SQL

Khadiri Marouan

**Atelier 1**

1/ Créer la base donnée sous nom "boutique".

**CREATE DATABASE boutique**

2/ Création des tables (Manufacturers, Products).

CREATE TABLE Manufacturers (

Code INTEGER,

Name VARCHAR(255) NOT NULL,

PRIMARY KEY (Code)

);

CREATE TABLE Products (

Code INTEGER,

Name VARCHAR(255) NOT NULL ,

Price DECIMAL NOT NULL ,

Manufacturer INTEGER NOT NULL,

PRIMARY KEY (Code),

FOREIGN KEY (Manufacturer) REFERENCES Manufacturers(Code)

) ENGINE=INNODB;

3/ Insérer dans le tableau Manufacturers les valeurs :

**INSERT** **INTO** Manufacturers ( Code, Name , Price , Manufacturer)

**VALUES** ( 11, 'Loudspeakers' , 70 , 2 );

(Code, Name)(1,Sony)(2,Creative Labs)(3,Hewlett-Packard)(4,Iomega)(5,Fujitsu)(6,Winchester)

4/ Insérer dans le tableau Products :

(Code,Name,Price,Manufacturer) (1,Hard drive,240,5) (2,Memory,120,6) (3,ZIP drive,150,4) (4,Floppy disk,5,6) (5,Monitor,240,1)

(6,DVD drive,180,2) VALUES(7,CD drive,90,2) VALUES(8,Printer,270,3) (9,Toner cartridge,66,3) (10,DVD burner,180,2)

**INSERT** **INTO** Products( Code, Name , Price , Manufacturer)

**VALUES** ( 11, 'Loudspeakers' , 70 , 2 );

5/ Sélectionnez les noms de tous les produits du magasin.

**SELECT** Name **FROM** Products;

6/ Sélectionner les noms et les prix de tous les produits du magasin.

**SELECT** Name, Price **FROM** Products;

7/ Sélectionner le nom des produits dont le prix est inférieur ou égal à 200 $.

**SELECT** Name **FROM** Products **WHERE** Price <= 200;

8/ Sélectionnez tous les produits dont le prix est compris entre 60 et 120 dollars.

**SELECT** \* **FROM** Products

**WHERE** Price >= 60 **AND** Price <= 120;

**SELECT** \* **FROM** Products

**WHERE** Price **BETWEEN** 60 **AND** 120;

9/ Sélectionnez le nom et le prix en cents (c'est-à-dire que le prix doit être multiplié par 100).

**SELECT** Name, Price \* 100 **FROM** Products;

**SELECT** Name, Price \* 100 **AS** PriceCents **FROM** Products;

10/ Calculer le prix moyen de tous les produits.

**SELECT** **AVG**(Price) **FROM** Products;

11/ Calculer le prix moyen de tous les produits dont le code fabricant est égal à 2.

**SELECT** **AVG**(Price) **FROM** Products **WHERE** Manufacturer=2;

12/ Calculer le nombre de produits dont le prix est supérieur ou égal à 180 dollars.

**SELECT** **COUNT**(\*) **FROM** Products **WHERE** Price >= 180;

13/ Sélectionner le nom et le prix de tous les produits dont le prix est supérieur ou égal à 180 dollars, et trier d'abord par prix (par ordre décroissant), puis par nom (par ordre croissant).

**SELECT** Name, Price

**FROM** Products

**WHERE** Price >= 180

**ORDER** **BY** Price **DESC**, Name;

14/ Sélectionnez toutes les données des produits, y compris toutes les données relatives au fabricant de chaque produit.

**SELECT** \* **FROM** Products, Manufacturers

**WHERE** Products.Manufacturer = Manufacturers.Code;

**SELECT** \*

**FROM** Products **INNER** **JOIN** Manufacturers

**ON** Products.Manufacturer = Manufacturers.Code;

15/ Sélectionnez le nom du produit, le prix et le nom du fabricant de tous les produits.

**SELECT** Products.Name, Price, Manufacturers.Name

**FROM** Products, Manufacturers

**WHERE** Products.Manufacturer = Manufacturers.Code;

**SELECT** Products.Name, Price, Manufacturers.Name

**FROM** Products **INNER** **JOIN** Manufacturers

**ON** Products.Manufacturer = Manufacturers.Code;

16/ Sélectionnez le prix moyen des produits de chaque fabricant, en indiquant uniquement le code du fabricant.

