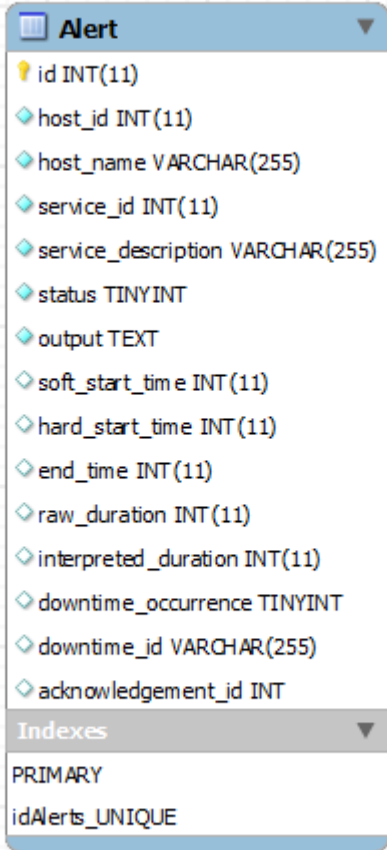


## 1. MCD

Le nom de la database : A définir dans le fichier de conf (dbnameAlert)

La table Alert → représente les alertes détectées par Centreon (avec date de début et fin)



Alert	
id	INT(11)
host_id	INT(11)
host_name	VARCHAR(255)
service_id	INT(11)
service_description	VARCHAR(255)
status	TINYINT
output	TEXT
soft_start_time	INT(11)
hard_start_time	INT(11)
end_time	INT(11)
raw_duration	INT(11)
interpreted_duration	INT(11)
downtime_occurrence	TINYINT
downtime_id	VARCHAR(255)
acknowledgement_id	INT
Indexes	
PRIMARY	
idAlerts_UNIQUE	

Le lien entre Alert et centreon\_storage.acknowledgement est assuré par le champ Alert.acknowledgement\_id.

Le lien entre Alert et Downtime est assuré par le champ Alert.downtime\_id.

**Attention**, ce dernier est au format texte car on peut avoir plusieurs downtimes associés à une alerte, chaque ID sera séparé par un espace.

## 2. LA TABLE ALERT

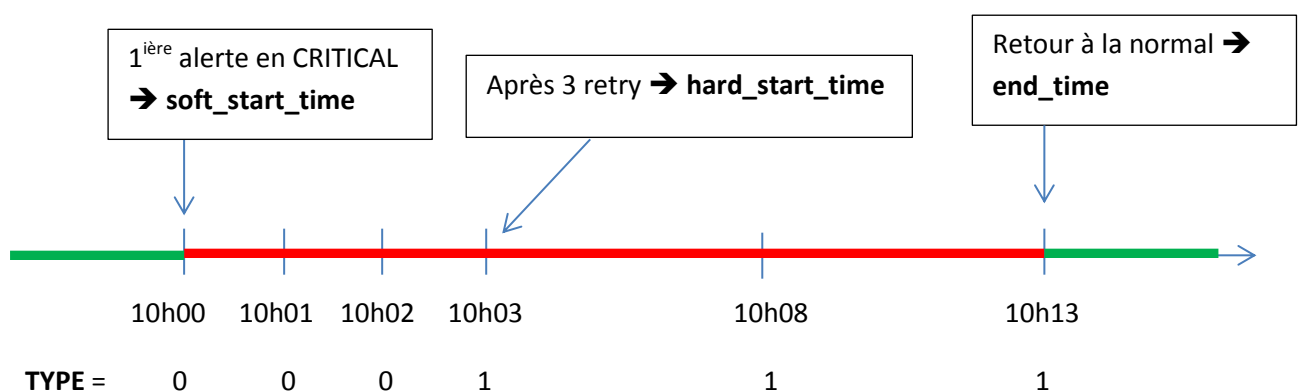
Champ violet → nouveau par rapport aux tables Centreon.

ALERT	EXEMPLE
id	5787154
host_id	1490
host_name	SR037506CTI3700
service_id	9702
service_description	NOHETO_URL_ExtranetPROD2_STATUS
status	2
output	Connexion refusée
soft_start_time	1472735956
hard_start_time	Détection en base → utilisation du champ type
end_time	1472736256
raw_duration	end_time - soft_start_time
interpreted_duration	
downtime_occurrence	Un entier (0 si pas de downtime)
acknowledgement_id	pointeur vers la table acknowledgement
downtime_id	pointeur vers la table downtime

### 2.1. soft\_start\_time

Unité : temps unix (sec)

Exemple simple d'une alerte CRITICAL survenue à 10h00 :



Il faut développer une fonction qui va s'occuper de créer une nouvelle entrée dans la table Alert si on détecte une nouvelle entrée dans la table centreon\_storage.logs

Les champs à copier :

centreon_storage.logs		<your_database>.Alert
id	➔	id
host_id	➔	host_id
host_name	➔	host_name
service_id	➔	service_id
service_description	➔	service_description
status	➔	status
output	➔	output
ctime	➔	soft_start_time

## 2.2. hard\_start\_time

Unité : temps unix (sec)

Il faut développer une fonction identique à soft\_start\_time sauf que le champ type de la table logs est utilisé (si type=1, on peut remplir le champ hard\_start\_time)

## 2.3. end\_time

Unité : temps unix (sec)

Dans la table logs, Si status=OK ou status=W|C|U et évtmt précédent non fini (c-a-d end\_time=NULL) du même couple Host+Service ➔ on peut enregistrer le champ ctime de l'entrée logs dans le champ end\_time de la table Alert.

## 2.4. raw\_duration

Unité : secondes

$\text{raw\_duration} = \text{end\_time} - \text{soft\_start\_time}$

## 2.5. downtime\_occurrence

Il s'agit du nombre de downtime(s) qui impacte(nt) une alerte (valeur par défaut = NULL).

