

ET 6 Ex3 4 Eléments de correction

Exercice 3

1
89₁₀ correspond à 1011001₂

2
11001110
⊕ 01101011

10100101

3

```
def xor_crypt(message, cle):  
    lst = []  
    for i in range(len(message)):  
        n = xor(ord(message[i]), ord(cle[i]))  
        lst.append(n)  
    return lst
```

4

```
def generer_cle(mot,n):  
    cle = ""  
    long_mot = len(mot)  
    long_n = len(n)  
    div = long_mot // long_n  
    reste = long_mot % long_n  
    for i in range(div):  
        cle = cle + n  
    for i in range(reste):  
        cle = cle + n[i]  
    return cle
```

5

E1	E2	E1 XOR E2	(E1 XOR E2) XOR E2
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

On remarque que (E1 XOR E2) XOR E2 correspond à E1.

Soit le message d'origine m, soit le message chiffré m' et soit la clé k.

Pour obtenir m' il faut faire un m XOR k.

Pour obtenir m à partir de m' et k, il suffit donc de faire un m' XOR k (m' correspond à "E1 XOR E2", m correspond à "E1" et k correspond à "E2").

Exercice 4

1a

L'attribut *nom* de la table *licencies* ne peut pas servir de clé primaire, car il peut exister des homonymes et que la clé primaire doit être unique.

1b

L'attribut *id_licencie* peut jouer le rôle de clé primaire.

2a

Cette requête renvoie le nom et le prénom des licenciés qui jouent dans l'équipe des "- 12 ans".

2b

En utilisant * à la place de prenom, nom, on obtient l'ensemble des attributs.

2c

```
SELECT date
FROM matchs
WHERE equipe = 'Vétérans' AND lieu = 'Domicile'
```

3

```
INSERT INTO licencies
(id_licencie, prenom, nom, annee_naissance, equipe)
VALUES
(287, 'Jean', 'Lavenue', 2001, 'Hommes 2')
```

4

```
UPDATE Licencies
SET equipe = 'Vétérans'
WHERE prenom = 'Joseph' AND nom = 'Cu viller'
```

5

```
SELECT nom FROM licencies
JOIN Matchs ON licencies.equipe = matchs.equipe
WHERE adversaire = 'LSC' AND date = 19/06/2021
```