**SELECT** **AVG**(Price), Manufacturer

**FROM** Products

**GROUP** **BY** Manufacturer;

17/ Sélectionnez le prix moyen des produits de chaque fabricant, en indiquant le nom du fabricant.

**SELECT** **AVG**(Price), Manufacturers.Name

**FROM** Products, Manufacturers

**WHERE** Products.Manufacturer = Manufacturers.Code

**GROUP** **BY** Manufacturers.Name;

**SELECT** **AVG**(Price), Manufacturers.Name

**FROM** Products **INNER** **JOIN** Manufacturers

**ON** Products.Manufacturer = Manufacturers.Code

**GROUP** **BY** Manufacturers.Name;

18/ Sélectionnez les noms des fabricants dont les produits ont un prix moyen supérieur ou égal à 150 $.

**SELECT** **AVG**(Price), Manufacturers.Name

**FROM** Products, Manufacturers

**WHERE** Products.Manufacturer = Manufacturers.Code

**GROUP** **BY** Manufacturers.Name

**HAVING** **AVG**(Price) >= 150;

19/ Sélectionnez le nom et le prix du produit le moins cher.

**SELECT** name,price

**FROM** Products

**ORDER** **BY** price **ASC**

**LIMIT** 1

20/ Sélectionnez le nom de chaque fabricant ainsi que le nom et le prix de son produit le plus cher.

**SELECT** A.Name, A.Price, F.Name

**FROM** Products A, Manufacturers F

**WHERE** A.Manufacturer = F.Code

**AND** A.Price =

(

**SELECT** **MAX**(A.Price)

**FROM** Products A

**WHERE** A.Manufacturer = F.Code

);

21/ Ajouter un nouveau produit : Loudspeakers, 70 $, manufacter 2

**INSERT** **INTO** Products( Code, Name , Price , Manufacturer)

**VALUES** ( 11, 'Loudspeakers' , 70 , 2 );

22/ Mettre à jour le nom du produit 8 en "laser Print".

**UPDATE** Products

**SET** Name = 'Laser Printer'

**WHERE** Code = 8;

23/ Appliquer une remise de 10 % à tous les produits.

**UPDATE** Products

**SET** Price = Price - (Price \* 0.1);

24/ Appliquer une remise de 10 % à tous les produits dont le prix est supérieur ou égal à 120 $.

**UPDATE** Products

**SET** Price = Price - (Price \* 0.1)

**WHERE** Price >= 120;

**Atelier 2**

-- 2.1 Sélectionnez le nom de famille de tous les employés.

SELECT LastName

FROM Employees

;

-- 2.2 Sélectionnez le nom de famille de tous les employés, sans doublons.

SELECT DISTINCT LastName

FROM Employees

;

-- 2.3 Sélectionnez toutes les données des employés dont le nom de famille est "Smith".

SELECT \*

FROM Employees

WHERE LastName = 'Smith'

;

-- 2.4 Sélectionnez toutes les données des employés dont le nom de famille est "Smith" ou "Doe".

SELECT \*

FROM Employees

WHERE LastName = 'Smith'

OR LastName = 'Doe'

;

SELECT \*

FROM Employees

WHERE LastName IN ('Smith', 'Doe')

;

-- 2.5 Sélectionnez toutes les données des employés qui travaillent dans le département 14.

SELECT \*

FROM Employees

WHERE Department = 14

;

-- 2.6 Sélectionner toutes les données des employés qui travaillent dans le département 37 ou le département 77.

SELECT \*

FROM Employees

WHERE Department = 14

OR Department = 77

;

-- 2.7 Sélectionner toutes les données des employés dont le nom de famille commence par un "S".

SELECT \*

FROM Employees

WHERE LastName LIKE 'S%'

;

-- 2.8 Sélectionner la somme des budgets de tous les départements.

SELECT SUM(Budget)

FROM Departments

;

-- 2.9 Sélectionnez le nombre d'employés dans chaque département (vous devez seulement indiquer le code du département et le nombre d'employés).

