

Correction Examen Donation

I.1. Entités/associations

(5.5points) :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> croissant_rouge	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> croissant_rouge_dons	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> don	★ Parcourir Structure Rechercher Insérer Vider Supprimer	6	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> employe	★ Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> entreprise	★ Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	16,0 kio	-
5 tables	Somme	15	InnoDB	utf8mb4_general_ci	128,0 kio	0 o

Table CroissantRouge

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
<input type="checkbox"/> 1	id_croissant_rouge	bigint(20)			Non	Aucun(e)
<input type="checkbox"/> 2	region	varchar(255)	utf8mb4_general_ci		Oui	NULL

Table Don (Avec Foreign Key)

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
<input type="checkbox"/> 1	id_don	bigint(20)			Non	Aucun(e)
<input type="checkbox"/> 2	date	date			Oui	NULL
<input type="checkbox"/> 3	montant	double			Non	Aucun(e)
<input type="checkbox"/> 4	type	enum('Espèces', 'Nature', 'Sang')	utf8mb4_general_ci		Oui	NULL
<input type="checkbox"/> 5	employee_id_employe	bigint(20)			Oui	NULL

Table Association croissantRouge_don

#	Nom	Type
<input type="checkbox"/> 1	croissant_rouge_id_croissant_rouge	bigint(20)
<input type="checkbox"/> 2	dons_id_don	bigint(20)

Table Employe(Avec Foreign Key)

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
<input type="checkbox"/> 1	id_employe	bigint(20)			Non	Aucun(e)
<input type="checkbox"/> 2	nom_employe	varchar(255)	utf8mb4_general_ci		Oui	NULL
<input type="checkbox"/> 3	poste	varchar(255)	utf8mb4_general_ci		Oui	NULL
<input type="checkbox"/> 4	prenom_employe	varchar(255)	utf8mb4_general_ci		Oui	NULL
<input type="checkbox"/> 5	entreprise_id_entreprise	bigint(20)			Oui	NULL

Table Entreprise

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
<input type="checkbox"/> 1	id_entreprise	bigint(20)			Non	Aucun(e)
<input type="checkbox"/> 2	adresse	varchar(255)	utf8mb4_general_ci		Oui	NULL
<input type="checkbox"/> 3	nom_entreprise	varchar(255)	utf8mb4_general_ci		Oui	NULL

I.2. Services (14 points) :

Question 1.

public Company addCompany(Company company);

The screenshot shows a REST client interface. The top bar indicates a POST request to `http://localhost:8089/addEntreprise`. The 'Body' tab is selected, showing a JSON payload: `{ "nomEntreprise": "Actia", "adresse": "Ariana Soghra" }`. The bottom section shows the response, which is a 200 OK status with a JSON body: `{ "idEntreprise": 3, "nomEntreprise": "Actia", "adresse": "Ariana Soghra", "employees": null }`.

id_entreprise	adresse	nom_entreprise
1	Lac1	Sofrecom
2	Ariana Soghra	ESPRIT
3	Ariana Soghra	Actia

Question 2.

**public Employee addEmployeeAndAssignToCompany(Employee employee, String
companyName);**

POST http://localhost:8089/addEmployeeAndAssignToEntreprise/Actia

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

```

1 {
2   "nomEmploye": "Ben Hssine",
3   "prenomEmploye": "Olfa",
4   "poste": "Enseignante"
5 }

```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```

1 {
2   "idEmploye": 6,
3   "nomEmploye": "Ben Hssine",
4   "prenomEmploye": "Olfa",
5   "poste": "Enseignante"
6 }

```

id_employe	nom_employe	poste	prenom_employe	entreprise_id_entreprise
1	Sofrecom	Ingenieur	Lac1	1
2	Hamdi	Ingenieur	Amine	1
3	Ben Hssine	Enseignante	Olfa	2
6	Ben Hssine	Enseignante	Olfa	3

Question 3.

public Donation addDonation(Donation donation);

POST http://localhost:8089/addDon

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

