

## Archivage sortie

Dans procédure

Nom de la procédure	<input type="text" value="archivage_sortie"/>										
Type	<input type="text" value="PROCEDURE"/>										
Paramètres	<table><tr><th>Direction</th><th>Nom</th><th>Type</th><th>Taille/Valeurs*</th><th>Options</th></tr><tr><td colspan="5"><b>Ajouter un paramètre</b></td></tr></table>	Direction	Nom	Type	Taille/Valeurs*	Options	<b>Ajouter un paramètre</b>				
Direction	Nom	Type	Taille/Valeurs*	Options							
<b>Ajouter un paramètre</b>											
Définition	<pre>1 BEGIN 2     CALL extraction_sortie(); 3     CALL suppression_participant_sortie(); 4     CALL suppression_sortie(); 5 END</pre>										
Est déterministe	<input type="checkbox"/>										
Ajuster les privilèges 	<input checked="" type="checkbox"/>										
Créateur	<input type="text" value="`root` @ `localhost`"/>										
Type de sécurité	<input type="text" value="DEFINER"/>										
Accès aux données SQL	<input type="text" value="CONTAINS SQL"/>										

## Détails

Nom de la  
procédure

Type

PROCEDURE ▼

Paramètres

Direction

Nom

Type

Taille/Valeurs\*

Options

Ajouter un paramètre

Définition

```
1 BEGIN
2 INSERT INTO histoSorties
3 (id_sortie, nom_sortie, date_heure_debut, duree,
4 date_limite_inscription,
5 nb_inscriptions_max,
6 infos_sortie, motif_annulation, nom_organisateur, prenom, telephone,
7 mail, username)
8 (SELECT s.id, s.nom, date_debut, duree, date_cloture,
9 nb_inscription_max,
10 description_info, motif, p.nom, prenom, telephone, email, pseudo
11 FROM sortie s
12 INNER JOIN participant p ON s.participant_id = p.id
13 INNER JOIN etat e ON s.etat_id = e.id
14 WHERE DATE_ADD(date_debut, INTERVAL duree DAY) <= DATE_SUB(now(),
15 INTERVAL
16 30 DAY));
17 END
```

Est  
déterministe

☐

Ajuster les  
privilèges ?

☒

Créateur

Type de  
sécurité

DEFINER ▼

Accès aux  
données SQL

CONTAINS SQL ▼

## Détails

Nom de la procédure

Type

PROCEDURE ▼

Paramètres

Direction	Nom	Type	Taille/Valeurs*	Options
-----------	-----	------	-----------------	---------

Ajouter un paramètre

Définition

```
1 BEGIN
2     DELETE FROM participant_sortie
3     WHERE sortie_id IN
4     (SELECT DISTINCT id_sortie FROM histosorties);
5 END
```

Est déterministe

☐

Ajuster les privilèges ⓘ

☒

Créateur

Type de sécurité

DEFINER ▼

Accès aux données SQL

CONTAINS SQL ▼

## Détails

Nom de la procédure

Type

PROCEDURE ▼

Paramètres

Direction

Nom

Type

Taille/Valeurs\*

Options

Ajouter un paramètre

Définition

```
1 BEGIN
2     DELETE FROM sortie
3     WHERE id IN
4     (SELECT DISTINCT id_sortie FROM histosorties);
5 END
```

Est déterministe

☐

Ajuster les privilèges ⓘ

☒

Créateur

Type de sécurité

DEFINER ▼

Accès aux données SQL

CONTAINS SQL ▼

## Dans évènement

Nom de l'évènement	<input type="text" value="archivage_sortie"/>
État	<input type="text" value="ENABLED"/>
Type d'évènement	<input type="text" value="RECURRING"/>
Exécuter chaque	<input type="text" value="20"/>
	<input type="text" value="SECOND"/>
Départ	<input type="text" value="2022-10-10 10:40:22"/>
Fin	<input type="text"/>
Définition	<pre>1 CALL archivage_sortie()</pre>
Suite à l'exécution, conserver	<input checked="" type="checkbox"/>
Créateur	<input type="text" value="root@localhost"/>
Commentaire	<input type="text"/>

