



## LE LANGAGE SQL

Exercice 4: Création de table

1. À partir des données présentées dans le tableau suivant, proposez les définitions de la table MAINTENANCE

Jour	Machine	Numéro	Vitesse	Température	Heure	Événement
Ven	Massicot	147			21:18	Défaut de lame
Sam	Relieuse	63	16		16:15	Arrêt pour maintenance
Jeu	Presse	87	6	62	11:40	Bavure encre
Sam	Relieuse	79	16		17:11	Reprise
Mer	Presse	89	6	55	08:28	Recadrage
Mar	Presse	132	8	68	09:58	Changement encre
Mer	Massicot	111			10:17	Graissage coulisseau

```
CREATE TABLE MAINTENANCE
(
  NUMERO INTEGER PRIMARY KEY,
  MACHINE CHAR(16) NOT NULL,
  JOUR CHAR(3) NOT NULL
  CHECK (VALUE IN
        ('Lun', 'Mar', 'Mer', 'Jeu', 'Ven', 'Sam', 'Dim')),
  HEURE TIME NOT NULL,
  EVENEMENT VARCHAR(64) NOT NULL,
  VITESSE FLOAT CHECK (VALUE > 0),
  TEMPERATURE FLOAT CHECK (VALUE > -273)
  CONSTRAINT UK_MAC_JOUR_HEUR UNIQUE (MACHINE, JOUR, HEURE)
);
```

Donnez la commande de création des tables contenues dans le schéma relationnel suivant (les clés primaires sont soulignées et les clés étrangères en italique) :

- `personne(id_personne, id_secu, nom, prenom, dateNaissance)`
- `cours(id_cours, id_enseignant, sigle, intitule, description)` où *id\_enseignant* est une clé étrangère qui fait référence au schéma de relation `personne`.
- `Suivre(id_etudiant, id_cours, note)` où *id\_etudiant* et *id\_cours* sont des clés étrangères qui font, respectivement, référence aux tables `personne` et `cours`.

**PERSONNE(ID\_PERSONNE, ID\_SECU,  
NOM, PRENOM, DATENAISSANCE)**

```
CREATE TABLE personne (  
    id_personne INTEGER PRIMARY KEY,  
    num_secu CHAR(15) UNIQUE,  
    nom VARCHAR(200) NOT NULL,  
    prenom VARCHAR(200) NOT NULL,  
    dateNaissance DATE  
);
```

**COURS(ID\_COURS, ID\_ENSEIGNANT,  
SIGLE, INTITULE, DESCRIPTION)**

```
CREATE TABLE cours (  
    id_cours INTEGER PRIMARY KEY,  
    id_enseignant INTEGER REFERENCES personne (id_personne)  
        ON DELETE SET NULL, sigle CHAR (7) UNIQUE NOT NULL,  
    intitule VARCHAR(250) NOT NULL,  
    description VARCHAR(200)  
);
```

**SUIVRE(ID\_ETUDIANT, ID\_COURS,  
NOTE)**

```
CREATE TABLE suivre (  
    id_personne INTEGER REFERENCES personne ON DELETE  
        RESTRICT,  
    id_cours INTEGER REFERENCES cours ON DELETE  
        RESTRICT, PRIMARY KEY (id_etudiant, id_cours),  
    Note REAL CHECK (note >0 OR note IS NULL));
```

## INTERROGATION DE BASE DE DONNÉE : LIVRAISON DE JOURNAUX

1-Donner pour chaque journal ses titres, prix, type et périodicité.

```
select titre, prix, type, periode from journal;
```

2-Donner les différents journaux triés par ordre alphabétique.

```
select titre from journal order by titre;
```

3-Codes des journaux mensuels.

```
select code_j from journal where periode = 'MENSUEL';
```

4-Tous renseignements connus sur journaux ordonnés par type et pour chaque type par prix.

```
select * from journal order by type, prix;
```

5-Prix minimum, moyen et maximum d'un journal.

```
select min(prix), avg(prix), max(prix) from journal;
```

6-Codes des journaux dont le type est inconnu.

```
select code_j from journal where type is null;
```



## INTERROGATION DE BASE DE DONNÉE : LIVRAISON DE JOURNAUX

7-. Pour chaque type de journal, prix minimum, moyen et maximum.

```
select type, min(prix), avg(prix), max(prix) from journal group by type;
```

8-Numéros des dépositaires dont la moyenne par livraison de quantité de journaux livrés est supérieure à 15.

```
select n_dep from livraison group by n_dep having avg(qte_l) > 15;
```

9-Nombre des villes où sont domiciliés au moins 4 journaux.

```
select count(distinct adr_j)from journal where adr_j in (select adr_j from journal group by  
adr_j having count(distinct code_j) >= 4);
```

10-Codes des journaux livrés au dépositaire Les Brasseurs.

```
select distinct code_j from livraison, depot where livraison.n_dep=depot.n_dep and  
nom_dep like 'LES BRASSEURS%';
```

## INTERROGATION DE BASE DE DONNÉE : LIVRAISON DE JOURNAUX

11-. Titre de ces journaux.

```
select distinct titre from journal, livraison, depot where  
livraison.n_dep=depot.n_dep and journal.code_j=livraison.code_j and  
nom_dep like 'LES BRASSEURS%';
```

12- Nombre de ces journaux

```
select count(distinct titre), sum(qte_1) from journal, livraison, depot  
where livraison.n_dep=depot.n_dep and journal.code_j=livraison.code_j  
and nom_dep like 'LES BRASSEURS%';
```

13-Codes et titres des journaux dont le prix est supérieur à deux fois le prix minimum.

```
select distinct code_j, titre from journal where prix > (select 2*min(prix)  
from journal);
```

## INTERROGATION DE BASE DE DONNÉE : LIVRAISON DE JOURNAUX

14-Numéros des dépositaires qui reçoivent plusieurs journaux.

```
select n_dep from livraison group by n_dep having count(distinct  
code_j) > 1;
```

15-Donner les journaux que le dépositaire 007 ne reçoit pas (sans minus).

```
select code_j from journal where code_j not in (select code_j from  
livraison where n_dep = '007');
```

16-Donner les journaux que le dépositaire 007 reçoit.

```
select code_j from livraison where n_dep = '007';
```

17-Nombre de journaux dans la base, et types de journaux.

```
select count(distinct titre), count(distinct type) from journal;
```

## INTERROGATION DE BASE DE DONNÉE : LIVRAISON DE JOURNAUX

18-Villes où sont domiciliés au moins 5 journaux.

```
select adr_j from journal group by adr_j having count(distinct code_j)  
      >= 5;
```

19-Codes des journaux livrés à Bures (2 manières).

```
select distinct code_j from depot, livraison where depot.n_dep =  
      livraison.n_dep and adr like 'BURES%';
```

20-Pour chaque journal repéré par son code la liste des numéros des  
dépositaires auxquels il est livré.

```
select journal.code_j, n_dep from journal, livraison where journal.code_j  
      = livraison.code_j order by journal.code_j;
```

# تعليم - e