Population) ASC;
```

+ Options					
				Continent	MoyenPopulationContinent
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Australie	25200000.0000
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Afrique	27883180.2727
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Europe	50425110.3000
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Amérique	120662100.6667
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Asie	539107647.0000

Si on veut afficher les continents par ordre décroissant de la population moyenne:

```
SELECT Continent, AVG(Population) AS MoyenPopulationContinent
FROM Pays
GROUP BY Continent
ORDER BY AVG(Population) DESC;
```

+ Options					
				Continent	MoyenPopulationContinent
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Asie	539107647.0000
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Amérique	120662100.6667
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Europe	50425110.3000
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Afrique	27883180.2727
<input type="checkbox"/>	✎ Éditer	📄 Copier	Supprimer	Australie	25200000.0000

4/- Donner le nombre totale de population pour les continent qui sont peuplé par plus de 40 000 000 (40 millions) d'habitant.

Pour faire cette requête, nous allons imposer une condition pour chaque groupe de lignes appartenant au même continent par la commande **HAVING** (le **WHERE** permet **d'imposer une condition sur chaque ligne**, le **HAVING** permet d'imposer **une condition sur chaque groupe de lignes**). On obtient ainsi :

```
SELECT Continent, SUM(Population) AS TotalPopulationContinent
FROM Pays
GROUP BY Continent
HAVING SUM(Population) > 40000000;
```

+ Options		Continent	TotalPopulationContinent
<input type="checkbox"/>	Éditer Copier Supprimer	Afrique	306714983
<input type="checkbox"/>	Éditer Copier Supprimer	Amérique	723972604
<input type="checkbox"/>	Éditer Copier Supprimer	Asie	3234645882
<input type="checkbox"/>	Éditer Copier Supprimer	Europe	504251103

EXO 2 :

Soit la relation suivante de ventes de voitures aux enchères:

Voiture (**Matricule**, Marque, Modèle, AnnéeMiseCirculation, Kilométrage, PrixVente)

Matricule	Marque	Modèle	Année de Mise en Circulation	Kilométrage	Prix de Vente
Mat1	Renault	Clio	2017	1000	11111
Mat2	TOYOTA	YARIS	2017	1000	11111
Mat3	HONDA	CIVIC	2017	1000	11111
Mat4	Hyundai	ACCENT	2017	1000	11111
Mat5	VOLVO	V60	2017	1000	11111
Mat6	FIAT	FIAT 500	2017	1000	11111
Mat7	Citroen	C4	2017	1000	11111
Mat8	Volkswagen	Polo	2017	1000	11111
Mat9	ŠKODA	Octavia	2017	1000	11111
Mat10	Renault	Megane	2018	2000	22222
Mat11	TOYOTA	Corolla	2018	2000	22222
Mat12	HONDA	Accord	2018	2000	22222
Mat13	Hyundai	SONATA	2018	2000	22222
Mat14	VOLVO	V40	2018	2000	22222
Mat15	FIAT	FIAT Punto	2018	2000	22222
Mat16	Citroen	C5	2018	2000	22222
Mat17	Volkswagen	Golf	2018	2000	22222
Mat18	ŠKODA	Fabia	2018	2000	22222

Mat19	Renault	Twingo	2019	3000	33333
Mat20	TOYOTA	PRIUS	2019	3000	33333
Mat21	HONDA	City	2019	3000	33333
Mat22	Hyundai	I30	2019	3000	33333
Mat23	VOLVO	S90	2019	3000	33333
Mat24	FIAT	FIAT Tipo	2019	3000	33333
Mat25	Citroen	Berlingo	2019	3000	33333
Mat26	Volkswagen	Tigwan	2019	3000	33333
Mat27	ŠKODA	Yeti	2019	3000	33333
Mat28	Renault	KANGOO	2020	4000	44444
Mat29	TOYOTA	RAV 4	2020	4000	44444
Mat30	HONDA	CRV	2020	4000	44444
Mat31	Hyundai	I10	2020	4000	44444
Mat32	VOLVO	V70	2020	4000	44444
Mat33	FIAT	FIAT 600	2020	4000	44444
Mat34	Citroen	C1	2020	4000	44444
Mat35	Volkswagen	CADDY	2020	4000	44444
Mat36	ŠKODA	RAPID	2020	4000	44444
Mat37	Renault	Captur	2021	5000	55555
Mat38	TOYOTA	AVENSIS	2021	5000	55555
Mat39	HONDA	Honda E	2021	5000	55555
Mat40	Hyundai	I40	2021	5000	55555
Mat41	VOLVO	CX90	2021	5000	55555
Mat42	FIAT	FIAT 700	2021	5000	55555
Mat43	Citroen	Ds5	2021	5000	55555
Mat44	Volkswagen	Jetta	2021	5000	55555
Mat45	ŠKODA	SUPERB	2021	5000	55555