## Mise a jour des états

Dans procédure

### Détails

Nom de la procédure

Type

PROCEDURE ▼

Paramètres

Direction

Nom

Type

Taille/Valeurs\*

Options

Ajouter un paramètre

Définition

```
1 BEGIN
2 CALL majEtat3PourMaxInscrit();
3 CALL majEtat3PourDateAtteinte();
4 CALL majEtat4EnCours();
5 CALL majEtat5Terminee();
6 END
```

Est déterministe

☐

Ajuster les privilèges ⓘ

☒

Créateur

Type de sécurité

DEFINER ▼

Accès aux données SQL

CONTAINS SQL ▼

## Détails

Nom de la procédure

Type

PROCEDURE ▼

Paramètres

Direction	Nom	Type	Taille/Valeurs*	Options
-----------	-----	------	-----------------	---------

Ajouter un paramètre

Définition

```
1 BEGIN
2 UPDATE sortie set etat_id = 3
3 WHERE date_cloture < NOW()
4 and (etat_id = 1 OR etat_id = 2);
5 END
```

Est déterministe

☐

Ajuster les privilèges ⓘ

☒

Créateur

Type de sécurité

DEFINER ▼

Accès aux données SQL

CONTAINS SQL ▼

## Détails

Nom de la procédure

Type

Paramètres

Direction	Nom	Type	Taille/Valeurs*	Options
-----------	-----	------	-----------------	---------

Ajouter un paramètre

Définition

```
1 BEGIN
2 DELETE FROM T1;
3
4 INSERT INTO T1 (id, nb_ins_max, nb_ins)
5 (SELECT p.sortie_id, s.nb_inscription_max,
6 COUNT(p.sortie_id)
7 FROM participant_sortie p
8 INNER JOIN sortie s ON s.id = p.sortie_id
9 GROUP BY p.sortie_id, s.nb_inscription_max);
10
11 UPDATE sortie set etat_id = 3
12 WHERE id IN (SELECT id FROM T1 WHERE nb_ins_max =
13 nb_ins)
14 AND (etat_id = 1 OR etat_id = 2);
15 END
```

Est déterministe ☐

Ajuster les privilèges



Créateur

Type de sécurité

Accès aux données SQL



## Détails

Nom de la procédure

Type

PROCEDURE ▼

Paramètres

Direction

Nom

Type

Taille/Valeurs\*

Options

Ajouter un paramètre

Définition

```
1 BEGIN
2 UPDATE sortie set etat_id = 4
3 WHERE date_debut = NOW()
4 and (etat_id = 1 or etat_id = 2 or etat_id = 3);
5 END
```

Est déterministe

☐

Ajuster les privilèges ⓘ

☒

Créateur

Type de sécurité

DEFINER ▼

Accès aux données SQL

CONTAINS SQL ▼

## Détails

Nom de la procédure

Type

PROCEDURE ▼

Paramètres

Direction	Nom	Type	Taille/Valeurs*	Options
-----------	-----	------	-----------------	---------

Ajouter un paramètre

Définition

```
1 BEGIN
2 UPDATE sortie set etat_id = 5
3 WHERE DATE_ADD(date_debut,INTERVAL duree DAY) < NOW()
   and
4 (etat_id = 1 or etat_id = 2 or etat_id = 3 or etat_id =
   4);
5 END
```

Est déterministe

☐

Ajuster les privilèges

☒

Créateur

Type de sécurité

DEFINER ▼

Accès aux données  
SQL

CONTAINS SQL ▼

## Dans évènement

### Détails

Nom de l'évènement

État

ENABLED



Type d'évènement

RECURRING



Exécuter chaque

SECOND



Départ



Fin



Définition

```
1 CALL majEtat()
```

Suite à l'exécution, conserver



Créateur

Commentaire