SELECT D.Code, COUNT(\*)

FROM Employees AS E, Departments AS D

GROUP BY D.Code

;

-- 2.10 Sélectionnez toutes les données des employés, y compris les données du département de chaque employé.

SELECT \*

FROM Employees AS E

INNER JOIN Departments AS D

ON D.Code = E.Department

;

-- 2.11 Sélectionnez le nom et le prénom de chaque employé, ainsi que le nom et le budget du département de l'employé.

SELECT E.Name, E.LastName, D.Name, D.Budget

FROM Employees AS E

INNER JOIN Departments AS D

ON E.Department = D.Code

;

-- 2.12 Sélectionnez le nom et le nom de famille des employés travaillant pour des ministères dont le budget est supérieur à 60 000 $.

SELECT E.Name, E.LastName

FROM Employees AS E

INNER JOIN Departments AS D

ON E.Department = D.Code

AND D.Budget > 60000

;

-- 2.13 Sélectionnez les départements dont le budget est supérieur au budget moyen de l'ensemble des départements.

SELECT \*

FROM Departments

WHERE Budget > (SELECT AVG(D.Budget)

FROM Departments AS D

)

;

-- 2.14 Sélectionnez les noms des départements ayant plus de deux employés.

SELECT D.Name

FROM Departments AS D

INNER JOIN Employees AS E

ON E.Department = D.Code

GROUP BY D.Code

HAVING COUNT(E.SSN) > 2

;

-- 2.15 Très important - Sélectionnez le nom et le nom de famille des employés travaillant pour les ministères dont le budget est le deuxième plus bas.

SELECT E.Name, E.LastName

FROM Departments AS D

INNER JOIN Employees AS E

ON E.Department = D.Code

ORDER BY D.Budget

LIMIT 1 OFFSET 1

;

-- 2.16 Ajoutez un nouveau service appelé "Quality assurance", avec un budget de 40 000 $ et le code de service

11. Et ajoutez un employé appelé "Mary Moore" dans ce département, avec le code SSN 847-21-9811.

INSERT INTO Departments(Name, Budge, Code)

VALUES ('Quality Assurance', 40000, 11)

;

INSERT INTO Employees(SSN, Name, LastName, Department)

VALUES ('847219811', 'Mary Moore', 'Moore', 11)

;

-- 2.17 Réduire le budget de tous les départements de 10 %.

UPDATE Departments

SET Budget = Budget \* 0.9

;

-- 2.18 Réaffecter tous les employés du département de la recherche (code 77) au département informatique (code 14).

UPDATE Employees

SET Department = 14

WHERE Department = 77

;

-- 2.19 Supprimer du tableau tous les employés du département informatique (code 14).

DELETE FROM Employees

WHERE Department = 14

;

-- 2.20 Supprimer du tableau tous les employés qui travaillent dans des départements dont le budget est supérieur ou égal à 60 000 $.

DELETE FROM Employees

WHERE Department IN

(SELECT Code FROM Departments

WHERE Budget >= 60000

)

;

-- 2.21 Supprimer du tableau tous les employés.

DELETE FROM Employees;

**Atelier 3**

--3.1 Sélectionner tous les entrepôts.

**SELECT** \* **FROM** Warehouses;

--3.2 Sélectionnez toutes les cases dont la valeur est supérieure à 150 $.

**SELECT** \* **FROM** Boxes

**WHERE** Value > 150;

--3.3 Sélectionner tous les contenus distincts dans toutes les cases.

**SELECT** **DISTINCT** Contents

**FROM** Boxes;

--3.4 Sélectionner la valeur moyenne de toutes les boîtes.

**SELECT** **AVG**(Value)

**FROM** Boxes;

--3.5 Sélectionner le code de l'entrepôt et la valeur moyenne des boîtes dans chaque entrepôt.

**SELECT** Warehouse, **AVG**(Value)

**FROM** Boxes

**GROUP** **BY** Warehouse;

--3.6 Identique à l'exercice précédent, mais ne sélectionnez que les entrepôts où la valeur moyenne des boîtes est supérieure à 150.

**SELECT** Warehouse, **AVG**(Value)

**FROM** Boxes

**GROUP** **BY** Warehouse

**HAVING** **AVG**(Value) > 150;

--3.7 Sélectionnez le code de chaque boîte, ainsi que le nom de la ville dans laquelle la boîte est située.