```
CREATE TABLE Voiture (
    Matricule VARCHAR(15) NOT NULL,
    Marque VARCHAR(30) NOT NULL,
    Modèle VARCHAR(30) NOT NULL,
    AnnéeMiseCirculation YEAR(4) NOT NULL,
    Kilométrage INT NOT NULL,
    PrixVente INT NOT NULL,
    PRIMARY KEY (Matricule)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
```

```
INSERT INTO Voiture (Matricule, Marque, Modèle, AnnéeMiseCirculation, Kilométrage,
PrixVente)
```

```
VALUES
```

```
('Mat1', 'Renault', 'Clio', '2017', 1000, 11111),
('Mat2', 'TOYOTA', 'YARIS', '2017', 1000, 11111),
('Mat3', 'HONDA', 'CIVIC', '2017', 1000, 11111),
('Mat4', 'Hyundai', 'ACCENT', '2017', 1000, 11111),
('Mat5', 'VOLVO', 'V60', '2017', 1000, 11111),
('Mat6', 'FIAT', 'FIAT 500', '2017', 1000, 11111),
('Mat7', 'Citroen', 'C4', '2017', 1000, 11111),
```

('Mat8', 'Volkswagen', 'Polo', '2017', 1000, 11111),
 ('Mat9', 'ŠKODA', 'Octavia', '2017', 1000, 11111),
 ('Mat10', 'Renault', 'Megane', '2018', 2000, 22222),
 ('Mat11', 'TOYOTA', 'Corolla', '2018', 2000, 22222),
 ('Mat12', 'HONDA', 'Accord', '2018', 2000, 22222),
 ('Mat13', 'Hyundai', 'SONATA', '2018', 2000, 22222),
 ('Mat14', 'VOLVO', 'V40', '2018', 2000, 22222),
 ('Mat15', 'FIAT', 'FIAT Punto', '2018', 2000, 22222),
 ('Mat16', 'Citroen', 'C5', '2018', 2000, 22222),
 ('Mat17', 'Volkswagen', 'Golf', '2018', 2000, 22222),
 ('Mat18', 'ŠKODA', 'Fabia', '2018', 2000, 22222),
 ('Mat19', 'Renault', 'Twingo', '2019', 3000, 33333),
 ('Mat20', 'TOYOTA', 'PRIUS', '2019', 3000, 33333),
 ('Mat21', 'HONDA', 'City', '2019', 3000, 33333),
 ('Mat22', 'Hyundai', 'I30', '2019', 3000, 33333),
 ('Mat23', 'VOLVO', 'S90', '2019', 3000, 33333),
 ('Mat24', 'FIAT', 'FIAT Tipo', '2019', 3000, 33333),
 ('Mat25', 'Citroen', 'Berlingo', '2019', 3000, 33333),
 ('Mat26', 'Volkswagen', 'Tiguan', '2019', 3000, 33333),
 ('Mat27', 'ŠKODA', 'Yeti', '2019', 3000, 33333),
 ('Mat28', 'Renault', 'KANGOO', '2020', 4000, 44444),
 ('Mat29', 'TOYOTA', 'RAV 4', '2020', 4000, 44444),
 ('Mat30', 'HONDA', 'CRV', '2020', 4000, 44444),
 ('Mat31', 'Hyundai', 'I10', '2020', 4000, 44444),
 ('Mat32', 'VOLVO', 'V70', '2020', 4000, 44444),
 ('Mat33', 'FIAT', 'FIAT 600', '2020', 4000, 44444),
 ('Mat34', 'Citroen', 'C1', '2020', 4000, 44444),
 ('Mat35', 'Volkswagen', 'CADDY', '2020', 4000, 44444),
 ('Mat36', 'ŠKODA', 'RAPID', '2020', 4000, 44444),
 ('Mat37', 'Renault', 'Captur', '2021', 5000, 55555),
 ('Mat38', 'TOYOTA', 'AVENSIS', '2021', 5000, 55555),
 ('Mat39', 'HONDA', 'Honda E', '2021', 5000, 55555),
 ('Mat40', 'Hyundai', 'I40', '2021', 5000, 55555),
 ('Mat41', 'VOLVO', 'CX90', '2021', 5000, 55555),
 ('Mat42', 'FIAT', 'FIAT 700', '2021', 5000, 55555),
 ('Mat43', 'Citroen', 'Ds5', '2021', 5000, 55555),
 ('Mat44', 'Volkswagen', 'Jetta', '2021', 5000, 55555),
 ('Mat45', 'ŠKODA', 'SUPERB', '2021', 5000, 55555) ;

1/-Donnez le nombre total de voitures vendus aux enchères.