```

1  {
2    ...
3    "date": "2023-11-28",
4    "type": "Sang",
5    "employee": {
6      "idEmployee": 3
7    }
8  }

```

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```

1  {
2    "idDon": 11,
3    "montant": 0.0,
4    "date": "2023-11-28T00:00:00.000+00:00",
5    "type": "Sang",
6    "employee": {
7      "idEmployee": 3,
8      "nomEmployee": null,
9      "prenomEmployee": null,
10     "poste": null
11   }

```

id_don	date	montant	type	employee_id_employe
5	2023-11-28	150	Especes	3
6	2023-11-28	0	Sang	2
7	2023-11-29	0	Sang	3
8	2023-11-29	0	Sang	1
9	2023-11-29	200	Especes	1
10	2023-11-29	1000	Especes	2
11	2023-11-28	0	Sang	3

Question 4.

AOP

```
INFO 9512 --- [nio-8089-exec-8] t.esprit.donation.aspect.LoggingAspect : Merci pour ce don
```

```

@Component
@Aspect
@Slf4j
public class LoggingAspect {

    @AfterReturning("execution(* tn.esprit.donation.services.ServiceIMP.addDon(..))")
    public void logMethodEntry(JoinPoint joinPoint){
        String name = joinPoint.getSignature().getName();
        log.info("Merci pour ce don ");
    }
}

```

Question 5.

public Set getDonationByType(DonationType type) ;

En Utilisant JPQL

```

public interface DonRepository extends JpaRepository<Don,Long> {

    1 usage
    public Set<Don> findByType(TypeDons type) ;
}

```

GET http://localhost:8089/getDonByType/Especes

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "idDon": 5,
4      "montant": 150.0,
5      "date": "2023-11-28",
6      "type": "Especes",
7      "employee": {
8        "idEmploye": 3,
9        "nomEmploye": "Ben Hssine",
10       "prenomEmploye": "Olfa",
11       "poste": "Enseignante"
12     }
13   },
14   {
15     "idDon": 9,
16     "montant": 200.0,
17     "date": "2023-11-29",
18     "type": "Especes",
19     "employee": {
20       "idEmploye": 1,
21       "nomEmploye": "Sofrecom",
22       "prenomEmploye": "Lac1",

```

Question 6.

```
public void getEmployeeByDonation() ;
```

```
2023-12-08T19:06:15.012+01:00 INFO 4508 --- [ scheduling-1] tn.esprit.donation.services.ServiceIMP : Le meilleur employé du mois est : Ben Hssine
2023-12-08T19:06:30.004+01:00 INFO 4508 --- [ scheduling-1] tn.esprit.donation.services.ServiceIMP : Le meilleur employé du mois est : Ben Hssine
2023-12-08T19:06:45.007+01:00 INFO 4508 --- [ scheduling-1] tn.esprit.donation.services.ServiceIMP : Le meilleur employé du mois est : Ben Hssine
```

Question 7.

```
public List getEmployeeByArea(String companyName , String area) ;
```

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8089/getEmployeeByRegion/Ariana/Sofrecom
- Query Params:** A table with two columns: Key and Value.
- Body:** Pretty view of a JSON object:

```
{  "idEmployee": 2,  "nomEmployee": "Hamdi",  "prenomEmployee": "Amine",  "poste": "Ingenieur"}
```
- Status:** 200 OK

Question 8.

En utilisant JPQL

```
public Float getTotalDonation(Date date1, Date date2) ;
```

```
1 usage
@Query("select SUM(d.montant) from Don d where d.date BETWEEN :date1 and :date2")
public Float getTotalByDon(@Param("date1") Date date1, @Param("date2") Date date2);
```

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:8089/getTotalDonation/2023-11-01/2023-12-01
- Query Params:** A table with two columns: Key and Value.
- Body:** Pretty view of a float value:

```
1 150.0
```
- Status:** 200 OK