**SELECT** Boxes.Code, **Location**

**FROM** Warehouses **INNER** **JOIN** Boxes

**ON** Warehouses.Code = Boxes.Warehouse;

--3.8 Sélectionnez les codes des entrepôts, ainsi que le nombre de boîtes dans chaque entrepôt.

-- Facultativement, tenez compte du fait que certains entrepôts sont vides (c'est-à-dire que le nombre de boîtes devrait apparaître comme zéro, au lieu d'omettre l'entrepôt du résultat).

**SELECT** Warehouse, **COUNT**(\*)

**FROM** Boxes

**GROUP** **BY** Warehouse;

**SELECT** Warehouses.Code, **COUNT**(Boxes.Code)

**FROM** Warehouses **LEFT** **JOIN** Boxes

**ON** Warehouses.Code = Boxes.Warehouse

**GROUP** **BY** Warehouses.Code;

--3.9 Sélectionnez les codes de tous les entrepôts qui sont saturés (un entrepôt est saturé si le nombre de boîtes qu'il contient est supérieur à la capacité de l'entrepôt).

**SELECT** Code

**FROM** Warehouses

**WHERE** Capacity <

(

**SELECT** **COUNT**(\*)

**FROM** Boxes

**WHERE** Warehouse = Warehouses.Code

);

**SELECT** Warehouses.Code

**FROM** Warehouses **JOIN** Boxes **ON** Warehouses.Code = Boxes.Warehouse

**GROUP** **BY** Warehouses.code

**HAVING** **Count**(Boxes.code) > Warehouses.Capacity

--3.10 Sélectionnez les codes de toutes les boîtes situées à Chicago.

**SELECT** Boxes.Code

**FROM** Warehouses **RIGHT** **JOIN** Boxes

**ON** Warehouses.Code = Boxes.Warehouse

**WHERE** **Location** = 'Chicago';

*/\* With a subquery \*/*

**SELECT** Code

**FROM** Boxes

**WHERE** Warehouse **IN**

(

**SELECT** Code

**FROM** Warehouses

**WHERE** **Location** = 'Chicago'

);

--3.11 Créer un nouvel entrepôt à New York avec une capacité de 3 boîtes.

**INSERT**

**INTO** Warehouses

(**Location**,Capacity)

**VALUES** ('New York',3);

--3.12 Créer une nouvelle boîte, avec le code "H5RT", contenant des "Papers" d'une valeur de 200 $, et située dans l'entrepôt 2.

**INSERT** **INTO** Boxes

**VALUES**('H5RT','Papers',200,2);

--3.13 Réduire la valeur de toutes les boîtes de 15 %.

**UPDATE** Boxes **SET** Value = Value \* 0.85;

--3.14 Retirer toutes les boîtes d'une valeur inférieure à 100 $.

**UPDATE** Boxes **SET** Value = Value \* 0.80

**WHERE** Value > (**SELECT** **AVG**(Value) **FROM** (**SELECT** \* **FROM** Boxes) **AS** X);

-- 3.15 Retirer toutes les boîtes des entrepôts saturés.

**DELETE** **FROM** Boxes **WHERE** Value < 100;

-- 3.16 Ajouter un indice pour la colonne "Entrepôt" dans le tableau "boîtes".

**DELETE** **FROM** Boxes **WHERE** Warehouse **IN**

(

**SELECT** Code

**FROM** Warehouses

**WHERE** Capacity <

(

**SELECT** **COUNT**(\*)

**FROM** Boxes

**WHERE** Warehouse = Warehouses.Code

)

);

**Atelier 4**

-- 4.1 Sélectionnez le titre de tous les films.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Title FROM Movies

-- 4.2 Afficher toutes les classifications distinctes dans la base de données.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) DISTINCT Rating FROM Movies

-- 4.3 Afficher tous les films non classés.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM Movies WHERE Rating [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#operator_is) NULL

-- 4.4 Sélectionner tous les cinémas qui ne diffusent pas de film actuellement.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM MovieTheaters WHERE Movie [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#operator_is) NULL

-- 4.5 Sélectionner toutes les données de tous les cinémas

-- En outre, les données du film qui est projeté en salle (si un film est projeté).