```
SELECT COUNT(*) AS NombreTotalVoituresVendus
FROM Voiture;
```

```
SELECT COUNT(Matricule) AS NombreTotalVoituresVendus
FROM Voiture;
```

2/- Donnez le nombre total de voitures vendus aux enchères par marque.


```
SELECT Marque, COUNT(*) AS NombreTotalVoituresVendus
FROM Voiture
GROUP BY Marque;
```

3/- Donnez le nombre total de voitures vendus aux enchères par marque et par modèle.

```
SELECT Marque, Modèle, COUNT(*) AS NombreTotalVoituresVendus
FROM Voiture
GROUP BY Marque, Modèle;
```

4/- Donnez la recette des ventes aux enchères de toute les voitures.

```
SELECT SUM(PrixVente) AS TotalVoituresVendus
FROM Voiture;
```

5/- Donnez la recette des ventes aux enchères par l'année mise en circulation.

```
SELECT AnnéeMiseCirculation, SUM(PrixVente) AS TotalVoituresVendus
FROM Voiture
GROUP BY AnnéeMiseCirculation;
```

6/- Donnez le kilométrage moyen des voitures vendus aux enchères.

```
SELECT AVG(Kilométrage) AS KilométrageMoyenVoituresVendus
FROM Voiture;
```

7/- Donnez le kilométrage moyen des voitures par marque.

```
SELECT Marque, AVG(Kilométrage) AS KilométrageMoyenVoituresVendus
FROM Voiture
GROUP BY Marque;
```

EXO 3 :

Soit la relation « Employé » suivante :

Employer (Nom, Prénom, Ville, Salaire)

