

## TD 7 – Reverse Engineering

Ce TD fait l'objet d'un rapport :

Nom du document : B2-SLAM1-TD7-NOM-Prenom.pdf

Adresse destinataire : dev.sio.lla@gmail.com

Objet du message : [B2][SLAM1] TD7

Ce TD aborde deux concepts :

- Le *Reverse Engineering* (partie 2) ;
- Les fausses ternaires (partie 3).

### 1- Contexte – Application Gazouille

---

*Gazouille* est un service de *microblogage* ou *microblogging*, qui permet à ses utilisateurs (*gazouilleurs*) de bloguer grâce à de courts messages, des *gazouillis*. Outre cette concision imposée, la principale différence entre *Gazouille* et un blog traditionnel réside dans le fait que *Gazouille* n'invite pas les lecteurs à commenter les messages postés. La promesse d'origine de *Gazouille*, "What are you doing?", le définit comme un service permettant de raconter ce qu'on fait au moment où on le fait. Prenant acte de l'utilisation du service pour s'échanger des informations et des liens, *Gazouille* le remplace par "What's happening ?".

### 2- Analyse de la base de données existante

---

Le script de création de la base de données MySQL/MariaDB *gazouille* (fichier *creerGazouille.sql*) est hébergé dans le dépôt *Github* dont l'URL est <https://github.com/dev-HR/gazouille.git>.

#### Exercice 1 – Préparation

1. Cloner le dépôt distant dédié à la base de données *gazouille* puis exécuter le script *creerGazouille.sql* ;

```
devadmin@dev-web:~$ ls
B1-Asla-Fonctions  2  B1-SQL  Club-hippique-La-licorne  Images  Musique  sanaya-bio-bd  SLAM1-02  SLAM1-05  test
B1-Asla-Tab-Assoc  Bureau  Documents  Modèles  Public  SLAM1-01  SLAM1-03  Téléchargements  Vidéos
devadmin@dev-web:~$ git clone https://github.com/dev-HR/gazouille.git
Clonage dans 'gazouille'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Dépaquetage des objets: 100% (6/6), fait.
devadmin@dev-web:~$ ls
B1-Asla-Fonctions  2  B1-SQL  Club-hippique-La-licorne  gazouille  Modèles  Public  SLAM1-01  SLAM1-03  Téléchargements  Vidéos
B1-Asla-Tab-Assoc  Bureau  Documents  Images  Musique  sanaya-bio-bd  SLAM1-02  SLAM1-05  test
devadmin@dev-web:~$ cd gazouille/
devadmin@dev-web:~/gazouille$
```

2. Créer le compte MySQL/MariaDB *gazou* qui a tous les droits sur la base de données *gazouille*.

```

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'gazou' IDENTIFIED BY 'azerty';
Query OK, 0 rows affected (0,003 sec)

MariaDB [(none)]> GRANT USAGE ON *.* TO 'gazou'@localhost IDENTIFIED BY 'azerty';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES
-> ;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> █

```

### Exercice 2 – Modèle Relationnel

Après analyse du Modèle Physique (requêtes de créations de la base de données), écrire le Modèle Relationnel de la base de données gazouille.