Si la valeur est NULL, cela indique que le comptage n'a pas été fait.

Si la valeur est égal à zéro, il n'y a pas de downtime qui impacte l'alerte.

## 2.6. downtime\_id

Il faut parcourir la table Alert et on cible les alertes dont le champ end\_time différent de NULL et pour chaque entrée → on va parcourir la table downtimes qui impacte l'alerte ciblée. Si un downtime impacte l'alerte, on stocke (de façon incrémentale) l'ID du downtime.

Par exemple, si 3 downtimes ont impactés une alerte (cas rare), il y aura 3 ID différents (séparés par un espace) pour le champ *downtime\_id* de l'alerte.

## 2.7. acknowledgement\_id

De façon un peu identique à downtime\_id :

Il faut parcourir la table Alert et on cible les alertes dont le champ end\_time différent de NULL (alertes terminées) et pour chaque entrée → on parcourt la table Acknowledgement pour y trouver une entrée qui tombe dans le créneau de l'alerte ciblée → si c'est le cas on stocke l'ID de l'acquiescement dans le champ acknowledgement\_id.

## 2.8. interpreted\_duration

La durée de l'alerte en prenant en compte (algorithme simplifié, voir ci-dessous) les plages de downtime.

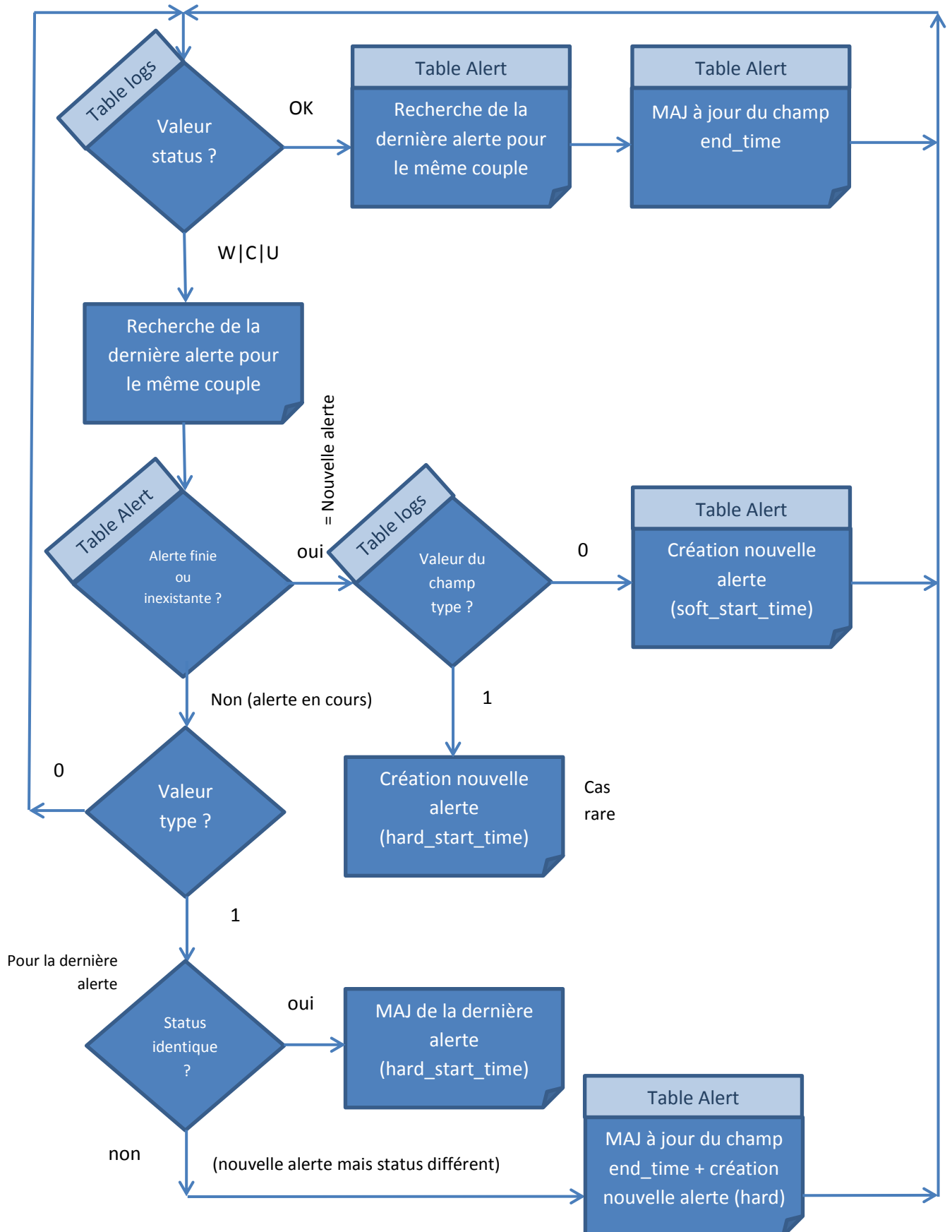
Voici l'algorithme :

Si nb de downtime pour une alerte = 0 → interpreted\_duration = raw\_duration

Si nb de downtime pour une alerte >= 1 → il y a 4 cas possible (**downtime au début, à la fin, au milieu ou total**) → il faut être capable de les distinguer et de calculer le champ interpreted\_duration en conséquence.

### 3. ALGORITHME

L'Algo parcourt la table logs chronologiquement, et pour chaque entrée de la table logs, on analyse :



#### 4. ANNEXE

BONUS : prévoir une fonction de purge des alertes de la table Alert pour ne garder que éviter que la table Alert ne soit trop volumineuse.