```
CREATE TABLE Employer (
    Nom VARCHAR(15) NOT NULL,
    Prénom VARCHAR(15) NOT NULL,
    Ville VARCHAR(15) NOT NULL,
    Salaire INT NOT NULL,
    PRIMARY KEY (Nom)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
```

INSERT INTO Employer (Nom, Prénom, Ville, Salaire)
VALUES

('Nom1', 'Prénom1', 'Alger', 15000),
('Nom2', 'Prénom2', 'Alger', 15000),
('Nom3', 'Prénom3', 'Alger', 15000),
('Nom4', 'Prénom4', 'Alger', 15000),
('Nom5', 'Prénom5', 'Alger', 15000),
('Nom6', 'Prénom6', 'Alger', 15000),
('Nom7', 'Prénom7', 'Alger', 15000),
('Nom8', 'Prénom8', 'Alger', 15000),
('Nom9', 'Prénom9', 'Alger', 15000),
('Nom10', 'Prénom10', 'Oran', 10000),
('Nom11', 'Prénom11', 'Oran', 10000),
('Nom12', 'Prénom12', 'Oran', 10000),
('Nom13', 'Prénom13', 'Oran', 10000),
('Nom14', 'Prénom14', 'Oran', 10000),
('Nom15', 'Prénom15', 'Oran', 10000),
('Nom16', 'Prénom16', 'Oran', 10000),
('Nom17', 'Prénom17', 'Oran', 10000),
('Nom18', 'Prénom18', 'Oran', 10000),
('Nom19', 'Prénom19', 'Oran', 10000),
('Nom20', 'Prénom20', 'Oran', 10000),
('Nom21', 'Prénom21', 'Bejaia', 20000),
('Nom22', 'Prénom22', 'Bejaia', 20000),
('Nom23', 'Prénom23', 'Bejaia', 20000),
('Nom24', 'Prénom24', 'Bejaia', 20000),
('Nom25', 'Prénom25', 'Bejaia', 20000),
('Nom26', 'Prénom26', 'Bejaia', 20000),
('Nom27', 'Prénom27', 'Bejaia', 20000),
('Nom28', 'Prénom28', 'Bejaia', 20000),
('Nom29', 'Prénom29', 'Bejaia', 20000),
('Nom30', 'Prénom30', 'Bejaia', 20000) ;

1/-Donnez le nombre total des employés.

SELECT COUNT(*) AS NombreTotalEmployes
FROM Employer;

2/-Combien de personnes sont embauchées par ville ?

SELECT Ville, COUNT(*) AS NombreTotalEmployés
FROM Employer
GROUP BY Ville;

3/-Quelle ville embauche le plus de personne ?

```
SELECT Ville, MAX(COUNT(*))  
FROM Employer  
GROUP BY Ville ;
```

Le **MAX** (**COUNT**(.)) n'est pas autorisé avec MySQL (car MySQL n'autorise pas plusieurs niveaux d'agrégation dans une seule sélection. La première agrégation permet d'avoir le nombre de ligne pour chaque groupe, la seconde agrégation est construite sur la première et calcul le Max à partir des valeurs obtenues par la première agrégation).

La bonne façon d'écrire cette requête est d'utiliser une sous-requête:

```
SELECT Ville, COUNT(Nom)  
FROM Employer  
GROUP BY Ville  
HAVING COUNT(Nom) = (SELECT MAX(NbrEmployéVille)  
FROM (SELECT Ville, COUNT(*) AS NbrEmployéVille  
FROM Employer  
GROUP BY Ville) AS MaSousRequête);
```

« **MaSousRequête** » est un alias requis pour la sous-requête.

On peut également faire :

```
SELECT Ville, COUNT(*)  
FROM Employer  
GROUP BY Ville  
ORDER BY COUNT(*) DESC  
LIMIT 1;
```

+ Options	
Ville	COUNT(Nom)
Oran	11

4/-Donnez la moyenne des salaires par ville.

```
SELECT Ville, AVG(Salaire) AS MoyenSalaire  
FROM Employer  
GROUP BY Ville;
```

			Ville	MoyenSalaire
<input type="checkbox"/>	Éditer	Copier	Supprimer	Alger 15000.0000
<input type="checkbox"/>	Éditer	Copier	Supprimer	Bejaia 20000.0000
<input type="checkbox"/>	Éditer	Copier	Supprimer	Oran 10000.0000

5/-Donnez le salaire maximal par ville.