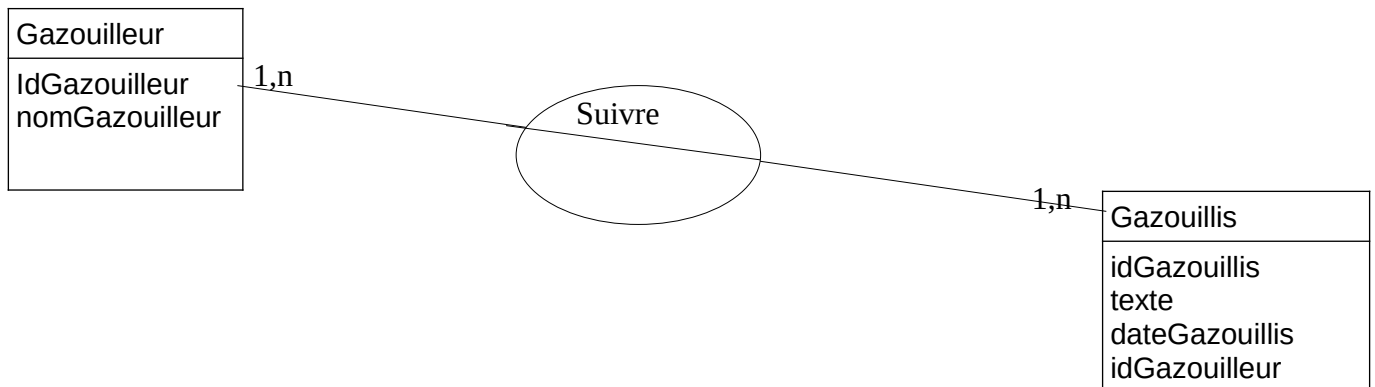
Gazouilleur (idGazouilleur, nomGazouilleur)

Gazouillis ( idGazouillis, texte, dateGazouillis, #idGazouilleur)

Suivre (#abonne, #suivi)

### Exercice 3 – Modèle Entité-Association

Représenter le Modèle Entité-Association de la base de données gazouille.



### Exercice 4 – Modèle Physique et interrogations

- Écrire puis exécuter le script peuplerGazouille.sql dont le rôle est de peupler la base de données gazouille (la rapport doit présenter l'état des tables Gazouilleur, Gazouillis et Suivre sous la forme de résultats de requêtes select \* from ...).

```

1 use gazouille;
2 insert into Gazouilleur (idGzouilleur,nomGazouilleur) values
3 (1,"Zakaria"),
4 (2,"Pepe"),
5 (3,"Ahmed"),
6 (4,"Yac");
7
8 insert into Gazouillis(idGazouillis, texte, dateGazouillis, idGazouilleur) values
9 (21,"Ok", "02-07-2021",1),
10 (22,"Ca marche", "06-12-2020",2),
11 (23, "vite", "07-11-2023",3),
12 (24, "yess", "01-03-2022",4),
13 (25, "Nonnn", "25-05-2021",3),
14 (26,"Alors", "30-05-2024",1);
15
16 insert into Suivre (idGazouilleur, idGazouillis) values
17 (1, 15),
18 (2, 16),
19 (3, 17),
20 (4, 18);
21
22
23
24
25

```

```

MariaDB [gazouille]> source creerGazouille.sql;
Query OK, 3 rows affected (0,066 sec)

Query OK, 1 row affected (0,000 sec)

Database changed
Query OK, 0 rows affected (0,014 sec)

Query OK, 0 rows affected (0,018 sec)

Query OK, 0 rows affected (0,016 sec)

MariaDB [gazouille]> source peuplerGazouille.sql;
Database changed
Query OK, 4 rows affected (0,017 sec)
Records: 4  Duplicates: 0  Warnings: 0

Query OK, 6 rows affected (0,004 sec)
Records: 6  Duplicates: 0  Warnings: 0

Query OK, 4 rows affected (0,011 sec)
Records: 4  Duplicates: 0  Warnings: 0

```

- Écrire les requêtes SQL d'interrogation suivantes (pour chaque requête d'interrogation, le rapport doit présenter le code de la requête ainsi que le résultat obtenu) :
  1. Liste des *gazouillis* postés par le *gazouilleur* numéro 2 ;

```
MariaDB [gazouille]> select * FROM Gazouillis where idGazouilleur = 2;
```

idGazouillis	texte	dateGazouillis	idGazouilleur
22	Ca marche	06-12-2020	2

2. Liste des *gazouillis* postés en 2021 ;

```
MariaDB [gazouille]> select * FROM Gazouillis where YEAR(dateGazouillis) = 2021;
```

idGazouillis	texte	dateGazouillis	idGazouilleur
21	Ok	2021-07-02	1
25	Nonnn	2021-05-25	3

