Correction Série n°5

Exercice 1

1. V V F F
2. V F V F
3. V V V V
4. F V V F

Exercice 2

1 → b - 2 → e - 3 → d - 4 → f - 5 → a - 6 → c - 7 → h - 8 → j - 9 → g - 10 → i

Exercice 3

1.

Anomalie	Contrainte
Le champ **Caché** de la table **Dossier** comporte une valeur non autorisée (**Oui**)	Contrainte de domaine
L'**IdDossier** (clé primaire) de la table **Dossier** contient la même valeur **D10015** deux fois.	Contrainte de table
Le champ **idDossier** de la table **Fichier** (clé étrangère) affiche la valeur E20132, qui ne correspond à aucune valeur **IdDossier** de la table **Dossier** (table mère).	Contrainte d'intégrité référentielle

2.

```
UPDATE Dossier SET Caché = 'O' WHERE Caché = 'Oui';
UPDATE Dossier
SET IdDossier = 'D10016'
WHERE IdDossier = 'D10015' AND NomD = 'Films';
INSERT INTO Dossier VALUES ('E20132', 'VSCode', 'D:/', '-rwxr-xr-', 'N');

ALTER TABLE Dossier
ADD CONSTRAINT chkCaché CHECK(Caché = 'O' OR Caché = 'N');

ALTER TABLE Dossier
ADD PRIMARY KEY (IdDossier);

ALTER TABLE Fichier
ADD FOREIGN KEY (idDossier) REFERENCES Dossier(IdDossier)
  ON UPDATE CASCADE ON DELETE CASCADE;
```

3.

```
CREATE TABLE Fichier (
  idF INT PRIMARY KEY AUTO_INCREMENT,
  NomF VARCHAR(30) NOT NULL,
  IdExt VARCHAR(10) NOT NULL,
  Date DATE DEFAULT CURR_TIMESTAMP CHECK(Date < NOW()),
  Taille INT NOT NULL CHECK(Taille >= 0),
  IdDossier VARCHAR(20) NOT NULL,
  FOREIGN KEY (IdDossier) REFERENCES Dossier(IdDossier)
    ON UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (IdExt) REFERENCES Extension(IdExt)
    ON UPDATE CASCADE ON DELETE CASCADE
);
```

4.

1 → Faux - 2 → Faux - 3 → Faux - 4 → Faux

Exercice 4

1.

```
SELECT cl.NumClient, Nom, Ville, Solde
FROM Client AS cl, Compte AS co
WHERE cl.NumClient = co.NumClient AND
  Solde < 0;
```

2.

```
SELECT Numclient, Nom, NumCompte, NumAgence, Solde
FROM Client AS cl, Compte AS co
WHERE cl.NumClient = co.NumClient
ORDER cl.NumClient;
```

3.

```
SELECT *
FROM Client
WHERE NumClient IN (
  SELECT NumClient
  FROM Compte AS co, Agence AS ag
  WHERE co.NumAgence = ag.NumAgence AND Nom = “L'investisseur”
);
```

4.

```
SELECT NumAgence, AVG(Solde) AS SoldeMoyen
FROM Compte
GROUP BY NumAgence
HAVING AVG(Solde) > 10000;
```

5.

```
SELECT NumClient, SUM(Montant) AS MontantEmprunt
FROM Emprunt
GROUP BY NumClient
HAVING SUM(Montant) > 10000;
```

Exercice 5 (Correction proposée par Safa)

1/

```
UPDATE assuré SET genre='H' WHERE genre='Homme';
UPDATE assuré SET genre='F' WHERE genre='Femme';
```

ou

```
UPDATE assuré set genre=LEFT(genre, 1);
```

N.B : la fonction LEFT() est hors programme.

2/

```
ALTER TABLE assuré
MODIFY COLUMN genre VARCHAR(1) CHECK(genre in ('H', 'F'));
```

3/

```
SELECT *
FROM résidence
ORDER BY délégation,gouvernorat;
```

4/

```
SELECT numero, nom, prenom
FROM assuré
WHERE status-mariomonial="divorcés";
```

5/

```
SELECT nom, prenom
FROM assurés
WHERE (YEAR(NOW()) - YEAR(date-naissance)) >= 60;
```

6/

```
SELECT numéro, nom, prénom
FROM résidence AS r, assuré AS a, TypeSecurité AS t,
WHERE r.delegation=a.CodeDelegation  AND
   t.CodeSec=a.CodeSec AND
   t.TypeSec='CNRPS' AND
   gouvernorat = 'ben arous'
ORDER BY nom DESC;
```

7/

```
SELECT count(*) as nbr
FROM assure
WHERE YEAR(date-naissance) = 2000;
```

8/

```
SELECT count(code-delegation) an nbr, codedelegation
FROM assuré
GROUP BY code-delegation
HAVING nbr>100;
```

9/

```
SELECT numéro, nom, prenom
FROM residences AS r,assurer AS a, TypeSecurite AS t
WHERE a.CodeDelegation=r.delegation AND
   a.CodeSec=t.CodeSec AND
   TypeSec="CNSS" AND
   nom like '%A%';
```

10/

```
SELECT numéro, nom, prénom
FROM assuré
WHERE MotPaasse = (
  SELECT motpasse
  FROM assuré
  WHERE nom="benaarous" AND
     prenom="ahmed"
);
```

11/

Non corrigée