```
SELECT Ville, MAX(Salaire)
FROM Employer
GROUP BY Ville;
```



	Ville	MAX(Salaire)
<input type="checkbox"/> Edit <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Alger	15000
<input type="checkbox"/> Edit <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Bejaia	20000
<input type="checkbox"/> Edit <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	Oran	10000

6/-Donnez toutes les informations des employés dont le salaire est supérieur ou égale au salaire maximal de leur ville.

```
SELECT *
FROM Employer E1
WHERE E1.Salaire >= (SELECT MAX(Salaire)
                     FROM Employer E2
                     WHERE E1.Ville=E2.Ville);
```

EXO 4 :

Soit la relation « Vente » suivante :

Vente (NumVente, NomCommerciale, NbrArtcileVendu, JourVente)

```
CREATE TABLE Vente (
    NumVente INT NOT NULL,
    NomCommerciale VARCHAR(15) NOT NULL,
    NbrArtcileVendu INT NOT NULL,
    JourVente VARCHAR(15) NOT NULL,
    PRIMARY KEY (NumVente)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
```

```
INSERT INTO Vente (NumVente, NomCommerciale, NbrArtcileVendu, JourVente)
VALUES
```

```
(1, 'Samir', 3, 'Samedi'),
(2, 'Mohammed', 6, 'Samedi'),
(3, 'Anis', 2, 'Samedi'),
(4, 'Maya', 1, 'Samedi'),
(5, 'Samir', 2, 'Samedi'),
(6, 'Samir', 1, 'Samedi'),
(7, 'Maya', 2, 'Dimanche'),
(8, 'Anis', 1, 'Dimanche'),
(9, 'Anis', 2, 'Dimanche'),
```

(10, 'Mohammed', 1, 'Dimanche'),
 (11, 'Anis', 1, 'Lundi'),
 (12, 'Maya', 2, 'Lundi'),
 (13, 'Samir', 1, 'Lundi'),
 (14, 'Samir', 3, 'Mardi'),
 (15, 'Mohammed', 1, 'Mercredi'),
 (16, 'Maya', 1, 'Mercredi'),
 (17, 'Mohammed', 2, 'Mercredi'),
 (18, 'Mohammed', 1, 'Jeudi'),
 (19, 'Anis', 1, 'Jeudi'),
 (20, 'Samir', 2, 'Jeudi'),
 (21, 'Samir', 3, 'Jeudi'),
 (22, 'Maya', 1, 'Jeudi') ;

1/-Donnez le nombre de vente réalisé par commerciale.

```
SELECT NomCommerciale, COUNT(*)
FROM Vente
GROUP BY NomCommerciale;
```

NomCommerciale	COUNT(*)
Anis	5
Maya	5
Mohammed	5
Samir	7

2/-Donnez le nombre total d'article vendu par commerciale.

```
SELECT NomCommerciale, SUM(NbrArtcileVendu)
FROM Vente
GROUP BY NomCommerciale;
```

NomCommerciale	SUM(NbrArtcileVendu)
Anis	7
Maya	7
Mohammed	11
Samir	15

3/-Donnez le nombre total d'article vendu par commerciale et par jour de semaine.

```
SELECT NomCommerciale, JourVente, SUM(NbrArtcileVendu)
FROM Vente
GROUP BY NomCommerciale, JourVente;
```

NomCommerciale	JourVente	SUM(NbrArtcileVendu)
Anis	Dimanche	3
Anis	Jeudi	1
Anis	Lundi	1
Anis	Samedi	2
Maya	Dimanche	2
Maya	Jeudi	1
Maya	Lundi	2
Maya	Mercredi	1
Maya	Samedi	1
Mohammed	Dimanche	1
Mohammed	Jeudi	1
Mohammed	Mercredi	3
Mohammed	Samedi	6
Samir	Jeudi	5
Samir	Lundi	1
Samir	Mardi	3
Samir	Samedi	6