3. Nombre de *gazouillis* postés par chaque *gazouilleur* ;

```
MariaDB [gazouille]> SELECT Gazouilleur.idGazouilleur, COUNT(Gazouillis.idGazouillis) AS PostGazouillis FROM Gazouilleur INNER JOIN Gazouillis ON Gazouilleur.idGazouilleur = Gazouillis.idGazouilleur GROUP BY Gazouilleur.idGazouilleur;
```

idGazouilleur	PostGazouillis
1	2
2	1
3	2
4	1

4. Liste des abonnés qui suivent le *gazouilleur* numéro 3 ;

```
MariaDB [gazouille]> SELECT suivi, abonne FROM Suivre WHERE suivi = 3;
```

suivi	abonne
3	4

5. Liste des *gazouilleurs* suivis par le *gazouilleur* numéro 4 ;

```
MariaDB [gazouille]> SELECT suivi, abonne FROM Suivre WHERE suivi = 4;
```

suivi	abonne
4	1

6. Nombre d'abonnés pour chaque *gazouilleur* ;

```
MariaDB [gazouille]> SELECT * FROM Suivre;
```

abonne	suivi
1	4
2	1
3	2
4	3

4 rows in set (0,000 sec)

7. Liste des *gazouilleurs* qui n'ont aucun abonné.

```
MariaDB [gazouille]> SELECT suivi, abonne FROM Suivre WHERE abonne = 0;
Empty set (0,000 sec)
```

Aucun

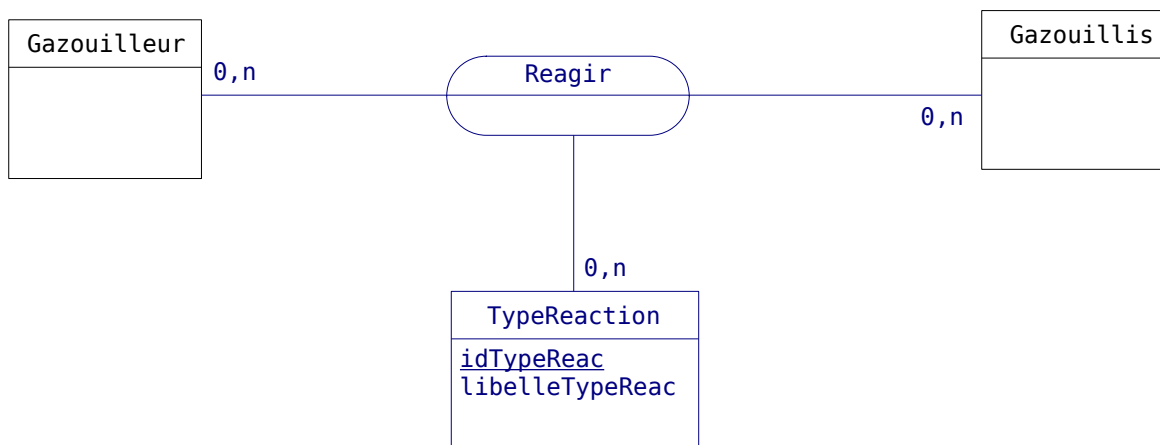
### 3- Évolution de la base de données

**Objectif :** Permettre aux utilisateurs (*gazouilleurs*) de réagir aux messages (*gazouillis*).

**Règles de Gestion :**

- **RG100 :** Tout *gazouilleur* peut réagir à un *gazouillis*.
- **RG101 :** Un type de réaction est caractérisé par un libellé ("J'adore", "Hilarant", "Amusant", "Content", "Triste", "Pourri", "Écœuré" ) et est identifié par un numéro.
- **RG102 :** Seule la dernière réaction d'un *gazouilleur* à un *gazouillis* est conservée dans la base de données.

