

INFOM218 - DBRE - Groupe 2

# Étape 1

BENIMEDOURENE, Charles	CAUCHETEUR, Maxime
DECROP, Alix	DIERICKX, Jeremie
	JACOBS, Pierre

Université de Namur  
30 novembre 2022

# Table des matières

1	Smoke . . . . .	2
2	Schémas . . . . .	2
3	Découverte des clefs étrangères . . . . .	2
3.1	Clefs explicitement définies . . . . .	2
3.2	Simulation de découverte . . . . .	3

# 1 Smoke

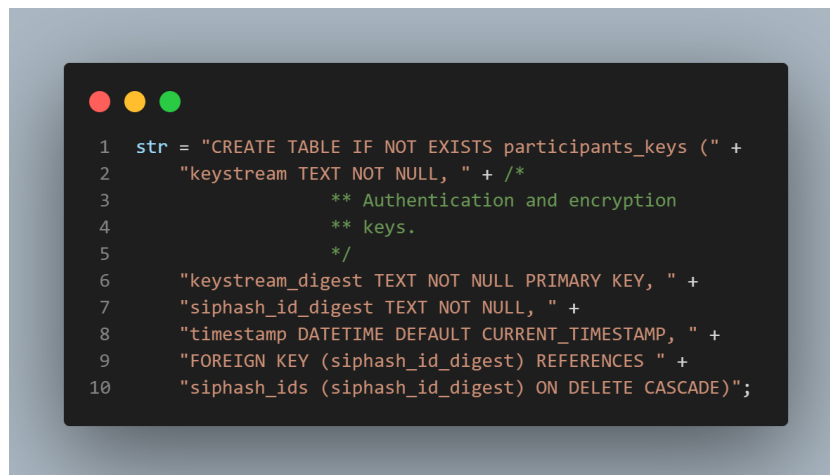
*Smoke* est une application Android de communication chiffrée entre participants. Celle-ci n'est cependant pas vouée à être utilisée en situation réelle, mais est plus une *expérimentation pour les curieux*. Celle-ci est écrite en Java et est disponible à cette adresse : <https://github.com/textbrowser/smoke>. Elle dispose d'une base de données *SQLite*. Celle-ci est par ailleurs composée de douze tables différentes.

L'objectif de ce document est de reconstruire les schémas physique, logique et conceptuel de la base de données à l'aide de l'outil *SQLInspect* et de les exprimer grâce au langage de modélisation *HyDRa*. Il propose également une simulation de découverte de clefs étrangères, afin de mesurer la facilité à retrouver lesdites clefs si elles n'étaient pas définies directement dans les requêtes *"CREATE TABLE"*.

## 2 Schémas

## 3 Découverte des clefs étrangères

Cette section se concentre sur la découverte des clefs étrangères présentes dans la base de données. La majorité de celles-ci sont explicitement décrites dans les requêtes *"CREATE TABLE"*.



```
1  str = "CREATE TABLE IF NOT EXISTS participants_keys (" +
2      "keystream TEXT NOT NULL, " + /*
3          ** Authentication and encryption
4          ** keys.
5          */
6      "keystream_digest TEXT NOT NULL PRIMARY KEY, " +
7      "siphash_id_digest TEXT NOT NULL, " +
8      "timestamp DATETIME DEFAULT CURRENT_TIMESTAMP, " +
9      "FOREIGN KEY (siphash_id_digest) REFERENCES " +
10     "siphash_ids (siphash_id_digest) ON DELETE CASCADE");
```

FIGURE 1 – Exemple de requête *CREATE TABLE*

La figure 1 montre bien que certaines des clefs étrangères sont explicitement définies à la création des tables : *FOREIGN KEY (siphash\_id\_digest) REFERENCES siphash\_ids (siphash\_id\_digest)*. Dans cet exemple, la table *participants\_keys* possède une référence à la table *siphash\_ids* grâce à son champ *siphash\_id\_digest*.

### 3.1 Clefs explicitement définies

Ci-dessous la liste des clefs étrangères explicitement définies, représentées sous la forme `<table>.<clef_etrangere> → <table>.<champ>` :

- `arson_keys.siphash_id_digest → siphash_ids.siphash_id_digest` ;
- `participants.siphash_id_digest → siphash_ids.siphash_id_digest` ;

- `participants_keys.siphhash_id_digest` → `siphhash_ids.siphhash_id_digest` ;
- `participants_messages.siphhash_id_digest` → `siphhash_ids.siphhash_id_digest`.

On notera tout de même que l'ensemble de ces clefs étrangères référencent le champ *siphhash\_id\_digest* de la table *siphhash\_ids*. Dans la suite de cette section, nous ferons l'hypothèse que ces clefs étrangères ne sont pas explicitement définies.

## 3.2 Simulation de découverte

En explorant le schéma généré par *SQLInspect*, l'on se rend compte que de quelques champs dispersés dans différentes tables portent des noms évocateurs et sont de bons candidats pour être des clefs étrangères.

**siphhash\_id\_digest** Les tables *arson\_keys*, *participants*, *participants\_keys*, *participants\_messages* et *siphhash\_ids* possèdent toutes un champ nommé *siphhash\_id\_digest*. Étant donné qu'une de ces tables se nomme *siphhash\_ids*, il est raisonnable de faire l'hypothèse que la table qui contient la référence *siphhash\_id\_digest* est effectivement la table *siphhash\_ids*. Par conséquent, les autres tables susmentionnées possèderaient effectivement la référence de ce champ-là comme clef étrangère.

**neighbor\_oid** La table *outbound\_queue* possède un champ nommé *neighbor\_oid*, qui coïncide singulièrement avec le champ *oid* de la table *neighbor*. Bien que la convention de nommage soit différente par rapport aux clefs étrangères *siphhash\_id\_digest*, il apparaît comme évident que le champ *neighbor\_oid* est effectivement un clef étrangère du champ *oid* de la table *neighbor*. En analysant la méthode *enqueueOutboundMessage* du fichier *Database.java*, il apparaît bel et bien que l'*oid* est ajouté après avoir récupéré les différents *oid* de la table *neighbor*, confirmant à nouveau que *neighbor\_oid* est bel et bien une clef étrangère.

**siphhash\_id** Le champ *siphhash\_id* de la table *participants* est également similaire au champ *siphhash\_id* de la table *siphhash\_ids*. Il est possible que ce champ soit également une clef étrangère. Cependant, en investigant le code source, il apparaît que le champ *siphhash\_id* de la table *participants* est uniquement utilisé dans la mise à jour des boîtes de dialogue des conversations entre participants. Par conséquent, le champ susmentionné ne semble pas être une clef étrangère.

Il est possible que d'autres clefs étrangères se cachent dans le code source. Cependant, aucun autre nom de champ n'apparaît de manière flagrante comme candidat pour être une clef étrangère. De plus, le code source étant non commenté et de manière plus générale complexe à comprendre, il a été décidé d'arrêter la recherche de clefs étrangères à celles trouvées précédemment.