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM MovieTheaters [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html#function_left) JOIN Movies ON MovieTheaters.Movie = Movies.Code

-- 4.6 Sélectionnez toutes les données de tous les films et, si ce film est diffusé dans une salle de cinéma, affichez les données de la salle de cinéma.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM MovieTheaters [RIGHT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html#function_right) JOIN Movies ON MovieTheaters.Movie = Movies.Code

-- 4.7 Afficher les titres des films qui ne sont actuellement diffusés dans aucun cinéma.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Title FROM Movies WHERE Code [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ( [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Movie FROM MovieTheaters WHERE Movie [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#operator_is) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL )

-- 4.8 Ajouter le film non coté "One, Two, three".

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO Movies(Code, Title,Rating) [VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values)(9,'One, Two, Three',NULL)

-- 4.9 Fixer la classification de tous les films non classés à "G".

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) Movies [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) Rating='G' WHERE Rating [IS](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#operator_is) NULL

-- 4.10 Supprimer les salles de cinéma qui projettent des films classés "NC-17".

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html) FROM MovieTheaters WHERE Movie [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Code FROM Movies WHERE Rating = 'NC-17')

**Atelier 5**

-- 5.1 Sélectionnez le nom de toutes les pièces.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Name FROM Pieces

-- 5.2 Sélectionnez les données de tous les fournisseurs(providers).

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM Providers

-- 5.3 Obtenir le prix moyen de chaque pièce (ne montrer que le code de la pièce et le prix moyen).

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Piece, [AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_avg)(Price) FROM Provides GROUP BY Piece

-- 5.4 Obtenir les noms de tous les fournisseurs qui fournissent la pièce 1.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Name FROM Providers WHERE Code [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Provider FROM Provides WHERE Piece = 1)

-- 5.5 Sélectionner le nom des pièces fournies par le fournisseur avec le code "HAL".

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Name FROM Pieces WHERE Code [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ([SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Piece FROM Provides WHERE Provider = 'HAL')

-- 5.6 Pour chaque pièce, trouvez l'offre la plus chère de cette pièce et indiquez le nom de la pièce, le nom du fournisseur et le prix

-- (notez qu'il peut y avoir deux fournisseurs qui fournissent la même pièce au prix le plus élevé).

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) Pieces.Name, Providers.Name, Price FROM Pieces INNER JOIN Provides ON Pieces.Code = Piece INNER JOIN Providers ON Providers.Code = Provider WHERE Price = ( [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [MAX](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_max)(Price) FROM Provides WHERE Piece = Pieces.Code )

-- 5.7 Ajouter une entrée dans la base de données pour indiquer que "Skellington Supplies" (code "TNBC") fournira des pignons (code "1") pour 7 cents chacun.

[INSERT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/insert.html) INTO Provides [VALUES](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/miscellaneous-functions.html#function_values) (1, 'TNBC', 7)

-- 5.8 Augmenter tous les prix d'un cent.

[UPDATE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/update.html) Provides [SET](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/set.html) Price = Price + 1

-- 5.9 Mettre à jour la base de données pour indiquer que "Susan Calvin Corp. (code "RBT") ne fournira pas de boulons (code 4).

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html) FROM Provides WHERE Provider = 'RBT' [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Piece = 4

-- 5.10 Mettre à jour la base de données pour refléter le fait que "Susan Calvin Corp. (code "RBT") ne fournira pas de boulons (code 4).

[DELETE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/delete.html) FROM Provides WHERE Provider = 'RBT'

**Atelier 6**

-- 6.1 Listez les noms de tous les scientifiques, les noms de leurs projets,

-- et les heures travaillées par ce scientifique sur chaque projet,

-- dans l'ordre alphabétique du nom du projet, puis du nom du scientifique.

|  |
| --- |
| SELECT scientists.Name, projects.Name, projects.Hours FROM Scientists |
| INNER JOIN AssignedTo ON scientists.SSN = assignedto.Scientist |
| INNER JOIN Projects ON assignedto.Project = projects.Code |
| ORDER BY projects.Name ASC, scientists.Name ASC; |

-- 6.2 Sélectionnez les noms de projets qui ne sont pas encore attribués

SELECT Name FROM Projects WHERE Code NOT In ( SELECT Project FROM AssignedTo );