**Modèle Entité-Association proposé :**

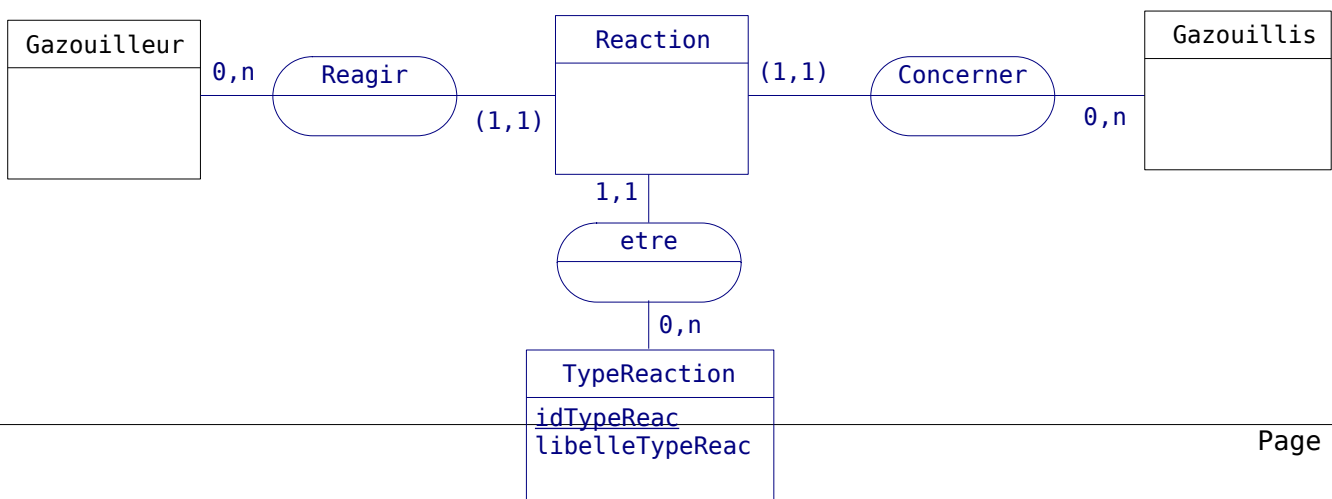


#### Exercice 5 – Fausse Ternaire

Le Modèle Entité-Association proposé ne respecte pas l'une des trois règles de gestion (nous sommes en présence d'une **fausse ternaire**). Quelle règle de gestion n'est pas respectée et pourquoi ?

La règle qui n'est pas respectée est la RG102 car on ne conserve que la dernière réaction et non les réactions d'avant

**Modèle Entité-Association corrigé :**



**Exercice 6 – Agrégat**

Que peut-on observer dans le Modèle Entité-Association corrigé ?

On peut observer l'ajout de deux associations relatif, « Reagir » entre les tables Gazouilleur et Reaction et « concerner » entre les tables Gazouillis et Reaction ainsi que l'association etre entre les tables Reaction et TypeReaction

**Exercice 7 – Modèle Relationnel**

Écrire le Modèle Relationnel relatif au Modèle Entité-Association corrigé.

Gazouilleur (idGazouilleur, nomGazouilleur)

Gazouillis (idGazouillis, texte, dateGazouillis, #idGazouilleur)

Reaction ( #idGazouilleur, # idGazouillis, #idTypeReac)

TypeReaction( idTypeReac, libelleTypeReac)

**Exercice 8 – Modèle Physique**

1. Compléter le code source du script creerGazouille.sql puis procéder à son exécution.

```
);  
  
create table TypeReaction (  
    idTypeReac int not null,  
    libelleTypeReac varchar (200) not null,  
    primary key (idTypeReac)  
);  
  
create table Reaction (  
    idGazouilleur int not null ,  
    idGazouillis int not null ,  
    idTypeReac int not null,  
    primary key (idGazouilleur, idGazouillis, idTypeReac)  
);
```

2. Exécuter le script peuplerGazouille.sql.

```
insert into TypeReaction ( idTypeReac,libelleTypeReac) values
(1, "Rapide"),
(2, "Drôle"),
(3, "Médiocre"),
(4, "Abusive"),
(5, "Triste");

insert into Reaction( idGazouilleur, idGazouillis, idTypeReac) values
(1,21,2),
(2,22,1),
(3,23,3),
(4,24,4),
(3,25,5),
(1,26,1);
```

3. Écrire puis exécuter le script `peuplerReactions.sql` dont le rôle est d'insérer des réactions dans la base de données gazouille (la rapport doit présenter l'état des tables `TypeReaction` et `Reagir` sous la forme de résultats de requêtes `select * from ...`).



```
MariaDB [gazouille]> source creerGazouille.sql;
Query OK, 3 rows affected (0,061 sec)

Query OK, 1 row affected (0,000 sec)

Database changed
Query OK, 0 rows affected (0,021 sec)

Query OK, 0 rows affected (0,021 sec)

Query OK, 0 rows affected (0,024 sec)

Query OK, 0 rows affected (0,020 sec)

Query OK, 0 rows affected (0,030 sec)

MariaDB [gazouille]> source peuplerGazouille.sql;
Database changed
Query OK, 4 rows affected (0,021 sec)
Records: 4 Duplicates: 0 Warnings: 0

Query OK, 6 rows affected (0,008 sec)
Records: 6 Duplicates: 0 Warnings: 0

Query OK, 4 rows affected (0,007 sec)
Records: 4 Duplicates: 0 Warnings: 0

Query OK, 5 rows affected (0,015 sec)
Records: 5 Duplicates: 0 Warnings: 0

Query OK, 6 rows affected (0,002 sec)
Records: 6 Duplicates: 0 Warnings: 0

MariaDB [gazouille]> show tables;
+-----+
| Tables_in_gazouille |
+-----+
| Gazouilleur          |
| Gazouillis           |
| Reaction             |
| Suivre               |
| TypeReaction         |
+-----+
5 rows in set (0,001 sec)
```



```

MariaDB [gazouille]> select * from Reaction
-> ;
+-----+-----+-----+
| idGazouilleur | idGazouillis | idTypeReac |
+-----+-----+-----+
|          1   |          21  |          2  |
|          1   |          26  |          1  |
|          2   |          22  |          1  |
|          3   |          23  |          3  |
|          3   |          25  |          5  |
|          4   |          24  |          4  |
+-----+-----+-----+
6 rows in set (0,000 sec)

MariaDB [gazouille]> select * from TypeReaction;
+-----+-----+
| idTypeReac | libelleTypeReac |
+-----+-----+
|          1 | Rapide          |
|          2 | Drôle           |
|          3 | Médiocre        |
|          4 | Abusive         |
|          5 | Triste          |
+-----+-----+
5 rows in set (0,001 sec)

```

### Exercice 9 – Interrogations

Écrire les requêtes SQL d'interrogation suivantes (pour chaque requête d'interrogation, le rapport doit présenter le code de la requête ainsi que le résultat obtenu) :

1. Type de la réaction du *gazouilleur* numéro 5 au *gazouillis* numéro 7 ;

Pour cette requête j'ai du rajouter quelques données dans le fichier peupler afin d'avoir un résultat.

```

7  (5, "Romain");
8
9  insert into Gazouillis(idGazouillis, texte, dateGazouillis, idGazouilleur) values
10 (21, "Ok", "2021-07-02", 1),
11 (22, "Ca marche", "2020-12-06", 2),
12 (23, "vite", "2023-11-07", 3),
13 (24, "yess", "2022-03-01", 4),
14 (25, "Nonnn", "2021-05-25", 3),
15 (26, "Alors", "2024-05-30", 1),
16 (7, "D'accord", "2024-07-02", 5);
17
18 insert into Suivre (abonne, suivi) values
19 (1, 4),
20 (2, 1),
21 (3, 2),
22 (4, 3);
23
24 insert into TypeReaction ( idTypeReac, libelleTypeReac) values
25 (1, "Rapide"),
26 (2, "Drôle"),
27 (3, "Médiocre"),
28 (4, "Abusive"),
29 (5, "Triste");
30
31 insert into Reaction( idGazouilleur, idGazouillis, idTypeReac) values
32 (1, 21, 2),
33 (2, 22, 1),
34 (3, 23, 3),
35 (4, 24, 4),
36 (3, 25, 5),
37 (1, 26, 1),
38 (5, 7, 2);
39

```

Résultat :

```
MariaDB [gazouille]> SELECT idGazouilleur, idGazouillis, libelleTypeReac FROM Reaction INNER JOIN TypeReaction on Reaction.idTypeReac = TypeReaction.idTypeReac WHERE idGazouilleur = 5 AND idGazouillis = 7;
```

idGazouilleur	idGazouillis	libelleTypeReac
5	7	Drôle

1 row in set (0,001 sec)

2. Type de la réaction de chaque *gazouilleur* ayant réagi au *gazouillis* numéro 7 ;

```
MariaDB [gazouille]> SELECT idGazouilleur, idGazouillis, libelleTypeReac FROM Reaction INNER JOIN TypeReaction on Reaction.idTypeReac = TypeReaction.idTypeReac WHERE idGazouillis = 7;
```

idGazouilleur	idGazouillis	libelleTypeReac
5	7	Drôle

1 row in set (0,001 sec)

3. Liste des *gazouilleurs* qui n'ont pas réagi au *gazouillis* numéro 7 ;

```
MariaDB [gazouille]> SELECT idGazouilleur, idGazouillis, libelleTypeReac FROM Reaction INNER JOIN TypeReaction on Reaction.idTypeReac = TypeReaction.idTypeReac WHERE idGazouillis != 7;
```

idGazouilleur	idGazouillis	libelleTypeReac
1	26	Rapide
2	22	Rapide
1	21	Drôle
3	23	Médiocre
4	24	Abusive
3	25	Triste

6 rows in set (0,002 sec)

4. Nombre de réactions au *gazouillis* numéro 7 ;

```
MariaDB [gazouille]> SELECT idGazouilleur, idGazouillis, count(*) AS NbTypeReac FROM Reaction INNER JOIN TypeReaction on Reaction.idTypeReac = TypeReaction.idTypeReac WHERE idGazouillis = 7 GROUP BY idGazouilleur, idGazouillis;
```

idGazouilleur	idGazouillis	NbTypeReac
5	7	1

1 row in set (0,001 sec)

5. Nombre de chaque type de réaction au *gazouillis* numéro 7 ;

```
MariaDB [gazouille]> SELECT idGazouilleur, idGazouillis, count(*) AS NbTypeReac FROM Reaction INNER JOIN TypeReaction on Reaction.idTypeReac = TypeReaction.idTypeReac WHERE idGazouillis = 7 GROUP BY idGazouilleur, idGazouillis;
```

idGazouilleur	idGazouillis	NbTypeReac
1	7	1
5	7	1

2 rows in set (0,001 sec)

6. Nombre de réactions de type "J'adore" pour chaque *gazouillis* ;  
j'ai modifier une réaction pour la mettre de type « j'adore » pour avoir un résultat.

```
MariaDB [gazouille]> SELECT idGazouilleur, idGazouillis, count(*) AS NbTypeReac FROM Reaction INNER JOIN TypeReaction on Reaction.idTypeReac = TypeReaction.idTypeReac WHERE TypeReaction.idTypeReac = 6 GROUP BY libelleTypeReac;
```

idGazouilleur	idGazouillis	NbTypeReac
1	7	3

1 row in set (0,001 sec)

7. Le ou les *gazouillis* qui ont obtenu le plus de réactions.

```
MariaDB [gazouille]> SELECT idGazouilleur, idGazouillis, count(*) AS NbTypeReac FROM Reaction INNER JOIN TypeReaction on Reaction.idTypeReac = TypeReaction.idTypeReac GROUP BY libelleTypeReac ORDER BY NbTypeReac DESC
```

idGazouilleur	idGazouillis	NbTypeReac
1	7	3
1	21	2
3	23	1
2	22	1

4 rows in set (0,001 sec)